

---

# **Product Recognition on Store Shelves project report**

Step A-B

Marco Scaramuzzi

2025-09-20

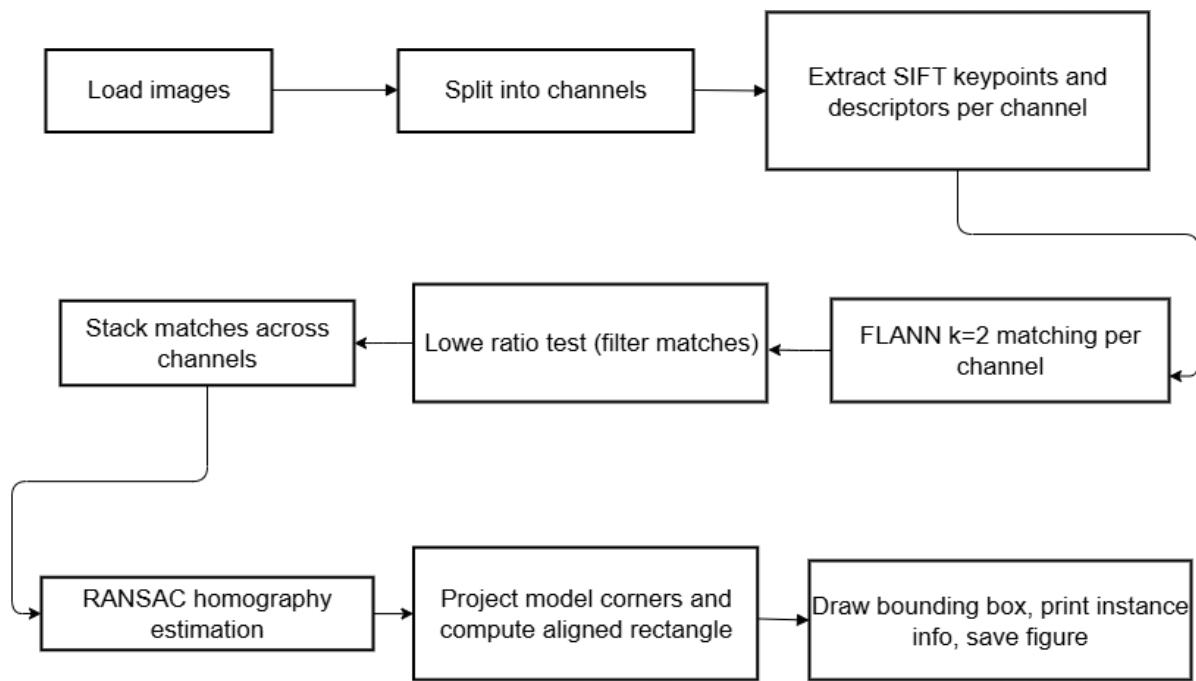
## 0.1 Introduction

This project implements a single-instance product detector that localizes cereal boxes in shelf images using SIFT features computed independently on the RGB channels, FLANN matching with Lowe's ratio test, and homography estimation with RANSAC. The notebook [step\\_A.ipynb](#) contains the runnable pipeline; this report highlights the workflow, key data structures, concise results (extracted from the notebook output), and a short conclusion.

## 0.2 Project overview

The project consists of a pipeline for detecting products on store shelves using computer vision techniques. The main steps include loading images, extracting features, matching them, and estimating the position of products in the scene.

The processing pipeline is compactly represented below.



**Figure 1:** Workflow representation

## 0.3 Approach and key data structures

The approach can be summarized as follows: compute per-channel SIFT features, match per-channel with FLANN + ratio test, stack good matches to compute a robust homography, then derive an axis-

aligned bounding rectangle from the projected model corners.

Data structures (representative):

```
1  keypoints_dict = {  
2      image_id: {  
3          'R': [cv2.KeyPoint, ...],  
4          'G': [cv2.KeyPoint, ...],  
5          'B': [cv2.KeyPoint, ...]  
6      }  
7  }  
8  
9  descriptors_dict = {  
10     image_id: {  
11         'R': np.ndarray(shape=(N_R, 128), dtype=np.float32),  
12         'G': np.ndarray(shape=(N_G, 128), dtype=np.float32),  
13         'B': np.ndarray(shape=(N_B, 128), dtype=np.float32)  
14     }  
15  }  
16  
17 matches_per_channel = {  
18     'R': [cv2.DMatch, ...],  
19     'G': [cv2.DMatch, ...],  
20     'B': [cv2.DMatch, ...]  
21 }  
22  
23 # homography: `3x3` NumPy array  
24 # projected corners: `4x1x2` `float32` array
```

Notes: model images were resized consistently before feature extraction; features are computed once and reused across model-vs-scene comparisons.

## 0.4 Results (parsed from notebook output)

Below are the detections produced by `main(min_count=75, ...)` for the five scenes. For each scene I include only products that had 1 instance (the single-instance constraint). The instance lines are verbatim from the notebook prints. The saved visualization for each scene is included below the textual results.

- Scene 1 ([figures/scene\\_1\\_bounding\\_boxes.png](#))
  - Product 0 - 1 instance found:
    - \* Instance 1 {position: (162, 216), width: 309px, height: 432px}
  - Product 11 - 1 instance found:
    - \* Instance 1 {position: (444, 180), width: 299px, height: 359px}



**Figure 2:** Scene 1 bounding boxes

- Scene 2 ([figures/scene\\_2\\_bounding\\_boxes.png](#))
  - Product 24 - 1 instance found:
    - \* Instance 1 {position: (167, 232), width: 334px, height: 464px}
  - Product 25 - 1 instance found:
    - \* Instance 1 {position: (878, 232), width: 311px, height: 440px}
  - Product 26 - 1 instance found:
    - \* Instance 1 {position: (538, 230), width: 332px, height: 461px}



**Figure 3:** Scene 2 bounding boxes

- Scene 3 ([figures/scene\\_3\\_bounding\\_boxes.png](#))
  - Product 0 - 1 instance found:
    - \* Instance 1 {position: (172, 234), width: 323px, height: 436px}
  - Product 1 - 1 instance found:
    - \* Instance 1 {position: (816, 197), width: 303px, height: 394px}
  - Product 11 - 1 instance found:
    - \* Instance 1 {position: (476, 192), width: 303px, height: 385px}



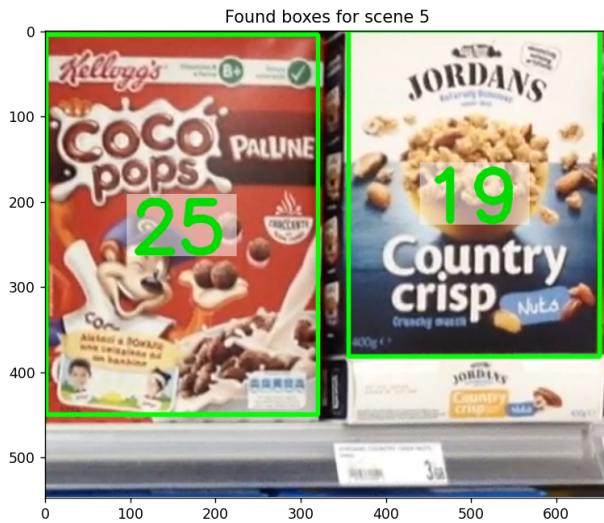
**Figure 4:** Scene 3 bounding boxes

- Scene 4 ([figures/scene\\_4\\_bounding\\_boxes.png](#))
  - Product 0 - 1 instance found:
    - \* Instance 1 {position: (160, 738), width: 320px, height: 435px}
  - Product 11 - 1 instance found:
    - \* Instance 1 {position: (463, 690), width: 304px, height: 396px}
  - Product 25 - 1 instance found:
    - \* Instance 1 {position: (554, 216), width: 318px, height: 433px}
  - Product 26 - 1 instance found:
    - \* Instance 1 {position: (206, 221), width: 341px, height: 442px}



**Figure 5:** Scene 4 bounding boxes

- Scene 5 ([figures/scene\\_5\\_bounding\\_boxes.png](#))
  - Product 19 - 1 instance found:
    - \* Instance 1 {position: (504, 192), width: 295px, height: 383px}
  - Product 25 - 1 instance found:
    - \* Instance 1 {position: (161, 228), width: 320px, height: 445px}



**Figure 6:** Scene 5 bounding boxes

## 0.5 Conclusions

Summary: the SIFT-per-channel + FLANN + RANSAC pipeline reliably localizes single instances of the provided product models in the five test scenes. The saved [figures/scene\\_\\*.png](#) images contain the annotated outputs used to verify detections.

Next steps: extend to multi-instance detection, add quantitative metrics (precision/recall), or move to a learned detector for greater robustness under heavy occlusion.