Product Recognition on Store Shelves

Francesco Olivo
Image Processing & Computer Vision

November 2023

Abstract

Object detection techniques based on computer vision can be deployed in super market scenarios for the creation of a system capable of recognizing products on store shelves. Given the image of a store shelf, such a system should be able identify the different products present therein and may be deployed, e.g. to help visually impaired costumers or to automate some common store management tasks (e.g. detect low in stock or misplaced products).

1 Introduction

The goal of the project is to develop a computer vision system that, given a reference image for each product, is able to identify boxes of cereals of different brands from one picture of a store shelf. For each type of product displayed in the shelf the system should report:

1. Number of instances.
2. Dimension of each instance (width and height of the bounding box that enclose them in pixel).
3. Position in the image reference system of each instance (center of the bounding box that enclose them in pixel).

In particular, the task is divided into two major steps, each with increasing difficulty, both on the task and on the dataset structure.

1.1 Step A - Multiple Product Detection

In this step of the project, the goal is to identify a single instance of given products: there is one reference image for each product and a scene image. The system should be able to correctly identify all of the products in the shelves image.

1.2 Step B - Multiple Instance Detection

In this step of the project, the goal is to identify a multiple instances of given products. The requirement is to adopt local invariant features together with the GHT (Generalized Hough Transform), where, instead of relying on the usual R-Table, the object model

acquired at training time should now consist in vectors joining all of the features extracted in the model image to their barycenter; then, at run time all of the image features matched with respect to the model would cast votes for the position of the barycenter by scaling appropriately the associated joining vectors (i.e. by the ratio of sizes between the matching features).

2 Literature Review

2.1 Local Invariant Feature Detection using SIFT

The Scale-Invariant Feature Transform (SIFT) algorithm, developed by David Lowe [1], represents a significant advancement in the field of computer vision, particularly in the context of local feature detection and matching. This algorithm has been extensively used and cited in a wide range of applications, from object recognition and image stitching to 3D modeling and motion tracking.

SIFT operates through several stages to detect and describe local features in images. The process begins with the *scale-space extrema detection*, where potential interest points are identified across various scales. This is achieved by generating a set of Difference of Gaussian (DoG) images and locating scale-space extrema.

The subsequent stage involves *keypoint localization*. Here, the algorithm refines the initial set of keypoints to enhance accuracy and eliminate points of low contrast or those poorly localized along an edge.

Following this, each keypoint is assigned an *orientation* based on local image gradients. This crucial step ensures the resulting feature descriptors are invariant to image rotation.

Finally, the *keypoint descriptor* is created. Each descriptor is a highly distinctive vector representing the local gradient information surrounding the keypoint, which allows for robust matching of features across different images.

The SIFT algorithm's ability to detect features that are invariant to scale and rotation, as well as robust to changes in illumination, noise, and minor variations in viewpoint, makes it a powerful tool in the realm of computer vision.

2.2 Computing Homography using RANSAC

The Random Sample Consensus (RANSAC) [2] algorithm is a robust method widely used in computer vision for estimating a mathematical model from a dataset that contains outliers. In the context of finding homographies, which are transformations that map one plane to another, RANSAC plays a pivotal role in accurately computing the transformation despite the presence of mismatches or outliers in feature point correspondences.

Homography estimation is crucial in various applications, such as image stitching, 3D reconstruction, and motion tracking. The process typically involves matching sets of points between two images (e.g., using feature detection algorithms like SIFT or SURF). However, these matched points may contain incorrect correspondences due to noise, feature ambiguity, or moving objects in the scene.

RANSAC addresses these challenges by iteratively selecting a random subset of the matched points and estimating the homography. It then counts the number of inliers, which are points that fit well to the estimated homography within a certain tolerance. This process is repeated for a predefined number of iterations, and the homography with the highest number of inliers is chosen as the final model.

The robustness of RANSAC lies in its ability to tolerate a significant percentage of outliers, ensuring that the estimated homography is predominantly influenced by the correct correspondences. This makes RANSAC an essential tool in scenarios where the data is contaminated with a substantial amount of noise or outliers.

Moreover, various enhancements and adaptations of the standard RANSAC algorithm have been proposed in the literature to improve its efficiency, accuracy, and speed, making it a continuously evolving technique in the field of computer vision.

2.3 LAB Color Space

The LAB color space, also known as CIELAB, is a color space defined by the International Commission on Illumination (CIE) in 1976. It is designed to be perceptually uniform with respect to human color vision, meaning that the same amount of numerical change in these values corresponds to roughly the same amount of visually perceived change.

The LAB color space consists of three components: L^* for lightness, and a^* and b^* for the color-opponent dimensions. L^* ranges from 0 to 100, representing black to white, while a^* and b^* represent color dimensions. a^* goes from green to red, and b^* from blue to yellow.

This color space is particularly useful in image processing tasks where a more human-like perception of color differences is required. Unlike RGB or CMYK color spaces, which are device-dependent and do not align well with the way humans perceive colors, the LAB color space offers a more intuitive understanding and manipulation of colors in digital images.

Applications of the LAB color space are widespread in tasks that require accurate color comparison and manipulation, such as image editing, color correction, and digital restoration. Its perceptual uniformity allows for more effective and natural adjustments in these applications.

2.4 Creating Color Bins and Computing Differences using Cosine Distance

Creating color bins and computing color differences using cosine distance is a significant method in image processing for tasks like image segmentation, object recognition, and color-based image retrieval. This approach involves quantizing the color space into a finite number of bins and then comparing color histograms of images or image regions.

The process begins with the division of a chosen color space (such as RGB, HSV, or LAB) into a number of bins. Each bin represents a range of colors. By mapping the colors of an image into these bins, a color histogram is created, which effectively captures the color distribution of the image.

Cosine distance is then employed to measure the similarity between two color histograms. The cosine distance is a metric used to evaluate the cosine of the angle between two vectors. In the context of color histograms, it provides a measure of similarity between two color distributions, where a smaller angle (and thus a higher cosine value) indicates a higher degree of similarity.

The application of cosine distance in comparing color histograms is particularly beneficial due to its invariance to changes in image scale and its robustness in handling variations in illumination. This makes it a suitable choice for comparing images in different lighting conditions or of different sizes, which is often the case in real-world scenarios.

This method's effectiveness in capturing and comparing color information makes it a valuable tool in various image processing and computer vision applications, especially where color plays a crucial role in the task at hand.

2.5 Generalized Hough Transform

The Generalized Hough Transform (GHT) [3] is a pivotal technique in computer vision, developed as an extension of the classical Hough Transform. It's primarily used for detecting and identifying arbitrary shapes in images, a significant expansion from the original method's limitation to simple geometric figures like lines and circles.

GHT operates on the principle of a voting procedure in a parameter space. For each feature point in an image, typically an edge point, votes are cast for all parameter sets that could potentially produce this feature. The accumulation of these votes results in local maxima in the parameter space, signifying the presence of a specific shape in the image. This innovative approach allows GHT to be highly effective in recognizing objects of various sizes, orientations, and positions, even in the presence of noise and partial occlusions.

In practical applications, GHT is extensively used in object recognition, pattern recognition, image segmentation, and tracking. Its robustness against variations in object appearance and its ability to synthesize information from individual pixels into more complex structures make it a vital tool in complex image analysis tasks. The Generalized Hough Transform thus remains a fundamental and versatile method in the field of computer vision, adaptable to a wide range of scenarios and capable of handling diverse pattern recognition challenges.

3 Methodology - Step A

The adopted method employs an iterative approach, systematically processing all scene images against each product image. This method is designed to accurately identify and localize products within complex scenes, employing a series of steps to ensure precision and robustness.

3.1 Feature Extraction and Initial Matching

Initially, for each product image, key features are extracted using the Scale-Invariant Feature Transform (SIFT). SIFT is particularly adept at identifying distinctive features that are invariant to scale and rotation, making it ideal for matching objects under varying conditions. Once the keypoints and their corresponding descriptors are obtained, the algorithm proceeds to match these features with those found in the scene images.

3.2 Adaptive Thresholding and Descriptor Count for Match Filtering

In the matching phase, an adaptive threshold of 0.6 is used to filter good matches, which is lower than the traditional 0.75 threshold suggested by Lowe. Additionally, a match is considered to be good only if there are at least 50 matching descriptors. This criterion ensures that the matches are not only quantitatively substantial but also qualitatively

strong. The combination of a lower threshold and a minimum descriptor count balances increased sensitivity in detecting potential matches with the assurance of match reliability, especially important in scenes with variable conditions.

3.3 Homography Computation and Localization

For each set of good matches, a homography is computed to establish a transformation matrix between the product and the scene image. This step is crucial for localizing the exact patch within the scene where the product is present. The computed homography accounts for potential changes in scale, orientation, and perspective, accurately mapping the product image to its corresponding location in the scene.

3.4 Verification Tests

Once the product patch is localized in the scene image, it undergoes a series of rigorous verification tests:

1. **Aspect Ratio Check**: The first test involves examining the aspect ratio of the detected patch. If the width of the patch exceeds its height, the patch is discarded. This criterion is based on the domain-specific knowledge that no product in the dataset is wider than its height. This check effectively filters out improbable matches and reduces false positives.

2. **Color Histogram Comparison**: Next, the color histograms in the LAB color space are computed for both the product image and the detected patch. The LAB color space is chosen for its perceptual uniformity, closely aligning with human color vision. The cosine distance between these two histograms is then calculated. If this distance exceeds 0.02, the patch is disregarded. This step is particularly crucial for distinguishing between products that are nearly identical in shape and size but differ in color, thereby enhancing the precision of the matching process.

3. **Non-Maxima Suppression (NMS)**: Finally, to prevent multiple detections of the same product instance, Non-Maxima Suppression is applied. This involves comparing the Intersection over Union (IoU) of detected patches. If the IoU exceeds 0.5, indicating a significant overlap, the patches are further compared based on the number of good matches. The patch with fewer matches is then discarded. This step ensures that each product instance is uniquely identified and avoids redundant detections.

This method, through its iterative and multi-stage approach, ensures high accuracy in product detection within various scene images. The combination of adaptive feature matching, precise homography computation, and rigorous post-detection checks effectively addresses the challenges posed by diverse and complex scene compositions.

4 Methodology - Step B

The second step of the project involves an advanced approach to detect multiple instances of given products within a scene, employing a modified version of the Generalized Hough Transform (GHT). This step is split into two distinct phases: the training (offline) phase

and the online phase for each scene, building upon methodologies similar to those used in the previous step but with significant enhancements to address the complexity of multiple instance detection.

4.1 Training (Offline) Phase

During the training phase, the focus is on constructing a robust model for each product. This involves computing the Hough space for matches found between the scene and the product images. The methodology here follows the modified GHT approach, where vectors from the local invariant features to the object's barycenter are used, as described earlier.

Once the Hough space is determined, the next critical step is the quantization and creation of an accumulator matrix. This matrix is pivotal, as it stores the votes for each potential object location in the scene, based on the computed Hough space. The votes are cast in accordance with the modified R-table approach, where the scaling of vectors is dynamically adjusted based on the size ratio of matching features.

After the voting process, the accumulator matrix is analyzed to identify local maxima, which indicates potential locations of the product in the scene. These local maxima are then subsampled to refine the detection process, ensuring that only the most likely product locations are considered for further analysis.

4.2 Online Phase

In the online phase, the trained model is applied to new scene images to detect instances of the product. This involves scanning the scene with the trained accumulator matrix to find correspondences between the scene and the product. Each detected correspondence undergoes a series of verification tests as outlined in Section 3.4, ensuring the reliability of the detection.

4.2.1 Non-Maxima Suppression (NMS)

A notable difference in this step is the adaptation of the Non-Maxima Suppression (NMS) method. Unlike the previous approach where NMS was based on the number of good matches, here the decisive factor for NMS is the number of votes in the accumulator matrix. When overlapping detections occur, the detection with the larger number of votes is retained. This adjustment acknowledges the importance of the voting process in the modified GHT and ensures that the most confidently detected instances are prioritized, enhancing the overall accuracy and reliability of multiple product detections in complex scenes.

5 Results

5.1 Step A

Among the five proposed scenes, our system was able to correctly identify all of the products within each scene. The most salient ones are proposed:

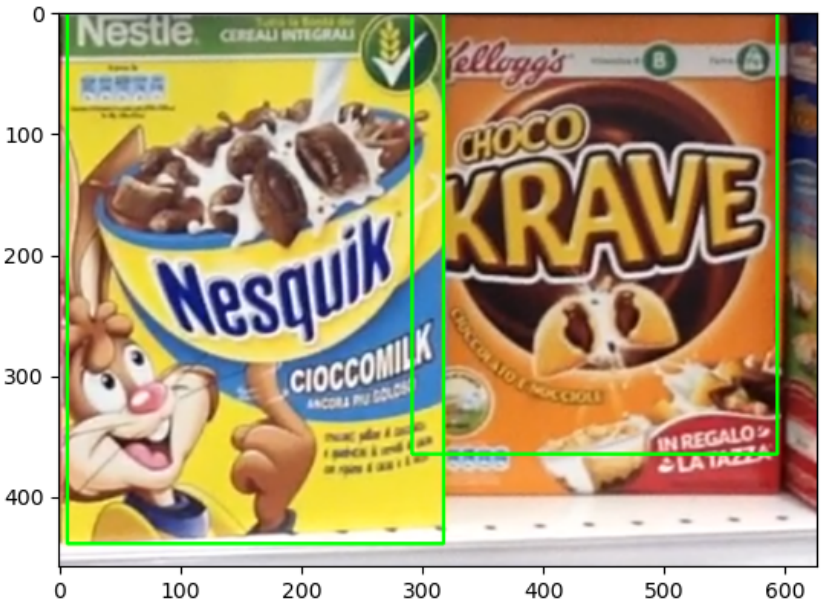

Figure 1: Scene e1

5.1.1 Scene 1

```
SCENE e1.png
        Product 0.jpg (Nesquik no scritte) - Instance found:
                Instance 1 {position: (162.50, 219.50), width:
                   311.00px, height: 439.00px}, good matches are
                   289.
        Product 11.jpg (ChocoKrave) - Instance found:
                Instance 1 {position: (443.00, 182.50), width:
                   302.00px, height: 365.00px}, good matches are
                   81.
```

This is the first proposed scene, we can see how the system is able to correctly identify both of the products, with only a minor overlapping between the two identified regions.

5.1.2 Scene 3

Scene 3 offers the interesting scenario where two of the products are basically identical, with the only relevant difference being the color of the box. This is the situations which made necessary to analyze the distance between color bins.

```
SCENE e3.png
        Product 0.jpg (Nesquik no scritte) - Instance found:
                Instance 1 {position: (170.00, 233.00), width:
                   328.00px, height: 440.00px}, good matches are
                   259.
        Product 1.jpg (ChocoKrave blu) - Instance found:
                Instance 1 {position: (817.00, 199.00), width:
                   314.00px, height: 398.00px}, good matches are
                   133.
        Product 11.jpg (ChocoKrave) - Instance found:
```

Figure 2: Scene e3

```
Instance 1 {position: (475.00, 193.50), width:
   308.00px, height: 387.00px}, good matches are
   59.
```

We can see how the system correctly identifies the two slightly different versions of the product.

5.1.3 Scene 4

Scene 4 offers another interesting scenario, since there are two products with a very similar box.

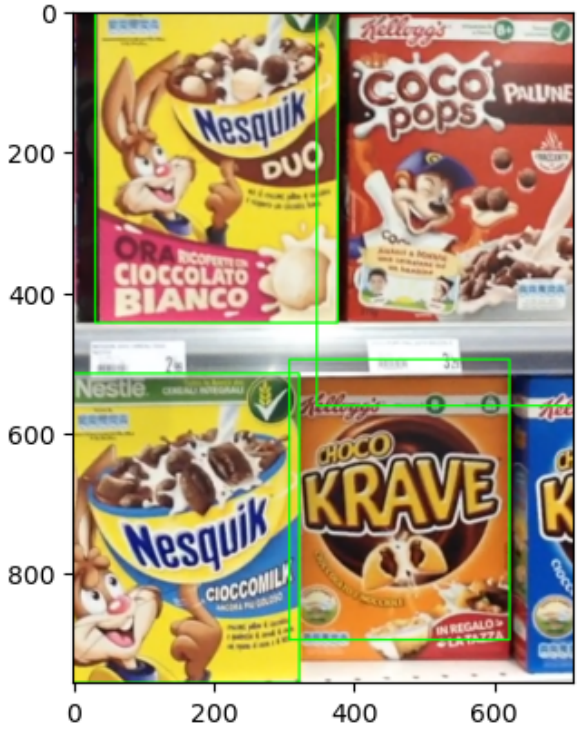

Figure 3: Scene e4

```
SCENE e4.png
        Product 0.jpg (Nesquik no scritte) - Instance found:
```

```
            Instance 1 {position: (161.00, 736.00), width:
               322.00px, height: 440.00px}, good matches are
               212.
        Product 11.jpg (ChocoKrave) - Instance found:
            Instance 1 {position: (464.50, 695.50), width:
               313.00px, height: 399.00px}, good matches are
               81.
        Product 26.jpg (Nesquik duo con regalo fucsia) - Instance
            found:
            Instance 1 {position: (204.50, 222.00), width:
               345.00px, height: 444.00px}, good matches are
               174.
        Product 25.jpg (CocoPops palline rosso con regalo) -
           Instance found:
            Instance 1 {position: (530.50, 281.00), width:
               367.00px, height: 562.00px}, good matches are
               160.
```

Also in this case can see how the system correctly identifies the two slightly different versions of the product. On the other hand, the bounding box of product 25 is not extremely accurate.

5.2 Step B

Among the five proposed scenes, our system was able to correctly identify all of the products within each scene, except for one error that will be shortly discussed.

5.2.1 Scene 1

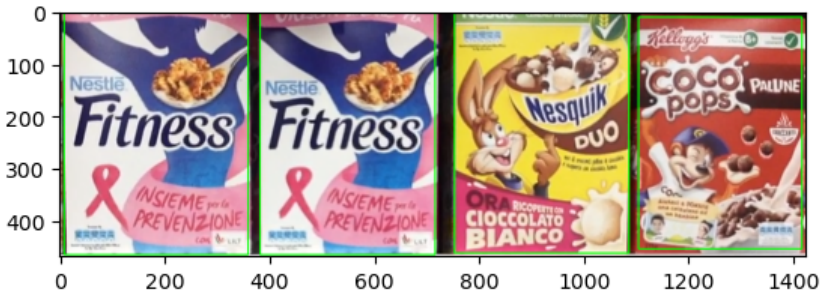

Figure 4: Scene m1

```
Scene m1.png
        Product 25 (CocoPops palline rosso con regalo): 1
           instance found:
            Instance 1 {position: (1261, 232), width: 312px,
               height: 442px
        Product 26 (Nesquik duo con regalo fucsia): 1 instance
           found:
            Instance 1 {position: (920, 230), width: 333px,
               height: 461px
```

```
        Product 24 (Fitness pink edition): 2 instances found:
                Instance 1 {position: (184, 232), width: 351px,
                   height: 464px
                Instance 2 {position: (550, 232), width: 334px,
                   height: 464px
```

5.2.2 Scene 2

Figure 5: Scene m2

```
Scene m2.png
        Product 0 (Nesquik no scritte): 1 instance found:
                Instance 1 {position: (178, 300), width: 343px,
                   height: 448px
        Product 1 (ChocoKrave blu): 2 instances found:
                Instance 1 {position: (1181, 254), width: 302px,
                   height: 409px
                Instance 2 {position: (848, 256), width: 312px,
                   height: 412px
        Product 25 (CocoPops palline rosso con regalo): 1
           instance found:
                Instance 1 {position: (510, 235), width: 281px,
                   height: 316px
```

Here we have the only error of the project: our system was not consistently able to correctly detect product 11 within the scene, misplacing it with product 25. This is likely due to some similarities within the images, such as the brand logo, and similarities in the colors. The error does not occur within every run.

5.2.3 Scene 3

```
Scene m3.png
        Product 26 (Nesquik duo con regalo fucsia): 1 instance
           found:
                Instance 1 {position: (195, 230), width: 354px,
                   height: 459px
        Product 19 (CountryCrisp nuts azzurro): 1 instance found:
```

Figure 6: Scene m3

```
            Instance 1 {position: (1232, 190), width: 296px,
               height: 381px
      Product 25 (CocoPops palline rosso con regalo): 2
         instances found:
            Instance 1 {position: (556, 225), width: 335px,
               height: 450px
            Instance 2 {position: (889, 226), width: 320px,
               height: 443px
```

5.2.4 Scene 4

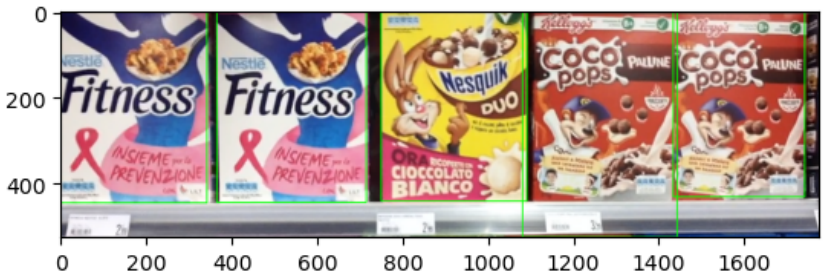

Figure 7: Scene m4

```
Scene m4.png
      Product 25 (CocoPops palline rosso con regalo): 2
         instances found:
            Instance 1 {position: (1263, 263), width: 362px,
               height: 526px
            Instance 2 {position: (1589, 216), width: 308px,
               height: 432px
      Product 26 (Nesquik duo con regalo fucsia): 1 instance
         found:
            Instance 1 {position: (922, 221), width: 341px,
               height: 442px
      Product 24 (Fitness pink edition): 2 instances found:
            Instance 1 {position: (172, 223), width: 343px,
               height: 446px
```

```
                Instance 2 {position: (541, 222), width: 350px,
                   height: 445px
```

5.2.5 Scene 5

Figure 8: Scene m5

```
Scene m5.png
        Product 1 (ChocoKrave blu): 2 instances found:
                Instance 1 {position: (832, 718), width: 302px,
                   height: 411px
                Instance 2 {position: (500, 720), width: 314px,
                   height: 411px
        Product 19 (CountryCrisp nuts azzurro): 1 instance found:
                Instance 1 {position: (908, 191), width: 297px,
                   height: 382px
        Product 25 (CocoPops palline rosso con regalo): 2
           instances found:
                Instance 1 {position: (564, 228), width: 321px,
                   height: 448px
                Instance 2 {position: (240, 313), width: 391px,
                   height: 620px
        Product 11 (ChocoKrave): 1 instance found:
                Instance 1 {position: (162, 716), width: 309px,
                   height: 401px
```

In this case, similarly to what happens in 5.1.3, the bounding box for product 25 is overestimated, but the system is still capable of correctly identify all of the products.

6 Conclusion

The project has achieved its primary objective of accurately identifying and localizing products and their instances within various scenes. The results are not only satisfactory but also demonstrate a high level of compliance with the project's targeted goals. The system's ability to correctly identify products in the majority of test cases is a testament to the effectiveness of the methodologies employed, including the modified Generalized Hough Transform and the innovative use of local invariant features.

The success of this project lays a solid foundation for future developments in this field and provides valuable insights into the challenges and possibilities of object detection and localization in complex environments.

6.1 Further improvements

While the current system has shown impressive performance, there are areas where further improvements can enhance its effectiveness and efficiency.

- **Refining Overlapping Region Handling**: A significant area for improvement is in the management of overlapping regions of interest. The current method, which somewhat relies on a "first come first served" basis for identifying products, can be optimized. A more sophisticated approach would involve assessing the level of confidence for each product in the overlapping regions. By implementing a system that prioritizes regions based on the confidence scores of detections, rather than the order of identification, the accuracy and reliability of the product localization can be significantly improved. This refinement would ensure that the most probable detections are prioritized, reducing the likelihood of misidentification in cases of overlap.

- **Extension to Additional Steps**: Another avenue for enhancement is the expansion into the third step of the project, which was not covered in the current scope but holds promise for further advancements.

In conclusion, while the project has reached a positive result in product identification and instance localization, the potential for further improvements and expansions presents opportunities for future research and development in this domain.

References

[1] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1150–1157, Ieee, 1999.

[2] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, pp. 381–395, 1981.

[3] "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 714–725, 1987.