Product Recognition on Store Shelves: Image Processing and Computer Vision Project Work

Nicola Carrassi - ID: 0001037813 - nicola.carrassi@studio.unibo.it

March 23, 2023

Abstract

In this report, the task of object detection has been developed in order to detect products on the shelves of a store. For this specific work, the samples images on which the solution has been tested are images of cereal boxes. The results obtained are promising, since the object detector correctly identified all the products in the shelves in both settings in which is has been tested.

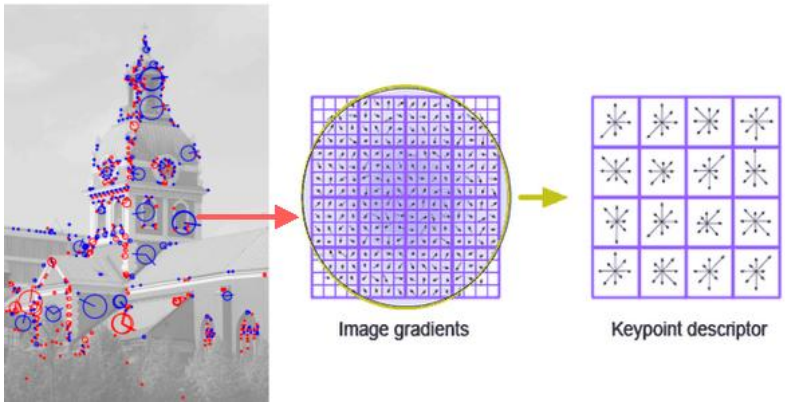
1 Introduction

Object recognition of products on store shelves is a crucial task for various applications in the field of computer vision, including retail automation, inventory management, and customer behavior analysis. With the advancement of deep learning techniques, object recognition has achieved significant progress in recent years. However, recognizing products on store shelves is still a challenging problem due to several factors such as occlusion, clutter, and lighting variations.
In this paper, we propose an approach based on the local invariant features paradigm for object recognition of products on store shelves. Local invariant features refer to distinctive features that can be extracted from an image and are invariant to transformations such as rotation, scaling, and illumination changes. Our approach uses these features to recognize products in a robust and efficient manner. In fact, the solution showed to perform well in both single instance and multiple instances detection.

2 Background

Object detection is a sub field of computer vision that deals with detecting objects of interest within an image or video. It involves finding the location and size of one or more objects in an image. Object detection has numerous real-world applications, including self-driving cars, surveillance systems, and robotics.
Object detection is a challenging task due to several factors such as occlusion, varying lighting conditions, and variations in object appearance and size. There are two primary approaches to object detection: traditional computer vision techniques and deep learning-based methods. Object detection is a challenging task due to several factors such as occlusion, varying lighting conditions, and variations in object appearance and size. There are two primary approaches to object detection: traditional computer vision techniques and deep learning-based methods.
Among traditional computer vision techniques, a common approach consists into following the Local Invariant Features Paradigm. With this approach it is possible to identify object in a scene from a single model image per object. This approach is invariant to scale and rotation and allows to work under partial occlusion. The problems of this approach are that it suffers from changes in camera viewpoint and does not work well with deformable objects. The first step of this process involves the detection of keypoints in both the scene image and the model image, followed by the description of the keypoints, the matching and the position estimation.
In order to detect and describe the keypoints, a very common approach is based on Lowe's work. His proposal relies on Lindberg's idea of scale normalized Laplacian of Gaussian but in a computationally efficient way. He proposed to detect a keypoint seeking for the extrema of Difference of Gaussians (DoG). This approach is a computationally efficient approximation of Lindberg's idea. The detections are rotation invariant and find blob-like features. The DoG images are computed by computing differences between two consecutive images in a octave. Given a stack of DoG images then a keypoint is an extrema with respect to its 26 neighbors (8 at the same scale, 9 at the upper

scale and 9 at the lower scale).
The Scale-Invariant Feature Transform (SIFT) descriptor is then computed taking into account the gradient orientation hisotgram of 8 bins in a 16x16 oriented pixel grid, divided into 16 regions of size 4x4. Each pixel in the region contributes to its bin according to the gradient magnitude and to the gaussian weighting function centered at the keypoint with σ equal to half the grid size. The descriptor is thus computed using the contribution of each gradient direction in each region. The SIFT descriptor is inspired by how neurons in the primary visual cortex match gradient orientations. The descriptors are then normalized to gain invariance with respect to affine intensity changes.



To do the matching step, a kd-tree (short for k-dimensional tree) is used. A kd-tree is a data structure used for efficient nearest neighbor searches in a k-dimensional space. It is a binary tree that recursively partitions the space into subspaces by splitting the data points along the median of a particular dimension at each level of the tree. The result is a tree in which each node represents a subspace of the original space, and the points in the subspace are divided between the two child nodes. In the last step a homography is found using the RANSAC algorithm. This algorithm allows to estimate the homography discarding the matches which are not correct from the good ones.

3 Description of the solution

The proposed object retrieval algorithm allows to detect one or multiple instances of an object into a target scene. The first step is the detection and description of the keypoints. These two steps have been done using the *SIFT detector object* of OpenCV. After that, the image has been divided into different clusters, using the *MeanShift* algorithm to cluster the keypoints. This step was done to simplify the identification process as now different instances of the object can be part of different clusters.
After the clustering, the matching step is done finding the two most similar points using FLANN. FLANN (Fast Library for Approximate Nearest Neighbors) is a library for performing fast approximate nearest neighbor searches in high-dimensional spaces, which is often used in computer vision applications such as object recognition and image matching. It is part of the OpenCV (Open Source Computer Vision) library, which is a popular library for computer vision tasks. A match is considered a good match if and only the distance ratio of the two nearest matches is below a threshold, as proposed by Lowe. The threshold was set to 0.55, requiring the best match to be almost twice as close as the second best match.

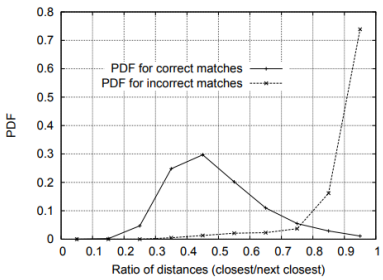


Figure 11: The probability that a match is correct can be determined by taking the ratio of distance from the closest neighbor to the distance of the second closest. Using a database of 40,000 keypoints, the solid line shows the PDF of this ratio for correct matches, while the dotted line is for matches that were incorrect.

If the number of good matches then is above a manually defined value, then we move to the last step which is the position estimation. In this phase, the method *findHomography* is used and then we obtain the region of interest in the image. Since there may be some products which are similar but differ for the color of the box, before considering the match a good match, a last filtering based on the color of the region of interest in the model and in the scene is performed, otherwise the system may consider two boxes with different color as the same.

4 Experimental setup and results

In this section the results obtained both in the case of single and multiple instance detection are reported.

4.1 Step A: Single Instance Detection

In the first experiment the system was tested in the case of recognition of a single instance of an object inside an image. For this task we were given a set of object we need to find and a set of scenes which contain at most one instance of the given object in the scene, such as the following:

Figure 1: Single Instance detection sample image

In this case, with some tuning of the quantile needed to perform the clustering, we managed to obtain a really good result. The system not only correctly found all the matches without making any false detection, but it also performed a very precise detection as we can see from this image

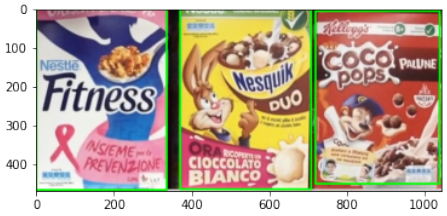


Figure 2: Result on the sample image

4.2 Step B: Multiple Instance Detection

In this step, the scenes became more complex since the problem moved from a binary problem of the kind detect if the model is in the scene or not to a situation in which we can have more than one single instance of each given object as we can see from this sample image.

Figure 3: Multiple Instance detection sample image

Also in this case, with some tuning of the parameters, the proposed solution performed well on the task of detecting all the products correctly.

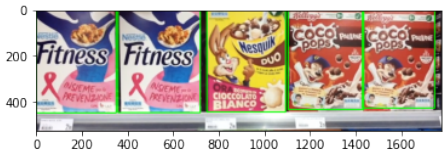


Figure 4: Result on the sample image

5 Conclusion

In this report a simple but effective approach for product recognition on store shelves has been deployed. This approach allowed to detect correctly the instances of the objects in the shelves. This approach showed good performances but, when tested on more complex scenes, it failed to detect the instances. A future work can be improving the solution, making it more robust also in more complex scenes.