An evaluation of local interest regions for non-rigid object class recognition

Oğuz Altun*, Songül Albayrak
Yıldız Technical University, Istanbul, Turkey

ARTICLE INFO

Keywords:
Non-rigid object class recognition
Local Interest Region
EdgeLap
SURF
HarLap
HarAff
HesLap
HesAff
kAS
FAST
IBR
PCBR
Salient
MSER
DoG
ExpRand
Discriminancy

ABSTRACT

Non-rigid object class recognition is a challenging computer vision problem. Using descriptors extracted from local interest regions has important advantages like robustness to occlusion and photometric effects. In this work we compare different local interest region detectors for non-rigid object class recognition through the success-rate of a Generalized Hough Transform based recognition system and a database of 29 non-rigid object classes. The results of the experiments show that the Edge–Laplace (Mikolajczyk, Leibe, & Schiele, 2006; Mikolajczyk, Zisserman, & Schmid, 2003) interest region detector leads. We also evaluate interest regions based on a novel discriminancy measure we introduce. This measure compares success-rates of detectors to success-rates of our novel random region generator, ExpRand. By this respect, ExpRand attain success-rate on par with best detector, and is more discriminant than most detectors.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

An important and challenging application area of computer vision is the non-rigid object class recognition problem. Among the different approaches, using descriptors extracted from local interest regions is a promising one, as recognition algorithms that use them are robust to problems like occlusion and photometric effects.

When building an object class recognition system that uses interest regions, one needs to choose among different interest region detectors. This paper aims to help reader in making such a choice by evaluating object-class recognition performance of 12 different interest region detectors.

The evaluation is performed on our Turkish Fingerspelling database (Altun & Albayrak, 2006), a database of 29 non-rigid object classes. The recognition algorithm used is the Generalized Hough Transform (Beinglass & Wolfson, 1991; Lowe, 2004), an algorithm that makes extensive use of relative region locations.

Seven of the detectors we evaluate (EdgeLap Mikolajczyk et al., 2003, 2006, SURF (Bay, Ess, Tuytelaars, & Gool, 2008), HarAff (Mikolajczyk & Schmid, 2004), HesAff (Mikolajczyk, Leibe, & Schiele, 2005), kAS (Ferrari, Tuytelaars, & Gool, 2006; Ferrari, Favre, Jurie, & Schmid, 2008; Martin, Fowlkes, & Malik, 2004), FAST (Rosten & Drummond, 2005, 2006), IBR (Tuytelaars & Gool, 2004; Tuytelaars & Van Gool, 2000), and PCBR (Hongli, Wei, Eric, Thomas, & Linda, 2007) were not evaluated for object class recognition before, to the best of our knowledge. For the detectors evaluated before (HarLap Mikolajczyk & Schmid, 2001, 2005, HesLap (Lowe, 2004; Mikolajczyk & Schmid, 2004, 2005), Salient (Kadir, Brady, & Zisserman, 2000), MSER (Matas, Chum, Martin, & Pajdla, 2002; Mikolajczyk, Tuytelaars et al., 2005), DoG (Lowe, 2004), we come up with a somewhat different ranking. The reason for this difference probably lies on the different natures of our database and recognition method.

EdgeLap comes up as the detector with the best recognition success-rate, but does so by using large numbers of regions per image. SURF and DoG regions follow closely with second and third best success-rates, and do so by using relatively small number of regions.

To analyse the region output behavior of detectors, we report average number of regions each detector produces. In general, as the number of regions increase, the recognition success also increases. However, there seems to be an optimum number of regions after which increasing the number of regions does not contribute positively.

We quantify how discriminant regions of a detector are by a novel “discriminancy with respect to random regions” statistic. To calculate it, we generate same number of random regions with the detected regions, and compare their recognition successes.

We notice that recognition success of the blind random region generator used for discriminancy calculations is on par with the
best detector success. In addition, when looked in the light of the
discriminany statistic, random generated regions beat all but
EdgeLap, SURF, and DoG regions! Adding the fact that they can
be generated as many as wanted independent of the image data-
base, very rapidly, we conclude that the random generated regions
actually be good choices for a recognition system.

Section 2 of this work reviews related work on evaluating inter-
est region detectors. Section 3 gives references and short descrip-
tions of the methods we compare. Section 4 explains the
discriminany criterion we introduce. Section 5 introduces our no-
vel random region generator, ExpRand. Section 6 explains the data-
base and protocol we use in our experiments. Section 7 discusses
the success rate equation we use, and the rationale behind it. Sec-
tion 8 discusses the experiment results. Finally, Section 9 and 10
gives a conclusion and comments on the future research directions.

2. Related work

There are extensive evaluations of interest regions for feature
matching (Mikolajczyk & Schmid, 2005; Mikolajczyk, Tuytelaars et al.,
2005; Moreels & Perona, 2007), however, this is not true for
evaluations for object class recognition. Below is a short discus-
sion of the related work we know.

Seemann, Leibe, Mikolajczyk, and Schiele (2005) evaluate Harris
(Harris & Stephens, 1988; Schmid & Mohr, 1997), Harris–Laplace
(Mikolajczyk & Schmid, 2001), Hessian–Laplace (Mikolajczyk &
Schmid, 2005) and DoG (Lowe, 2004) region detectors in combina-
tion with local descriptors and global Chamfer matching (Gavrila,
2000) for pedestrian detection. Detection step for the local ap-
proach is done using the Implicit Shape Model (Leibe & Schiele,
2003). They report that Hessian–Laplace leads as the best region
detector, choosing the right detector makes a big difference, more
regions results in better detection results, and local descriptors
based detection results beat global Chamfer matching results.

Mikolajczyk, Leibe et al. (2005) evaluate Harris–Laplace
(Mikolajczyk & Schmid, 2004), Hessian–Laplace (Mikolajczyk,
Tuytelaars et al., 2005), DoG (Lowe, 2004), Salient (Kadir & Brady,
2001) and MSER (Matas et al., 2002) region detectors and three
descriptors for object class recognition. Since "many state-of-the-
art approaches for object class recognition use clustering of local
features as an intermediate level of representation"(Mikolajczyk
yk, Leibe et al., 2005) they evaluate the quality of interest-region clus-
ters, and then verify their results on a pedestrian detection system.
They reach to a new ranking of detectors than Mikolajczyk and
Schmid’s previous evaluation (Mikolajczyk & Schmid, 2004) which
was not based on object class recognition. This time Hessian–
Laplace comes up as best, followed by Salient regions. They also
report that Hessian–Laplace and Salient regions complement each
other well, since the number of shared clusters between them is
low, compared to other pairs. They also report that MSER does
not perform well as it provides too few regions.

Stark and Schiele (2007) evaluate Harris–Laplace (Mikolajczyk
& Schmid, 2005), Hessian–Laplace (Mikolajczyk & Schmid, 2005)
and Salient (Kadir, Zisserman, & Brady, 2004) regions in combina-
tion with different shape and appearance based descriptors using
recognition methods Naïve Bayes and Localized Bag-of-Words.
They also evaluate quality of interest-region clusters as in Miko-
lajcz yk, Leibe et al. (2005). Since their emphasis is on comparing
shape based detectors against appearance based descriptors, they
use a database of 10 classes suited for that purpose. They also
use 10 selected classes of Caltech-101 (Li, Fergus, & Perona, 2004)
database. Mostly the ranking goes as Hessian–Laplace, Harris–
Laplace and Salient regions. They report that for the kind of
database they use, using location information of the local features
affects performance more than detector or descriptor selection.

Our work differs from these works in that we evaluate more re-
gion-detectors from each of them, we evaluate methods that were
not evaluated based on object class recognition before independ-
ently (EdgeLap, HarAff, HesAff, IBR, and PCBR) and we evaluate
methods that were not evaluated at all before independently (kAS, SURF and FAST), to the best of our knowledge. We reached
to a somewhat different ranking of detectors, too. The reason of
this different ranking may be the different nature of our database
or recognition algorithm, or it may be the fact that we maximized
the number of regions detected by using appropriate settings,
when possible. E.g. by default MSER implementation provides too
few regions to be useful for object category recognition, but its set-
tings can be altered to provide many more regions. After this opti-
mization MSER beat Hessian–Laplace, a method that it loses
consistently in other recognition evaluations. It can be suspected
that if the implementations of methods like IBR and PCBR had set-
tings that make them supply more regions, they could also have a
better recognition success rate.

Nowak, Jurie, and Triggs (2006) compare Laplacian-of-Gaussian
(Lindeberg, 1993) and Harris–Laplace (Lazebnik, Schmid, & Ponce,
2003) with a random region producer based on image classification
using a Bag-of-Features approach. Random regions are selected from
a scale pyramid where each scale plane is divided by regular grids in
space and each grid is equally likely to be selected. As a result there
are more regions from lower scales. They report that the single most
important factor that affects the success of classification is the num-
ber of regions. And since region-detectors can not produce as many
regions as random region generator, they lose to it. A more surpris-
ing result is that with high number of regions the region-detectors
do not produce more discriminant regions, i.e. when they produce
close number of regions the random region producer still leads.

Our work resembles Nowak et al. in that we also compare region-
detectors with a random region generator. The differences being
that we evaluate many more region-detectors than that work, we
produce random regions with a different method, and we propose
an analytic measure of region quality based on random regions.
Another difference is that we use a Hough Transform based recogni-
tion approach in which the relative localization of the regions are
heavily used, where they use a Bag-of-Features based recognition
approach in which the relative localization of the regions are ignored.
They make their evaluation out of six different databases, where we
only use our own database. Also theirs is an image classification
approach, ours is a nonrigid object class recognition application. Still,
we mostly confirm their result about success-rate and region count:
as the number of regions increase, recognition success-rate increase
as well. However, we notice that there is a saturation point, e.g. a point
in the number of regions where adding more regions does not affect
success-rate positively. We also confirm that most region-detectors
are less discriminant than a random region generator that produces
the same number of regions.

3. The compared interest region detectors

We tried to make evaluation range as broad as possible, and did
not eliminate any interest region detector that we could find a
working implementation. Below are small description and refer-
ces to each detector.

regions are determined by the points in the scale space on which both
Harris function (Harris & Stephens, 1988) in space and the Laplacian-
of-Gaussian (Lindeberg, 1998) in scale attain maxima.

HarAff2 (Mikolajczyk & Schmid, 2004): Harris–Affine regions are
found by first finding Harris–Laplace regions, then applying an af-

1 http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/
extract_features2.tar.gz
fine adaptation process (Lindeberg & Garding, 1997) based on the second moment matrix.

HesLap$^1$ (Lowe, 2004; Mikolajczyk & Schmid, 2004, 2005): Hessian–Laplace regions are determined by the points in the scale space on which both Hessian determinant in space and Laplacian-of-Gaussian (Lindeberg, 1998) in scale attain maxima.

HesAff$^2$ (Mikolajczyk, Tuytelaars et al., 2005): Hessian-Affine regions are found by starting from Hessian–Laplace regions and applying a second moment matrix based affine adaption process (Lindeberg & Garding, 1997) to each region.

EdgeLap$^3$ (Mikolajczyk et al., 2003, 2006): Edge–Laplace region centers are found by using multi-scale Canny edge detector output. Each point on the edge is a candidate. Points that have a distinctive Laplacian extremum in scale are selected. The scale that gives the Laplacian extremum is also accepted as the scale of the region.

IBR$^4$ (Tuytelaars & Gool, 2004; Tuytelaars & Van Gool, 2000): Intensity Based Regions are found by starting from intensity extrema and studying intensity along rays emanating from them. A sudden change in intensity clues the end of region in that ray direction. Closed regions obtained by linking these end-of-region points are then mapped to ellipses.

Salient$^5$ (Kadir et al., 2000): Salient regions are found by calculating pdf of intensity values in ellipsoidal regions, and then calculating entropy from pdfs. The extrema of the entropy in scale become candidates. Candidates with higher derivative of pdf with respect to scale are selected as salient regions.

MSER$^6$ (Matas et al., 2002; Mikolajczyk, Tuytelaars et al., 2005): Maximally Stable Extremal Regions are connected components detected out of the intensity image through watershed algorithm. Only regions that are stable over a large range of intensity thresholds are selected.

PCBR$^7$ (Hongli et al., 2007): Principal Curvature Based Regions are similar to MSER in that they are connected components detected through watershed algorithm, but they are detected out of the principal curvature images, images obtained by taking maxima of the eigenvalues of the Hessian matrix at each pixel. The stability is seen over the range of scales.

Doc$^8$ (Lowe, 2004): Difference of Gaussian regions are found by detecting points that Laplacian-of-Gaussian yields extrema in both space and scale. Difference-of-Gaussian approximation of Laplacian of Gaussian is used to speed up the process.

kAS$^9$ (Ferrari et al., 2006, 2008; Martin et al., 2004): k Adjacent Segments are set of approximately straight contour segments. k is the number of segments in the set. E.g. single segments are in 1AS, shapes like L are 2AS, and shapes like Z and F are 3AS. Center of a kAS is defined as the average of the segment centers, and the scale of a kAS is the distance between the farthest segments.

SURF$^{10}$ (Bay et al., 2008): Speeded Up Robust Feature constructs are the extrema of Hessian determinant in both scale and space. To extract Hessian determinant rapidly, an approximation based on integral images and box filters is used.

FAST$^{11}$ (Rosten & Drummond, 2005, 2006): Features from Accelerated Segment Test detector is also designed for speed. The interest points are found using a segment test: To decide if a given pixel is a corner/interest point, the pixels on a circle around this pixel is checked. If n contiguous pixels (a segment) are different in brightness from the center pixel by a threshold, the center pixel is decided to be a corner pixel. In the primitive form of the algorithm n is to be larger than 11. Later using machine learning techniques (ID3 Quinlan, 1986) this restriction is removed. To make calculations faster, authors use smart checks to eliminate non-corners early. FAST does not give scale information. Since our recognition algorithm needs scale information to work properly, we extracted Laplacian scale (Lindeberg, 1998; Mikolajczyk & Schmid, 2001) on the locations given by FAST detector.

4. Discriminatory

One obvious criterion for evaluation of different detectors is the recognition success-rate. However, a detector used in an object recognition application must achieve highest success-rate with least number of regions; in other words, the detected regions must be as “discriminant” as possible. This is because usually interest region detection takes place early in the run-time of a recognition system, and if too many regions are produced, the later steps that use these regions would take too much time. Unnecessary regions could decrease the success-rate, too, by confusing the recognition algorithm.

Ideal region detector would detect only “necessary”, or “discriminant” regions, leading to efficient and successful recognition applications.

Since each detector detects a different number of regions in average, and recognition success is affected by number of regions a lot, discriminancy can not be quantified using success–rate and number of average regions. Instead, we define discriminancy with respect to random regions success.

We define discriminancy as

\[ q = \frac{d}{r} \quad (1) \]

where q is the discriminancy, d is the success rate for the detector, and r is the success rate for regions produced by random generator.

We make sure that random generator generates just the same number of regions per image as the detector. As success-rate of random generator may change on each run, we use mean of six runs. Standard deviations of success-rates of our random generator ExpRand can be seen in Table 1.

By definition the discriminancy of random region generator is 1. Discriminancy shows how good a detector is in comparison with a blind random generator. We expect a good detector to at least beat a random generator, hence get a discriminancy higher than 1. In contrary, a discriminancy value less than 1 indicates that the detector is inferior to a blind random region generator; hence its use for this type of application is questionable.

Notice that one should not respect discriminancy values when the number of regions is too little, that is, when the random region generator success-rates are too low or even zero. As random region generator success diminishes faster than detector success with very low number of regions, the discriminancy values of the detectors increase too much. An example of this occurs for IBR and PCBR, which have too few number of regions in average, hence both their and random region generator success-rates are too low.

5. ExpRand: the exponential random region generator

To generate a random region, our novel random region generator ExpRand starts by selecting a random scale s with:

\[ s = s_{\text{min}} + \text{rand} \quad (2) \]

Where \( s_{\text{min}} \) is the minimum acceptable scale and \text{rand} is a random number from the exponential probability distribution given as:

\[ \text{std} \text{deviation} = \frac{1}{\text{mean}} \]
One hundred random scales generated by our random region generator. Scales were sorted before plotting to make their distribution obvious.

**Table 1**

<table>
<thead>
<tr>
<th>Average region count</th>
<th>ExpRand success mean</th>
<th>ExpRand success standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>67</td>
<td>0.0046</td>
<td>0.0056</td>
</tr>
<tr>
<td>325</td>
<td>0.7057</td>
<td>0.0438</td>
</tr>
<tr>
<td>384</td>
<td>0.7575</td>
<td>0.0384</td>
</tr>
<tr>
<td>428</td>
<td>0.8195</td>
<td>0.0452</td>
</tr>
<tr>
<td>444</td>
<td>0.8287</td>
<td>0.0379</td>
</tr>
<tr>
<td>482</td>
<td>0.8322</td>
<td>0.0399</td>
</tr>
<tr>
<td>555</td>
<td>0.8471</td>
<td>0.0450</td>
</tr>
<tr>
<td>862</td>
<td>0.8759</td>
<td>0.0265</td>
</tr>
<tr>
<td>875</td>
<td>0.8575</td>
<td>0.0369</td>
</tr>
<tr>
<td>876</td>
<td>0.8736</td>
<td>0.0307</td>
</tr>
<tr>
<td>1577</td>
<td>0.8897</td>
<td>0.0338</td>
</tr>
<tr>
<td>1658</td>
<td>0.8713</td>
<td>0.0222</td>
</tr>
<tr>
<td>2282</td>
<td>0.8667</td>
<td>0.0113</td>
</tr>
</tbody>
</table>

Even though (2) gives scales within range \( s_{\text{min}} \) to infinity, larger scales have very little probability, and if a scale that is too large to fit in image is produced, it is discarded. We choose \( s_{\text{min}} = 5 \) and \( \mu = 15 \). These are ad hoc values chosen to mimic scale output range of detectors like HesLap and MSER in our database. In the future versions, they will be decided using image dimensions. Fig. 1 show the distribution of scales generated for 100 regions in this way:

ExpRand finishes by generating a random \( x \) coordinate from the uniform probability distribution (Walpole, Myers, & Myers, 2002) and a random \( y \) coordinate again from the uniform probability distribution.

Using this scheme we can generate any number of random regions.

When designing ExpRand we use continuous probability distributions instead of discrete ones, because our recognition algorithm uses real \( s, x, \) and \( y \) values.

6. Database and protocol

Our database consists of 29 classes of non-rigid Turkish Alphabet fingerspellings. Each letter corresponds to a different class. For each class we use six training and five test examples. Fig. 2 shows one example from each class.

7. Success-rate

The output of our object class recognition system can be classified as true, false, and empty, where true indicates correct recognitions, false indicates wrong recognitions, and empty indicates the case when the recognition system outputs no decision for the object. As success rate, we use

\[
\text{SuccessRate} = \frac{t - f}{a}
\]

where \( t \) is the number of true recognitions, \( f \) is the number of false recognitions, and \( a \) is the number of all recognition attempts. We use (4) instead of the usual \( \frac{t}{a} \) because we prefer empty recognitions to false recognitions.

8. Evaluation and results

To evaluate an interest region detector, we use regions detected by it in our Generalized Hough Transform based object class recognition system (Altun & Albayrak, 2011; Beinglass & Wolfson, 1991; Lowe, 2004; Munib, 2007; Youssef & Salem, 2007) and determine the system recognition success-rate by (4). Fig. 3 shows a match by this method. We also note the average number of regions detected per image. As a third step, we run ExpRand random region generator to generate the same number of regions, use generated regions in the same recognition system, and get a success-rate. The success values for ExpRand are obtained averaging values obtained out of six independent runs. Using detector and ExpRand success rates, we obtain the discriminacy value of the detector by (1).

Table 2 lists detector success rates obtained by (4), discriminancy values obtained by (1), and average region counts, sorted by success-rate. We want a good detector to have both a high success rate and a high discriminancy value. EdgeLap, DoG, and SIFT are detectors like this.

From the table we see that EdgeLap and ExpRand with 1577 regions per image lead. It seems they both attain the maximum success-rate this recognition method can achieve on this database and training/test set split. This also makes discriminancy value of the EdgeLap method equal to 1. We leave breaking the tie between ExpRand and EdgeLap as a future work.

Second and third places for success-rate ranking go to SURF and DoG. These detectors shine as they reach very good recognition levels with very low numbers of regions. Low numbers of regions mean faster processing in the following recognition steps. In addition, these detectors themselves were also designed for speed, so they may be preferred when real time processing is needed.

Hessian–Laplace, a detector that leads in other evaluations in the literature, gets a lower rank, as it neither produce enough regions in our database, nor its regions are particularly discriminant. This supports an argument of Nowak et al. (2006): a random generator may be favorable to a detector for recognition, because it can generate as much regions as needed, independent of the nature of the dataset. In contrast, a region detector that produces enough regions for one database may not be able to do so for another database.

MSER, which is known to perform well in feature matching, had low rankings for recognition in other evaluations in literature, yet has a respectable ranking in our work. Default settings for the MSER implementation hinders the detectors capacity for recognition, since by arranging settings to increase number of
regions (555 per image for our database), one obtains much better results.

When considering discriminancy we see that only SURF and DoG and Surf has really high discriminancy values, and the Edgelap is only as discriminant as random regions. We do not consider discriminancy of PCBR and IBR at all; the success of ExpRand at those region-count levels is too low to be comparable.

In general, success-rate increases with increasing number of regions, but only up to an optimal number of regions (1577 in our database). After that number, success-rate starts to decrease. This can be observed for random generator and detectors 4AS and FAST in Table 2, and for ExpRand in Table 1.

At very low number of regions success-rates of detectors (E.g. IBR and PCBR) and random generator diminish quickly, random generator diminishing faster. This again can be observed in Tables 1 and 2.

We did not evaluate the running time of the detectors, as some implementations are written in different languages like C and MATLAB, some were compiled for different operating systems like Microsoft Windows and Linux, and some implementations does extra work while detecting interest regions (e.g. extract descriptors). They also output different number of regions. In summary it is though to make a fair evaluation of running times.

### 9. Summary and conclusions

In this work we evaluate interest region detectors Edgelap, SURF, HarAff, HesAff, kAS, FAST, IBR, PCBR, HarLap, HesLap, Salient, MSER, and DoG based on non-rigid object class recognition success-rate. We use recognition success-rate of a Generalized Hough

![Fig. 2. An example from each class (Turkish Alphabet Letter Fingerspelling) of our database.](image1)

![Fig. 3. Generalized Hough Transform elements shown on a true match. Left image is the query image. Right image bears the matched object. Matched object (shown in green color) is also transformed on to the query image. Circles show interest regions. Matched interest regions are in the same color. Square on the right image show train object reference point. Squares on the left show reference point votes by matched interest regions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)](image2)

<table>
<thead>
<tr>
<th>Detector</th>
<th>Detector success rate</th>
<th>Discriminancy</th>
<th>Average region count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edgelap</td>
<td>0.8897</td>
<td>1.0000</td>
<td>1577</td>
</tr>
<tr>
<td>Surf</td>
<td>0.8690</td>
<td>1.1472</td>
<td>384</td>
</tr>
<tr>
<td>Dog</td>
<td>0.8552</td>
<td><strong>1.2117</strong></td>
<td>325</td>
</tr>
<tr>
<td>Mser</td>
<td>0.8207</td>
<td>0.9688</td>
<td>555</td>
</tr>
<tr>
<td>Haraff</td>
<td>0.8207</td>
<td>0.9370</td>
<td>862</td>
</tr>
<tr>
<td>3As</td>
<td>0.8069</td>
<td>0.9237</td>
<td>876</td>
</tr>
<tr>
<td>Harlap</td>
<td>0.7931</td>
<td>0.9294</td>
<td>875</td>
</tr>
<tr>
<td>Salient</td>
<td>0.7734</td>
<td>0.9425</td>
<td>428</td>
</tr>
<tr>
<td>Fast</td>
<td>0.6897</td>
<td>0.7916</td>
<td>1658</td>
</tr>
<tr>
<td>4As</td>
<td>0.6690</td>
<td>0.7719</td>
<td>2282</td>
</tr>
<tr>
<td>Hesaff</td>
<td>0.6069</td>
<td>0.7323</td>
<td>444</td>
</tr>
<tr>
<td>Heslap</td>
<td>0.5862</td>
<td>0.7044</td>
<td>482</td>
</tr>
<tr>
<td>IbR</td>
<td>0.3734</td>
<td>Inf</td>
<td>37</td>
</tr>
<tr>
<td>PcbR</td>
<td>0.0138</td>
<td>3.0000</td>
<td>67</td>
</tr>
</tbody>
</table>
Transform based object class recognition system trained and tested on a database of 29 non-rigid object classes (fingerspellings). We find that EdgeLap regions achieve best success-rates, followed by SURF and DoG regions. The success of EdgeLap can be attributed to the fact that it detects quite high number of respectably discriminant regions. The success of SURF and DoG can be attributed to the high discriminancy of their regions since they detect relatively low numbers of regions. In order to quantify how discriminant region detectors are, we introduce a novel discriminancy measure that compares detected region success with same number of randomly generated region success.

From Nowak et al. (2006), we already know that as the number of regions increase, the recognition success-rates also increase. However, in this work we observe that there is an optimum number of regions, after which the success rate does not increase. We also observe that random generated regions can be actually used for object class recognition: In addition to the fact that they can be produced very rapidly and as many as wanted for any database, their success-rates are on par with the best detector success-rates, and they are more discriminant than most of the detectors.

10. Future research directions

In an object class recognition system that uses local regions, another important step is extracting descriptors from regions. In this work, we used SIFT descriptor for all experiments. A future research direction is to evaluate recognition success of other descriptors in conjunction with detectors. This is also important because of the fact that some region detectors we evaluate (kAS, SURF) here actually come with their own descriptors.

Another research direction is to evaluate complementarity of detectors to each other. This is important because a combination of detectors may beat any single detector.

This work and Nowak et al. (2006) show that using random generated regions may be actually a good idea for object class recognition. Then a better inspection of random region generators is needed. Some questions to be inspected are: Given an image, how many regions should be generated? What is the best random region generating method? Is it a good idea to combine random generator with detectors? If so, which detector or detectors should one combine?

Acknowledgements

We acknowledge all the authors that put their interest region detectors to the web.

References


Ferrari, V., Tuytelaars, T., & Gool, L. V. (2006). Object detection by contour segment networks, in ECCV.


Li, F.F., Ferguson, R., & Perona, P. (2004). Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories, in CVPR.


Martin, D., Fowlkes, C., & Malik, J. Learning to detect natural image boundaries using local brightness, color, and texture cues, in PAMI.


Mikolajczyk, K., Leibe, B., & Schiele, B. (2006). Multiple object class detection with a generative model, in CVPR.


