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

Chiara Malizia

The main task of the project is to develop an object detection system that can be deployed in supermarket scenarios in order to recognize products on store shelves. Given a reference image for each item in the shelf, the system should identify boxes of cereals of different brands from one picture of a store shelf.

Step A - Multiple Product Detection

The first step consists in developing an object detection system able to identify a single instance of different products.
The task is achieved by using SIFT descriptor and a FLANN-based matcher using kd-tree to perform the nearest neighbour search.
As the models are the same for different scenes, the keypoints and the descriptor of each model are computed once for all and stored. In this way it is not necessary to recreate them every time to compare the two images for matches and homography.
Once models are loaded, the keypoints and descriptors are computed for each scene and matched with those of each model. The found matches are then filtered to store all the good ones as per Lowe's ratio test. At this point if enough good matches are found, they are used to extract the keypoints in both the model image and the training image to find a homography using RANSAC.
After these steps, the perspective distortion of the original object into the image scene is calculated so that to highlight the sought object in the scene by finding its corners and its barycentre and drawing the bounding box. The height and the width of the box are seen as the average length of the two corresponding sides.
This system has however some drawbacks concerning its ability to distinguish very similar products. Indeed, if two products are very similar it considers both their instances, which in turn get overlapped in the scene image.

To solve this problem detected instances at the same position of previously found products are ignored: the barycentre of the sought image in the scene is compared to those of the already detected objects. In other words, if they are too close, the object instance is not considered.
Obviously by so doing the result will depend on the order in which the models are evaluated. A further improvement could be thus to sort the order in which the model images are given. Despite all, the system continues to confuse similar boxes (e.g. Kellogg's Krave cereals), perhaps because the model provided is not the same as that found in the scenes, as shown in the pictures below.

Figure 1: Reference images of two different types of Kellogg's Krave cereals.

Figure 2: Scene image in which cereals boxes are different from those of the model images.

Step B – Multiple Instance Detection

The second task is to create a system able to detect multiple instances of the same product in each of the proposed scenes.
To realize such a system, local invariant feature together with the Generalized Hough Transform are used.
As in the case of the task A, the model keypoints are computed and stored, and the scene keypoints and the model keypoints are matched and then filtered using Lowe's ratio test. Unlike before, alongside the model keypoints, also the joining vectors are computed by choosing the centre of the model image as the reference point.

Then the voting process begins: for each model image the joining vectors are applied to the corresponding scene keypoints, so as to cast a vote for the position of the barycentre by taking into account only the scale transformation between the two images. Indeed, given that in the scene the model appears always upright but with different scales, scale invariance is required for the considered system, while rotation invariance can be ignored. The estimated hypothesis for the barycentre have slightly different values (that is, they are not aggregated in a single point), even if they are consistent. Therefore, for each model Mean Shift clustering provided by Scikit-learn is applied to the estimated barycentre positions to group points belonging to the same object instance.

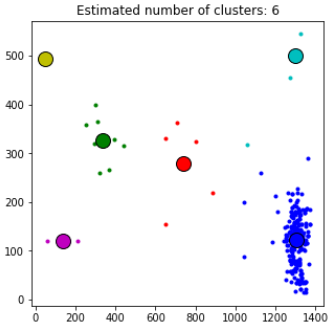

Figure 3: Example of clusters obtained using Mean Shift clustering.

The choice of Mean Shift clustering is due to the fact that no fixed number of clusters needs to be specified (by so doing, this system can be exploited in the task C too).

If the number of points composing a cluster is higher than a chosen threshold, a new instance object is created. The matches related to the consider cluster are used to find a homography using RANSAC, height and width of the object are computed and the bounding box is drawn. However, an improvement is required given that votes for different instances could have been clustered together, as displayed in the figure below.

Figure 4: On the left a plot of clusters where votes for two different instances are clustered together. On the right a shelf picture in which Neste Fitness boxes correspond to the instances clustered together.

The problem can be solved by considering the fact that RANSAC uses only the most fitting matches of the cluster: the process is repeated again employing the discarded keypoints to find other instances of the same object.
Like to the system developed in the task A, this system fails to detect very similar objects. To overcome this limitation, the barycentre of the sought image in the scene is compared to those of the already detected objects, so as to reject overlapped instances.
An additional constraint finally checks the rectangular-like shape of the detected object, allowing thus to decrease the minimum number of matched keypoints in order to consider a cluster as valid.

Step C (optional) - Whole shelve challenge

In this part the goal is to detect as much products as possible in a more challenging scenario, where low-resolution scene images contain distractors elements and many different products instances for each picture.
The system developed in the task B can be again used, but some parameters should be changed in order to take into account that products appear smaller and even slightly blurred in the scene image. For instance, the minimum number of good matches becomes 6 (rather than 290) since less good matches are now found. In addition, the distance between barycenters of two nearby products is reduced and consequently the threshold determining whether two instances are overlapped should be reduced too.
Moreover, the model image is resized to make it smaller before extracting SIFT features in order to increment the possibility to have more good matches between query and train images.

Final Considerations

In each of the previous steps, similar objects are mistaken and some products are missed. As already pointed out, this is probably done to the fact that the model provided is not the same as that found in the scenes: some differ by a little advertisement in a corner (as in the case of Kellogg's Krave cereals), others are quite different because of the huge amount of ads applied on them (Kellogg's Fitness). Furthermore, low resolution of the scene images and blurred pictures certainly affect the performances of the system.

However, if two products are very similar, but they appear in the scene exactly as they are shown in the model (e.g. Jordans Country crisp cereals), the system can correctly detect them. Finally, in the whole shelves challenge some products are detected as present although no model image is provided. In such a case, the product in the target image is considered the one that is the most similar between the possible models, as in the case of Kellogg's Corn Flakes instances in the image scene 5.

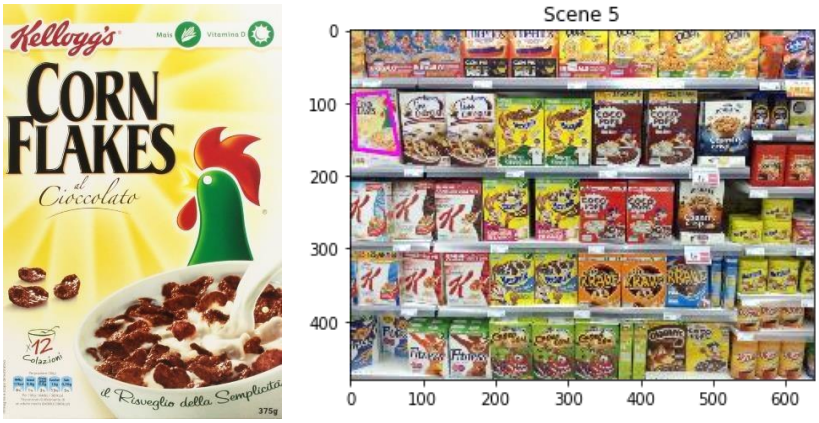

Figure 6: On the left the model image of the chocolate Kellogg's Corn Flakes. On the right a shelf picture in which a box of original Kellogg's Corn Flakes is wrongly detected as the chocolate one.