# Object Recognition - Practical I

Author: Erol Kazancli

## 1. Introduction

In this practical work, Find-Object application downloaded from the website http://introlab.github.io/find-object/ is used to test SIFT and SURF descriptors and compare their performances.

## 2. The Effect of SIFT Parameters

**Contrast Threshold:** This threshold is used to filter out weak features in low-contrast regions. The contrast threshold set by the application was 0.040. When this threshold was increased, the number of features detected decreased and this led to a degrading performance in the instance recognition performance especially in the cases of 3D viewing changes. This performance degradation can be seen from the figure 2. In the picture on the right the cat was not detected.

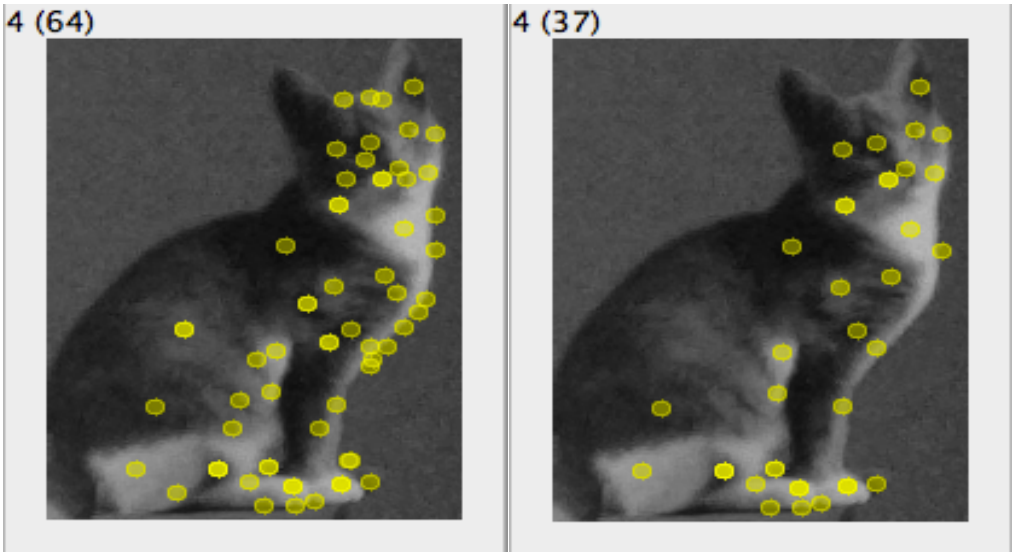


Figure 1 - SIFT with Contrast Threshold 0.040 and 0.060

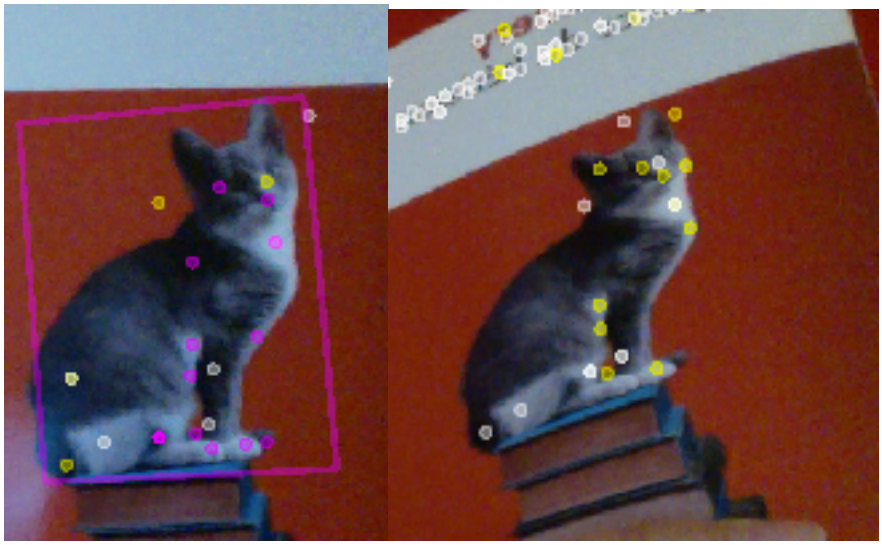

Figure 2 - SIFT with Contrast Threshold 0.080

However when we decreased the contrast threshold to 0.010 this increased the number of features, adding many false or redundant features as can be seen from the figure 3. This is because many low contrast regions, which are due to noise, now pass the filter. In my case this led to the algorithm slowing down due to the increased number of calculations and the detection performance degraded due to many false features.

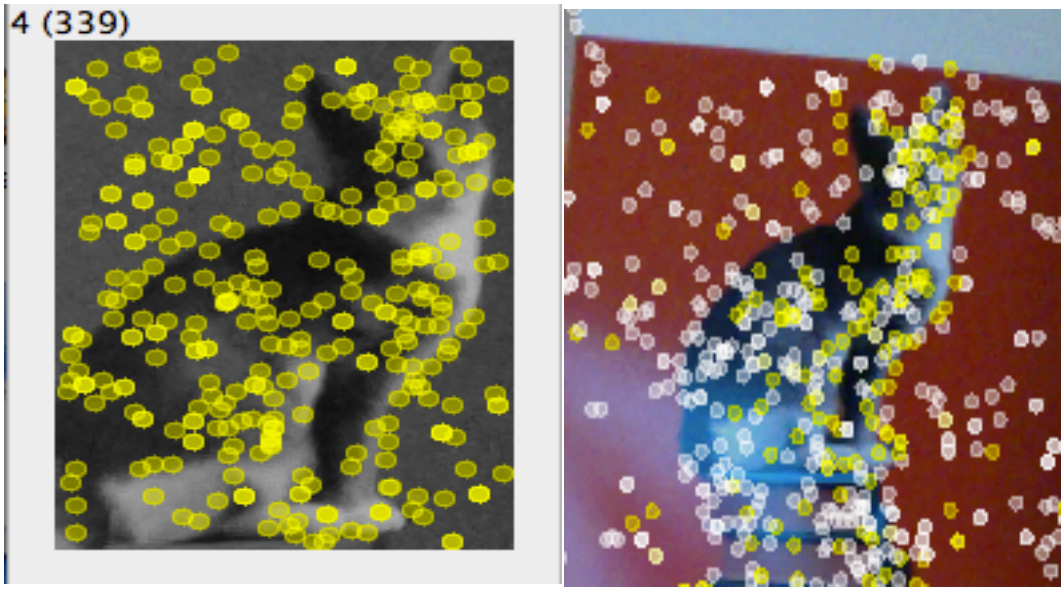


Figure 3 - SIFT with Contrast Threshold 0.010 and failed detection

**Edge Threshold:** Edge threshold is used in rejecting some edges as features. The default edge threshold used by the application was 10. When the edge threshold is decreased the number of features detected decreased. This led to a decrease in detection rates especially in cases of rotation or 3D viewing transformations due to the loss of valuable features. When increased from 10 to 20, the performance did not change.

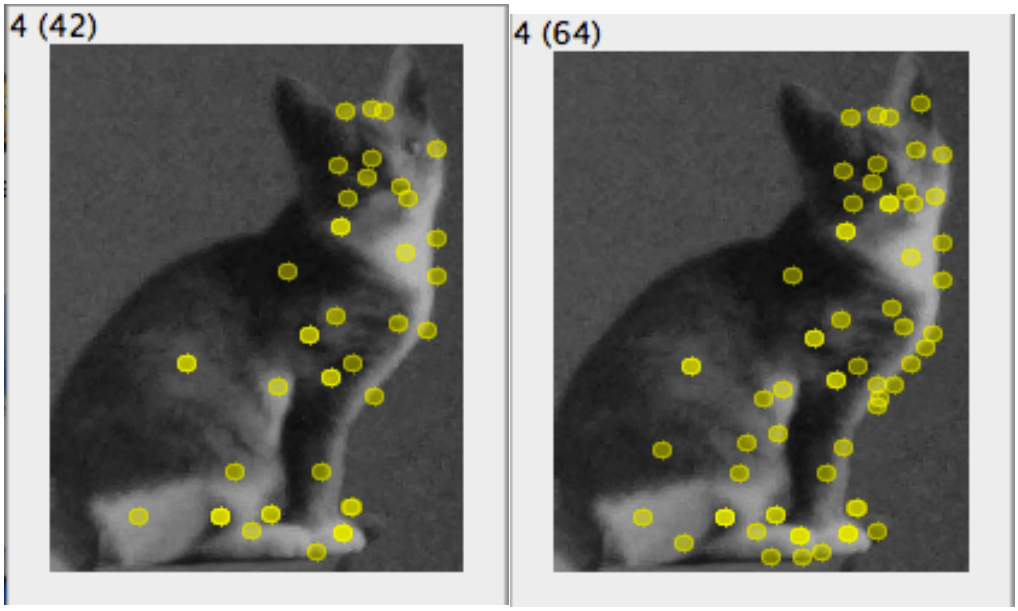


Figure 4 - SIFT with edge thresholds 4 and 10

**Octave Layers:** Octave layers are the layers used in each octave. This number is automatically set by the application according to the image resolution. For the object I used, the number of layers set by the program was 3. When increased to 8, the algorithm slowed down significantly because of the increased number of DoG calculations and increased number of detected features. The detection performance has not improved as a result of the increased number of octave layers, probably due to the fact that the algorithm already calculates the optimum number of layers for a specific resolution and therefore additional blurring did not bring new useful features. We can see from the figures that the number of detected features increased when the number of layers is increased but we can also see bad feature detections among the features.

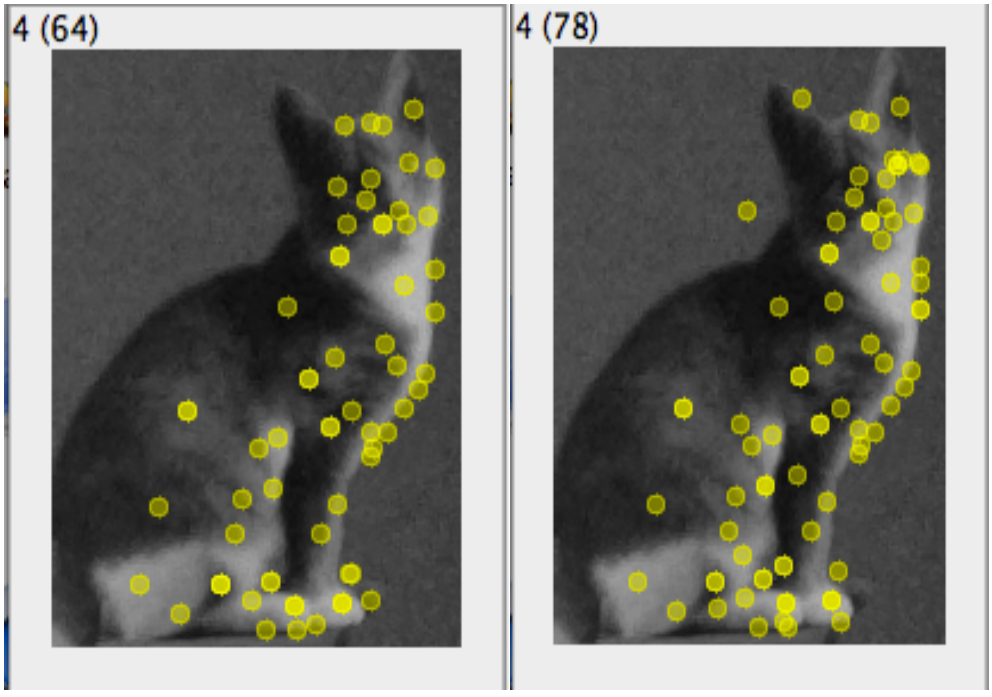


Figure 5 - SIFT with 3 and 8 Octave Layers

**Sigma Value:** The sigma is the parameter used by the Gaussian filter. The sigma value chosen by the application was 1.6. As the sigma value is decreased from 1,6 to 0,5 the performance of detection dropped in spite of an increasing number of features. This might be due to bad features added to the feature set. Interestingly the detection performance also dropped as it increased from 1,6 to 5, since in this case the number of features dropped and some valuable features were lost. Therefore we can say that there must be an optimal value of sigma for a specific object or resolution. In addition the algorithm slowed down as the sigma increased.

## 3. The Effect of SURF Parameters

**Octave Layers:** The number of Octave Layers set by the application was 2 and increasing this number did not bring additional performance gain in detection. Moreover, the computational performance degraded due to increased number of calculations and added unnecessary features.

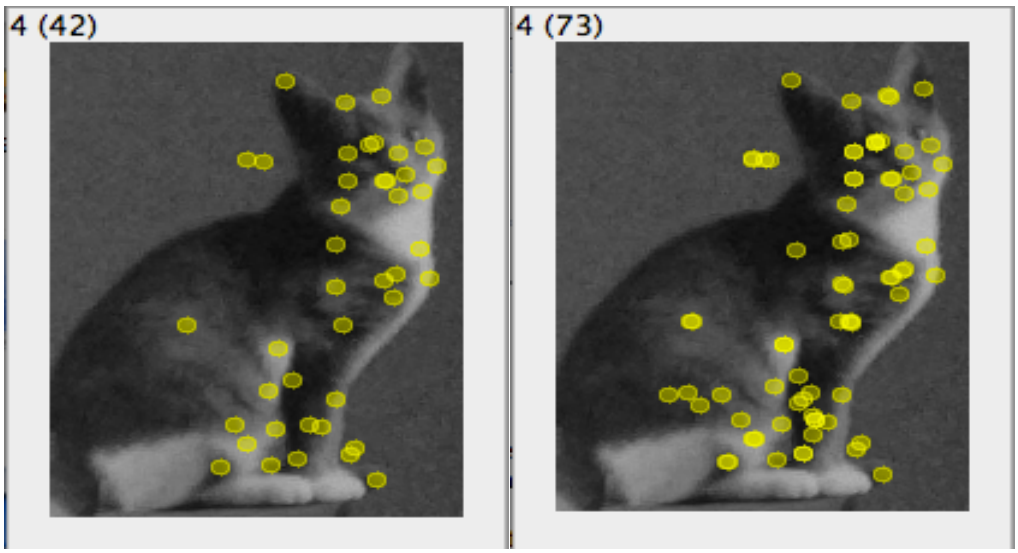


Figure 6 - SURF with Octave Layers 2 and 5

**Hessian Threshold:** Hessian threshold determines which feature to be accepted according to the output of the Hessian Filter. The hessian threshold used by the application was 600. When this threshold is decreased to 200 the number of features detected increased which did not bring additional detection improvement and worsened the computational performance. When this number is increased to 800 the features detected decreased but this did not reduce the detection rate possibly due to the fact that the best features were still kept.

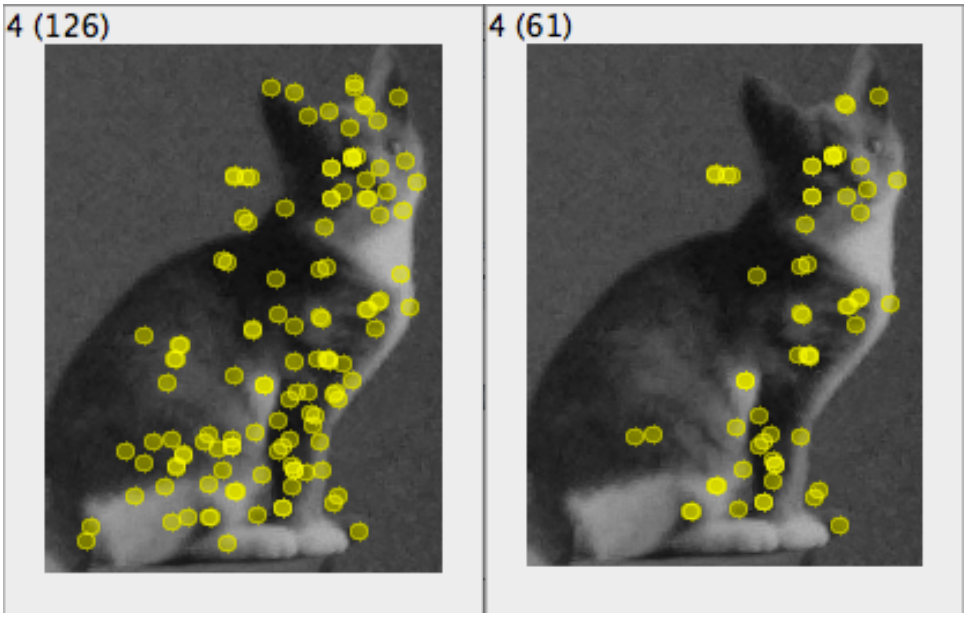


Figure 7 - SURF with Hessian Threshold 200 and 800

**Upright Flag:** When the upright flag was checked the rotation invariance is omitted and therefore the detection was not successful if the object is not in its original orientation. This meant a substantial degradation in instance recognition. This feature can be useful when the camera rotates only in its vertical axis.

## 5. SIFT and SURF Comparison

SIFT and SURF were equally good detectors with respect to scale and rotation changes. However SIFT slightly outperformed SURF when the object went through 3D viewpoint change. This difference can be seen from figures 8 and 9. SIFT was also slightly better in illumination changes, which can be seen from figure 10 and 11. On the other hand SURF outperformed SIFT in terms of computational efficiency, which is due to integral images and fewer dimensional features used in SURF.

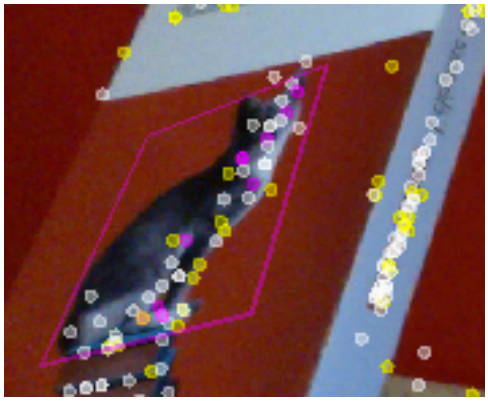

Figure 8 - SIFT with 3D Viewpoint Change

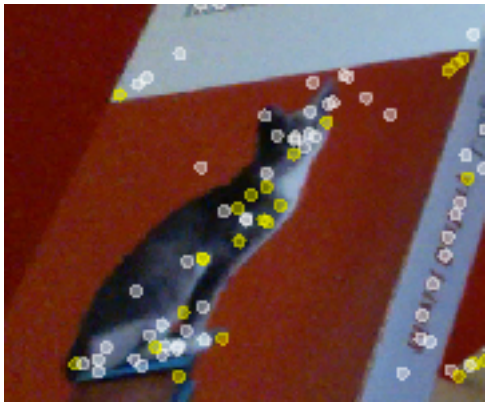

Figure 9 - SURF with 3D Viewpoint Change(not detected)

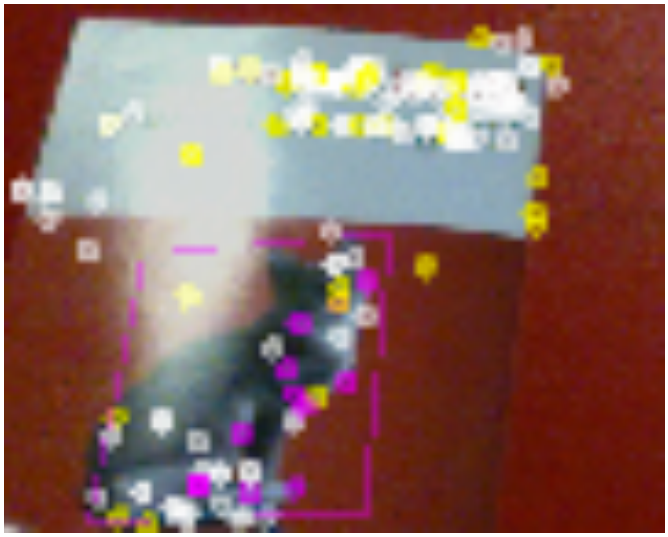

Figure 10 -SIFT with illumination change

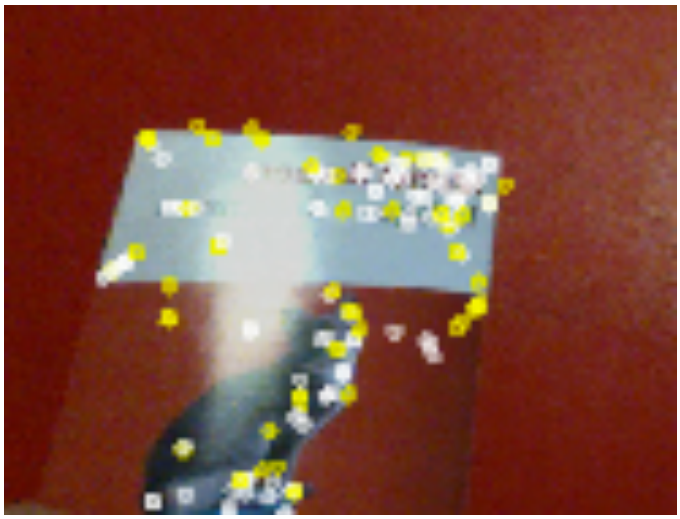

Figure 11 - SURF with Illumination Change(not detected)

# 6. Analysis of Heuristic Approaches and RANSAC

The features are chosen to be highly distinctive and invariant to change in illumination, scale and rotation. High dimensional feature vectors are used for more accurate matching. Along with the principal orientation, orientations within 80 percent of the highest peak is also kept and used for peak interpolation, which leads to higher accuracy and better matching.

Another heuristic used is the nearest neighbor algorithm used for matching. Histograms of gradients(feature vectors) for each descriptors are compared to find the closest matching descriptor using the Euclidean Distance. The closest distance is also compared to the second closest distance, which is assumed to be an incorrect match, and if the closest distance is not significantly closer in comparison than this match is discarded as a false match. Hough transform, which is based on voting of clusters of features on object pose, is also used to increase the matching accuracy.

To be able to make more correct matches, instead of averaging methods like least mean squares, a method named RANSAC is used, which is more appropriate where accuracy is quite important and there is a significant number of gross errors. RANSAC is based on selecting a subset of points to form a model and checking all the points to see the level of agreement on this estimated model. If the agreement is above a certain threshold the model is kept, if not another subset is chosen to find the best or a good enough model. This method is quite effective in object recognition, since there will be many false feature matches which need to be eliminated and RANSAC algorithm helps remove these errors by testing many subsets until it finds a good enough representative set of features.

# 7. Conclusion

These findings may not be representative of the general performances of these descriptors and may be due to the chosen object or the chosen parameters. But in general, using the parameters set by the application and making multiple trials, SIFT performed slightly better than SURF in recognition with this specific instance of an object although it was much slower.

**References:**


. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, November 2004.

. Bay, H. and Ess, A. and Tuytelaars, T. and Van Gool, L. Speeded-Up Robust Features (SURF), Computer Vision and Image Understanding, June 2008.

. Fischler, M.A. and Bolles, R.C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM. June 1981.