Computer Vision and Image processing

Product Recognition on Store Shelves

Marini Luca
Baraghini Nicholas

University of Bologna
Artificial Intelligence Master Degree course
Automation Engineering Master Degree course

Contents

1 Introduction

2 Step A - Multiple Product Detection

Step A's objective is to identify and provide the position and dimensions of a single instance of five products, given as models, in scene images that contain multiple products. This task is solved by using a local invariant feature pipeline. This pipeline is repeated for each couple of model-target images.

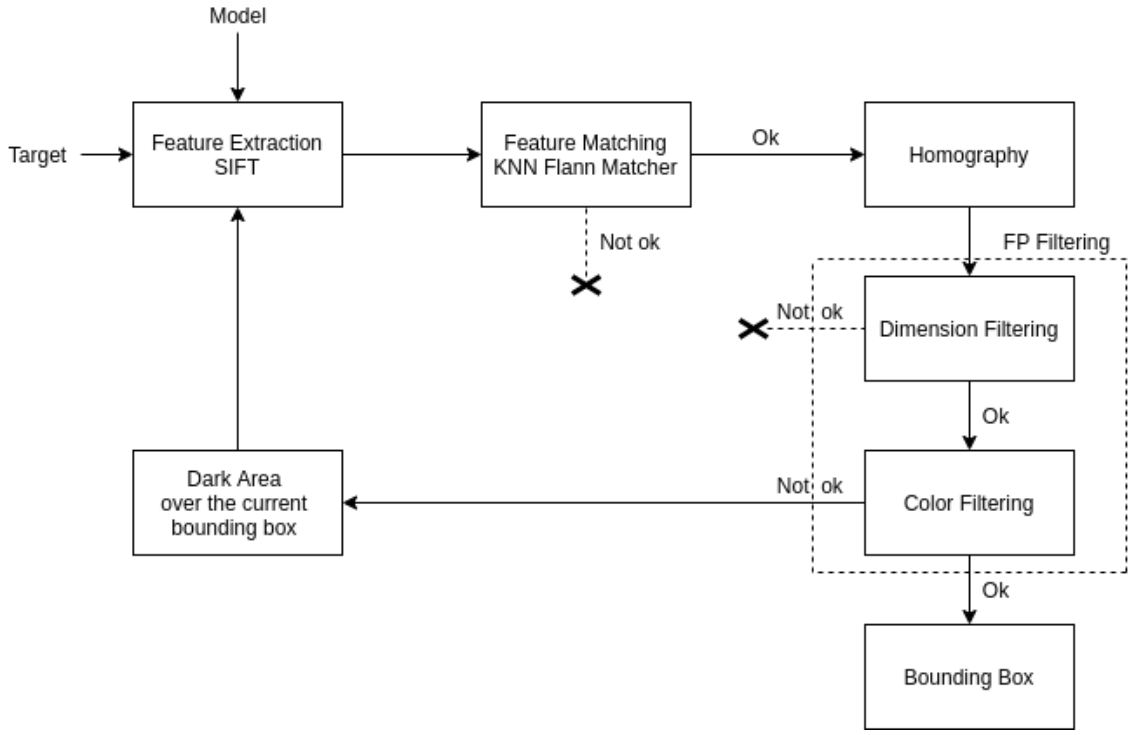

Figure 1: Step A Diagram

- **Model**:
  Image of the single product.
- **Target**:
  Image of the scene that contains multiple products.
- **Feature Extraction - SIFT**:
  Computation of SIFT keypoints and descriptors for each Model and Target image. Models' and Targets' keypoints and descriptors are computed a single time at startup to speed up the following steps' computation.
- **Feature Matching - KNN Flann Matcher**:
  Matching between keypoints using a KNN Flann Matcher based on KD-trees. If there are not enough matches, then the current Model is not found in the Target (scene) image.
- **Homography**:
  In case enough matches are found, then they are exploited to estimate an

Homography with RANSAC. So, a first possible bounding box is found in the target image.

- **Dimension Filtering**:
  Some inspections are implemented with the aim of discard false-positive instances. The first one verifies whether the current bounding box has a rectangular shape. In particular, if its width is greater than its height, then the bounding box is discarded.

- **Color Filtering**:
  Model and Target images are split into 12 bins. The mean of the intensities of the three color channels (r, g, b) is computed for each bin. Then, a pairwise bin comparison is performed. The means of the three color channels of two corresponding bins are compared. If the absolute difference between a single channel is higher than a chosen threshold, then the current pair of bins is considered not valid. In the case the number of pairs targeted as invalid overcomes a certain threshold, then the analyzed bounding box is discarded. Otherwise, if the current bounding box passes this color filtering, it means that the product is found in the scene.

- **Dark Area over the current bounding box**:
  The following step is applied if the color filter discards the current bounding box. A black rectangle is drawn over the discarded bounding box in the Target image, and the process is repeated from the beginning.

  The addition of the dark area solves the following issue: the local invariant feature pipeline can detect most of the keypoints of a Model into a wrong instance in the Target, which results to be identical from a feature point of view, but different in color (so it is a false positive). SIFT can cause this possible mistake because it finds keypoints invariantly to color.

  The dark area will cover the wrong colored product so that it will not be detected anymore.

  Afterward, the pipeline is restarted for the current Model-Target pair. In particular, keypoints and descriptors are recomputed only for the Target image with the added dark area, and the other steps are performed. This loop stops when one of the two following conditions are satisfied:

  - The bounding box passes the color filtering step, which means that the product's instance is found in the scene.
  - Alternatively, the number of found matches by the KNN Flann Matcher is not enough, which means that the product is not found in the Target image.

- **Bounding Box**:
  If a bounding box overcomes all the previous steps, then the Model's instance is found in the Target image.

3 Step B - Multiple Instance Detection

Step B's objective is to identify and provide the position and dimensions of one or multiple instances of five products, given as models, in scene images that contain multiple products. This task is solved by using a local invariant feature pipeline together with the GHT (Generalized Hough Transform). This pipeline is repeated for each couple of model-target images.

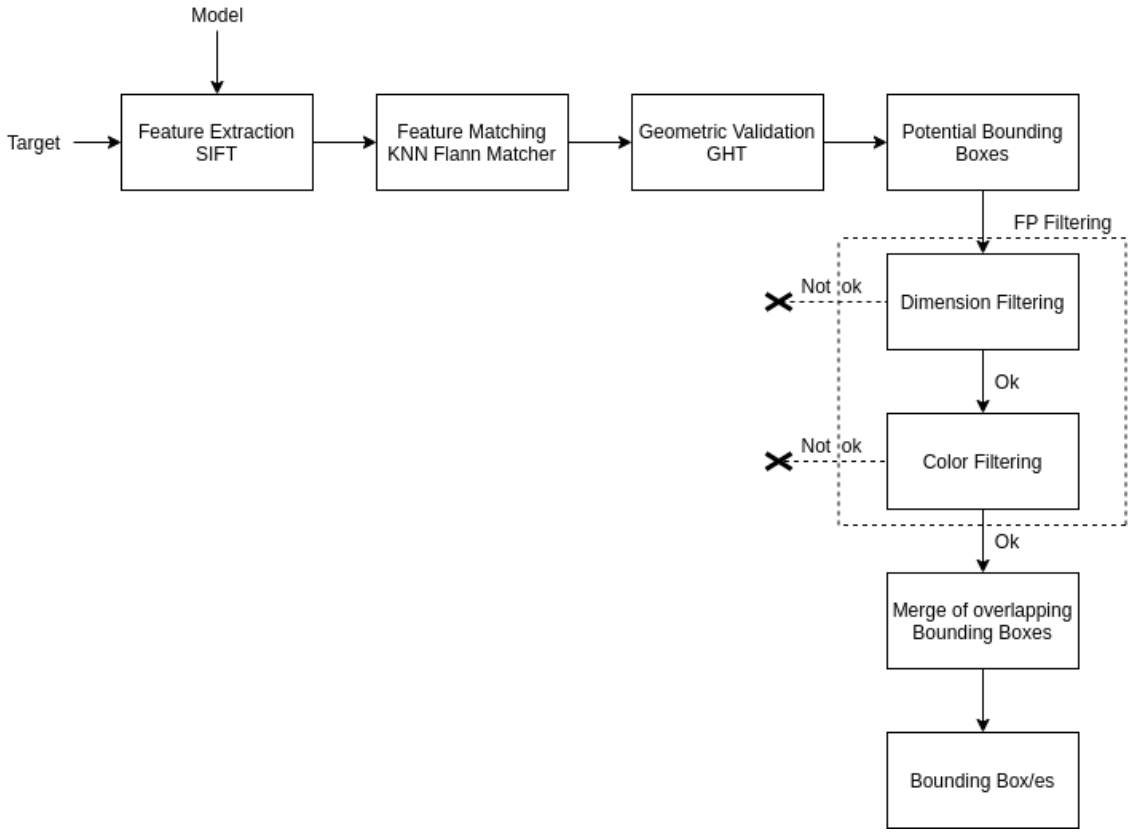

Figure 2: Step B Diagram

- **Model**:
  Same as Step A.
- **Target**:
  Same as Step A.
- **Feature Extraction - SIFT**:
  Same as Step A. Computation of Model's center point, which is the considered reference point. It is computed a single time at startup.
- **Feature Matching - KNN Flann Matcher**:
  Matching between keypoints using a KNN Flann Matcher based on KD-trees.
- **Geometric Validation - GHT**:
  The implementation of the GHT is the following:

1. Computation of the Model's joining vectors. The considered reference point is the center of the image.
2. The joining vectors are resized by the ratio between the scales of the matching keypoints. Rotation of joining vectors is not applied due to the presence of only upright products.
3. The voting process is performed. Each scaled joining vector casts a vote into the quantized Accumulator Array (AA) of the Target image. Thus, peaks are found in the AA. If there are no peaks, no Model's instances are found in the scene.
4. Then, for each peak, a bounding box is plotted. A peak represents a center of a bounding box. The bounding box dimensions (height and width) are calculated by applying a scale factor to the Model image's dimensions. The scale factor is computed as the mean of the ratio of the scales of the matching keypoints.

- **Potential Bounding Boxes**:
  Bounding boxes, found by the Generalized Hough Transform, that represent potential instances of the Model in the Target.
- **Dimension Filtering**:
  Same as Step A.
- **Color Filtering**:
  Same as Step A.
- **Merge of overlapping Bounding Boxes**:
  If two bounding boxes are too close (if the Euclidean distance between their top-left corners is smaller than a threshold), they are merged. The resulting bounding box has as corners coordinate the mean of the corresponding corners of the merged ones. This inspection is performed for every possible pair of bounding boxes.
- **Bounding Box/es**:
  All bounding boxes that overcome all the previous steps are instances of the Model found in the Target image.

4 Step C - Whole shelve challenge

Step C's objective is to identify and provide the position and dimensions of one or multiple instances of 24 products, given as models, in scene images that contain more than 40 different product instances for each picture and distractor elements (e.g., price tags). Some target images are low-resolution images. This task is solved using a local invariant feature pipeline and the GHT (Generalized Hough Transform). This pipeline is repeated for each couple of model-shelf images.

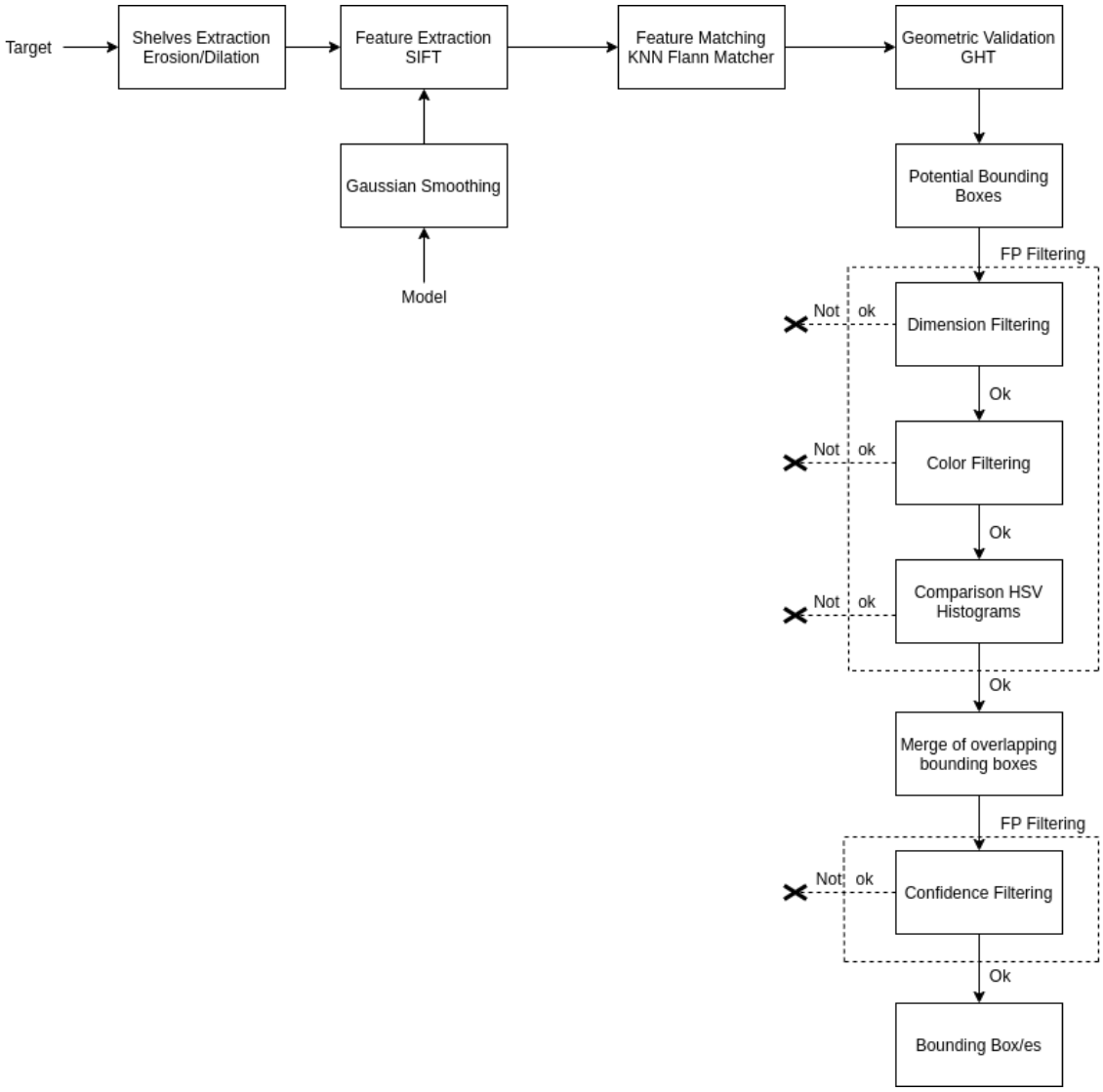

Figure 3: Step C Diagram

- **Model**:
  Same as Step A.

- **Target**:
  Same as Step A.

- **Gaussian Smoothing**:
  The Model image is smoothed with a Gaussian filter to make it more similar to its instances in the low-resolution Target images.

- **Shelves Extraction - Erosion/Dilation**:
  The whole image (Target) is split into the shelves to simplify the product detection task. The shelves' shape recognition is achieved by using binary morphology operators. In particular, a structuring element for extracting horizontal lines is applied to perform two erosions (to discard other shapes) and then some dilations to unify the partial horizontal shapes into shelves. An example of this step is visible in Figure 4 and Figure 5.

- **Feature Extraction - SIFT**:
  Same as Step B.

- **Feature Matching - KNN Flann Matcher**:
  Same as Step B.

- **Geometric Validation - GHT**:
  Same as Step B.

- **Potential Bounding Boxes**:
  Same as Step B.

- **Dimension Filtering**:
  If the dimensions (height and width) of a bounding box are higher than two respective thresholds, then the potential instance is discarded. The two bounds are computed by taking into account the shelf's height and the model image's ratio of dimensions.

- **Color Filtering**:
  It is the same as Step A, but now bins that are on the top and in the bottom rows are not compared for two reasons:

  1. To ignore possible labels or portions of shelves that fall into those bins and output high color differences.
  2. To neglect the products' top and bottom parts because they are often similar although the products are different.

  Therefore, the comparison of means between the three color channels' intensities is performed across six bins instead of twelve.

- **Comparison HSV Histograms**:
  The following step is added to improve the similarity of intensities between two images and discard more false positives.

  A comparison between the Model and each bounding box's image is performed. The two images are converted to HSV format to separate color from intensity. Then, the corresponding HS histograms are calculated and also normalized in order to compare them. Finally, the histograms' comparison is computed by using the Chi-Square distance as the metric.

$$d(H_1, H_2) = \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I)}$$

  If the distance is higher than a certain threshold, then the bounding box is discarded.

- **Merge of overlapping Bounding Boxes**:
  Same as Step B.

- **Confidence Filtering**:
  A confidence value based on the three parameters is computed to discard other false positives. The parameters are:

- The ratio between the number of votes received by the bounding box in the 3rd step of the GHT, and the minimum number of votes that a bin should receive to be considered a peak in the Accumulator Array.
- 1 - the ratio of the number of not valid pairs of bins of the color filtering step over the maximum allowed number of not valid pairs of bins.
- The fraction of the comparison value of the HSV histograms and the threshold of it.

The confidence value of a single bounding box is obtained through the following linear polynomial, where c1, c2, and c3 are three coefficients multiplied respectively to the three parameters:

$$Confidence = c1 \cdot \frac{\#_{votes^{AA}}}{min(\#_{votes^{AA}})} + c2 \cdot [1 - (\frac{\#_{not\_valid\_pairs\_bin}}{max(\#_{not\_valid\_pairs\_bin})})] + c3 \cdot \frac{histogram\_comparison}{min(histogram\_comparison)}$$

The previous formula gives different weights (coefficients) to the parameters (steps) and decreases false positives. If a bounding box's confidence value is higher than a threshold, then an instance of the product is found.

- **Bounding Box/es**:
  All bounding boxes that overcome all the previous steps are instances of the Model found in the Target image.

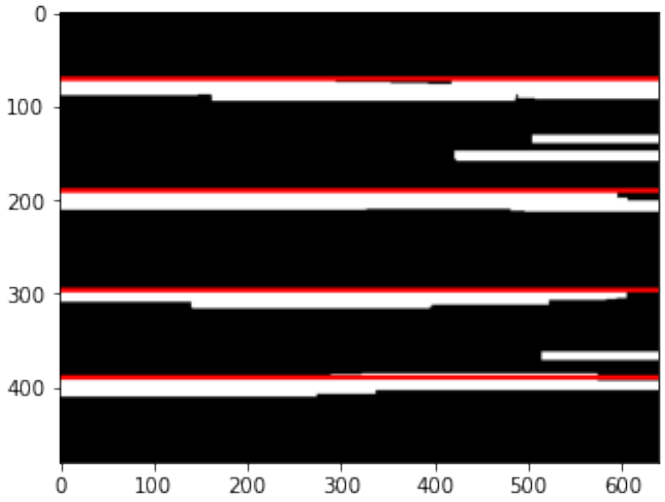

Figure 4: Shelves' shape recognition by using Binary Morphology operators

Figure 5: Detected shelves in the Target image

5 Results

5.1 Step A

All products are detected correctly in Step A, and the system finds zero false positives and does not miss any true positives. An example is visible in Figure 6.

Figure 6: Product recognition on a Target image - Step A

5.2 Step B

Also, in the whole Step B, all the instances of each product are detected correctly. The system finds zero false positives and does not miss any true positives. An example can be seen in Figure 7.

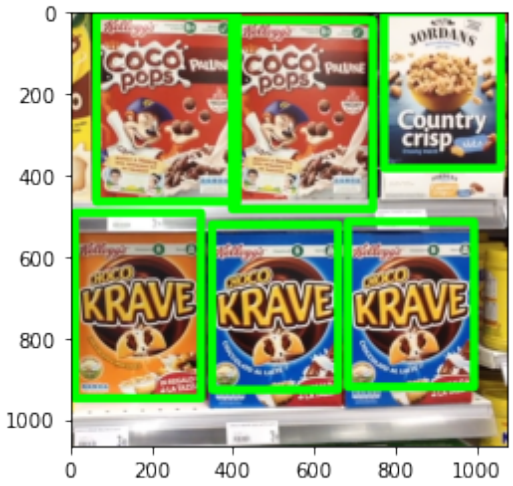

Figure 7: Product recognition on a Target image - Step B

5.3 Step C

The step C algorithm's performance is evaluated with the following metrics: *Precision*, *Recall*, and *F1-score*.

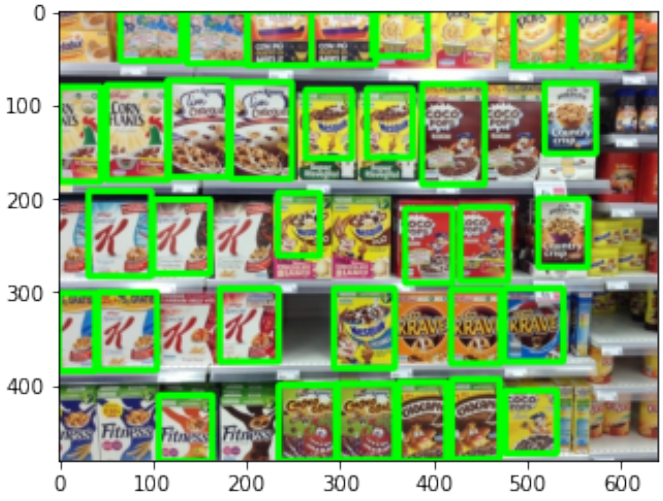

Figure 8: Product recognition on a Target image - Step C

Statistical data have been presented and rearranged in tables with seven columns:

- Model N°: ID associated with the model image;
- Name: textual identifier associated with the model;
- Counter Models: number of models associated with the specific row;
- Precision(%): Precision expressed as a percentage;
- Recall(%): Recall expressed as a percentage;
- F1(%): harmonic mean between Precision and Recall, expressed in percentage;

- % : the ratio between Counter Models, and the total number of models, expressed as a percentage;

Scene h1						
Model N°	Name	Counter Models	PRECISION (%)	RECALL (%)	F1(%)	%
0, 1, 3, 4, 11, 12, 14, 15, 18, 19, 22, 23, 6, 16, 17	Nchocomilk, MilkCKrave, ChokoGoal, SlimChocolate, NutCKrave, FitnessFruit, Chocapic, Rotelle, JordansChocolate, JordansNuts, HoneyCheerics, KellogsClassic2,NesquikDuo, MielPopsNut, MielPopsAnellini	15	100	100	100	62.5
2, 5, 7	CornFlakes, Nesquik, RisoChock	3	66.66666667	100	80	12.5
8, 10, 20, 21	ChocoPopsPalline, KellogsDarkChocolate, KellogsStrawberrys, RiceKrispies	4	100	50	66.6667	16.6667
9,13	KellogsClassics1, FitnessLigth	2	100	0	0	8.33333

Figure 9: Statistical data referred to scene h1

Scene h2						
Model N°	Name	Counter Models	PRECISION (%)	RECALL (%)	F1(%)	%
0, 1, 2, 3, 4, 6, 7, 8, 11, 14, 15, 18, 19, 21, 22, 23	Nchocomilk, MilkCKrave, CornFlakes, ChokoGoal, SlimChocolate, NesquikDuo, RisoChock, ChocoPopsPalline, NutCKrave, Chocapic, Rotelle, JordansChocolate, JordansNuts, RiceKrispies, HoneyCheerics, KellogsClassic2	16	100	100	100	66.6667
16	MielPopsNut	1	100	50	66.6667	4.16667
5, 17	Nesquik, MielPopsAnellini	2	50	50	50	8.33333
12, 13, 20	FitnessFruit, FitnessLigth, KellogsStrawberrys	3	100	0	0	12.5
9	KellogsClassics1	1	0	100	0	4.16667
10	KellogsDarkChocolate	1	0	0	0	4.16667

Figure 10: Statistical data referred to scene h2

Scene h3						
Model N°	Name	Counter Models	PRECISION (%)	RECALL (%)	F1 (%)	%
0, 1, 3, 4, 6, 7, 8, 11, 13, 14, 15, 16, 17, 18, 19, 21, 22	Nchocomilk, MilkCKrave, ChokoGoal, SlimChocolate, NesquikDuo, RisoChock, ChocoPopsPalline, NutCKrave, FitnessLigth, Chocapic, Rotelle, MielPopsNut, MielPopsAnellini, JordansChocolate, JordansNuts, RiceKrispies, HoneyCheerics	17	100	100	100	70.8333
2, 20, 23	CornFlakes, KellogsStrawberrys, KellogsClassic2	3	100	50	66.6667	12.5
5	Nesquik	1	50	100	66.6667	4.16667
10	KellogsDarkChocolate	1	28.57142857	100	44.4444	4.16667
9, 12	KellogsClassics1, FitnessFruit	2	100	0	0	8.33333

Figure 11: Statistical data referred to scene h3

Scene h4						
Model N°	Name	Counter Models	PRECISION (%)	RECALL (%)	F1 (%)	%
0, 1, 2, 3, 4, 8, 14, 15, 18, 19, 21, 22, 6, 9	Nchocomilk, MilkCKrave, CornFlakes, ChokoGoal, SlimChocolate, ChocoPopsPalline, Chocapic, Rotelle, JordansChocolate, JordansNuts, RiceKrispies, HoneyCheerics, NesquikDuo, KellogsClassics1	14	100	100	100	58.3333
23	KellogsClassic2	1	75	100	85.7143	4.16667
5	Nesquik	1	66.66666667	100	80	4.16667
7, 10, 11, 12	RisoChock, KellogsDarkChocolate, NutCKrave, FitnessFruit	4	100	50	66.6667	16.6667
17	MielPopsAnellini	1	33.33333333	50	40	4.16667
13, 16, 20	FitnessLigth, MielPopsNut, KellogsStrawberrys	3	100	0	0	12.5

Figure 12: Statistical data referred to scene h4

Scene h5						
Model N°	Name	Counter Models	PRECISION (%)	RECALL (%)	F1 (%)	%
0, 1, 2, 3, 4, 7, 8, 11, 15, 18, 19, 22, 23, 6, 9, 13	Nchocomilk, MilkCKrave, CornFlakes, ChokoGoal, SlimChocolate, RisoChock, ChocoPopsPalline, NutCKrave, Rotelle, JordansChocolate, JordansNuts, HoneyCheerics, KellogsClassic2, NesquikDuo, KellogsClassics1, FitnessLigth	16	100	100	100	66.6667
5, 10	Nesquik, KellogsDarkChocolate	2	66.66666667	100	80	8.33333
16	MielPopsNut	1	100	50	66.6667	4.16667
17	MielPopsAnellini	1	50	50	50	4.16667
12, 14, 20, 21	FitnessFruit, Chocapic, KellogsStrawberrys, RiceKrispies	4	100	0	0	16.6667

Figure 13: Statistical data referred to scene h5

The tables above show that for most models, the harmonic mean between precision and recall results to be maximum in all the five scenes provided.

For the remaining portion of models, it can be noticed a small percentage of models with an F1 score equal to zero. An exception to this case is scene h2, which shows 20.84% of models (5 product brands) with a null harmonic mean due to a lack of prominent keypoints or low resolution of the image.

Nevertheless, the algorithm still shows proper functioning in other scene images, retaining a small number of non-detected boxes and a considerably high number of well-recognized instances. This conclusion can be drawn out, considering that the percentage of model boxes detected with an F1 score above 60% is always higher than 70% of the total number of models provided. Moreover, positive peaks of 91.7% and 87.5% of models with an F1 score above 66% can be acknowledged respectively from scenes h1 and h3.

6 Conclusions

In conclusion, the algorithm performs perfectly in both Step A and Step B. Finally, in Step C, it can always find the majority of product instances in target images.

However, it commits errors in detecting some instances. Those mistakes are mostly present in products that show very general features that are similar across multiple models.

An additional source of errors is the presence of blur in the scenes. In particular, it causes the feature matching algorithm not to find keypoints in low-resolution instances of products.

From these considerations, it can be deduced that, in general, the algorithm shows a good trend concerning instances of the model that occur at relatively small dimensions in the scene and with disturbing elements, such as bar codes, price labels of the product, etc. Finally, when the instances are presented in good resolution conditions, the algorithm thus developed performs with maximum precision and recall.