Cue Integration by Similarity Rank List Coding — Application to Invariant Object Recognition

Raul Grieben and Rolf P. Würtz
Institut für Neuroinformatik, Ruhr-Universität Bochum,44780 Bochum, Germany
raul.grieben@ini.rub.de, rolf.wuertz@ini.rub.de

***Abstract*—Similarity rank lists provide a method for learning generalization of classifiers from examples. Here, we apply it to invariant object recognition and demonstrate that it performs better than other approaches on view and illumination invariant recognition. Recognition from a single view reaches 87% success rate. To study its real world capabilities we introduce subsqare rank matching that works on image patches and RUBJECTS100, a database of 100 objects under varying pose and illumination, and a set of natural scenes containing these objects.**

I. Introduction

For any classification task in machine learning, the *generalization* of the learned classifier is the most important capability. Ideally, it should cover the whole natural class while rejecting everything outside of it. However, usually little is known about the shape of the true class in a high-dimensional space.

In the case of visual object recognition, deep convolutional neural networks have recently advanced to defining the state of the art [8]. However, with the help of backpropagation it is possible to construct images that are visually indistinguishable from an image of a certain object but classified as another [17], [5]. Such adversarial examples show that even large networks fail to completely model the shape of the object class. Also, with the use of evolutionary algorithms, it is possible to construct images that are classified with high confidence but show no visual similarity to the recognized object [10].

Therefore, it might be fruitful to attempt to actively control the generalization of a classifier by learning how different image instances of the same object are transformed into each other and applying this learned transformation to new objects.

In [13] we have developed *similarity rank list comparison* as such a method, which takes a model set of faces of some persons in different situations and applies the learned transformation to new persons. In combination with elastic bunch graph matching [19] this was able to outperform all previous methods in the literature on the CAS-PEAL database [3] in recognition under varying viewpoint and illumination.

Similarity rank list comparison also provides a natural way of integrating different cues into a single decision by simple averaging of the rank list similarities provided by each cue.

In this study, we apply the technique to the recognition of objects under varying viewing angles and illuminations. We introduce an extension that processes image patches instead of whole images. This is finally tested on images containing the objects in cluttered backgrounds.

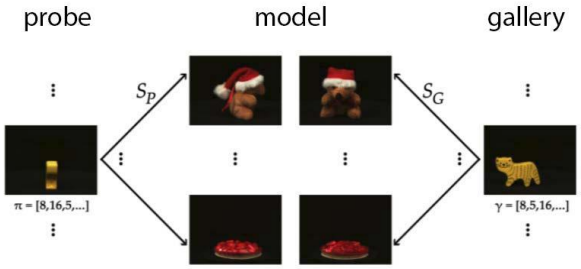

Fig. 1. A model database (center) captures the variations that objects undergo in different situations. A gallery (right) contains all object in a subset of situations. Probe images as well as gallery images are coded as similarity rank lists to the various model images. Recognition is on the basis of rank list similarity

II. Recognition by similarity rank lists

Let $S_c^{\mathrm{im}}(I_1, I_2)$ be any measure of similarity between two images I_1 and I_2, the index c enumerating different image cues.

For the recognition of an arbitrary object a large *gallery* database is created, which contains known objects in certain situations, in the extreme case only one image per object.

Object variation is described by a *model* database containing some objects in all possible situations together with the information which model images belong to the same object.

In most of [13], it was assumed that the situations be estimated beforehand by some other algorithm. The paper also briefly described a version, in which situations were unlabeled, that is the only information in the model database was which set of images belongs to the same person. This leads to much longer rank lists and consequently higher computational demands, but avoids the arduous task of estimating the situation beforehand.

In this work we concentrate on the latter. It is also appropriate for invariant object recognition, because there is no natural definition of what the same pose would be for different objects. Model images are denoted by M_m^v with m enumerating identity and v enumerating the various instances. Similarly, the gallery consists of gallery images G_g^v.

Object identity is coded by a similarity rank list to all models, for both probe and gallery images. The rank list for any image T is created as follows. First, all similarities S^{im} to all model images M_m^v are calculated. A rank list r is created,

978-1-5090-6558-5/17 $31.00 © 2017 IEEE
DOI 10.1109/FAS*W.2017.38

which contains the rank of similarity for each model index m, so that for each pair of model images $M_m^v, M_{m'}^{v'}$ the following holds ($r_c(T,m) \in \{1,\dots N_M\}$):

$$r_c(T,m) < r_c(T,m') \quad \Rightarrow \quad S_c^{\mathrm{im}}(T, M_m^v) \geq S_c^{\mathrm{im}}(T, M_{m'}^{v'})\,. \tag{1}$$

The most similar model candidate would be $r_c(T,1)$, the follower-up $r_c(T,2)$, etc. These lists now serve as a representation of a test image T.

Each subject G_g in the gallery is assigned a rank list representation $\gamma_{g,c}$ by matching each of its landmarks to those of the model subjects in the preferred situation:

$$\gamma_{g,c}(G_g, m) = r_c(G_g, m)\,,\ m = 1\dots N_M\,. \tag{2}$$

For recognition, an unknown probe image P is also represented as a similarity rank list π_c for each cue:

$$\pi_c(P, m) = r_c(P, m)\,,\ m = 1\dots N_M\,. \tag{3}$$

A. Rank list comparison

Having represented the gallery and probe images by rank lists of equal length all that is required for invariant recognition is a function $S_{\mathrm{rank}}(\pi,\gamma)$ that measures the similarity of these rank lists. Such a function should should take values between 0 and 1, be high when many model indices appear at the same rank, and maximal for two identical rank lists. Cooccurrences with high image similarities (i.e., with low values of r) should be weighted more strongly than those with low ones. One example for such a similarity is

$$S^{\mathrm{rank}}(r_1, r_2) = \frac{\sum_{m=1}^{N_M} f(r_1(m) + r_2(m))}{\sum_{m=1}^{N_M} f(2m)}\,, \tag{4}$$

where f is a monotonically decreasing. Here we are using $f(x) = (x+1)^d$ with $d \in [-2, 0)$, like in [11], [12]. Another tested possibility for S^{rank} is the rank-biased overlap function (RBO) developed by [18]. The rank list similarity for a single cue c is

$$S^{\mathrm{rank}}(\pi, \gamma_{g,c}) = \frac{1}{N_M} \sum_{m=1}^{N_M} (\pi(m) + \gamma_g(m) + 1)^d\,. \tag{5}$$

This rank list similarity can be evaluated separately for each cue, and the resulting similarities are averaged over all cues.

$$S^{\mathrm{rec}}(P, G_g) = \frac{1}{N_C} \sum_{c=1}^{C} S^{\mathrm{rank}}(\pi_c, \gamma_{g,c})\,. \tag{6}$$

As usual, the recognized object is the one with the index g that maximizes this similarity

$$g^{\mathrm{rec}} = \arg\max S^{\mathrm{rec}}(P, G_g)\,. \tag{7}$$

B. Features and similarity functions

Color images are denoted as $(I_R(p), I_G(p), I_B(p))$, grey value images as $I_{\mathrm{gray}}$with p running over all pixels. The simplest similarity function comparison of images I, J is the negative mean squared error (**MSE**):

$$S_{\mathrm{MSE}}^{\mathrm{im}}(I,J) = -\sum_p \sum_{k\in\{R,G,B\}} (I_k(p) - J_k(p))^2 \tag{8}$$

Bit code	Code	Cue
10000000	MSE	Mean squared error
01000000	PCC	Pearson correlation coefficient of grey levels
00100000	CHI	Averaged color histogram intersection
00010000	HHI	Hue color histogram intersection
00001000	LBP	Local binary pattern histogram intersection
00000100	GLCM	Gray level cooccurrence matrix features
00000010	SHI	Saturation color histogram intersection
00000001	BHI	Brightness color histogram intersection

TABLE I
THE IMAGE CUES USED IN THIS STUDY. CUE COMBINATIONS ARE CODED BY A BIT STRING WITH ONES AT RESPECTIVE POSITIONS.

The second is the Pearson correlation coefficient on the greyscale (**PCC**)

$$\begin{aligned} \overline{I} &= \sum_p I(p),\ \overline{J} = \sum_p J(p) \\ S_{\mathrm{PCC}}^{\mathrm{im}}(I,J) &= \frac{\sum_p (I(p)-\overline{I})(J(p)-\overline{J})}{\sqrt{\sum_p (I(p)-\overline{I})^2 \sum_p (J(p)-\overline{J})^2}} \end{aligned} \tag{9}$$

Next we are using various histograms, which are all compared by histogram intersection:

$$S^{\mathrm{Hist}} H_1, H_2 = \frac{\sum_i \min(H_1(i), H_2(i))}{\sum_i H_2(i)} \tag{10}$$

Applying this to normalized RGB histograms normalized by the sum of R-, G- and B-channels [16], yields the **CHI** similarity $S_{\mathrm{CHI}}^{\mathrm{im}}$ as the average of the single channel histogram intersections.

Histograms intersections in the HSB (Hue, saturation, brightness) space are denoted by **HHI**, **SHI**, and **BHI**, respectively with similarity functions $S_{\mathrm{HHI}}^{\mathrm{im}}$, $S_{\mathrm{SHI}}^{\mathrm{im}}$, and $S_{\mathrm{BHI}}^{\mathrm{im}}$.

Another useful grey-level image feature is Local Binary Patterns (**LBP**) [15], which can be histogrammed and compared by histogram intersection to yield $S_{\mathrm{LBP}}^{\mathrm{im}}$.

Finally, we are have used several features like difference of contrast, energy, and homogeneity based on *gray level cooccurrence matrices* (**GLCM**) to build the $S_{\mathrm{GLCM}}^{\mathrm{im}}$. As these did not prove useful at all (fig. 2) we skip the details here.

III. PROCESSING OF SEGMENTED OBJECTS

To evaluate the recognition capabilities we used the ALOI (Amsterdam Library of Object Images) [4]. It contains 4 image sets of 1000 objects, only 3 of them are used in this study. The first set consists of a variation of the viewing angle by rotating the object in the plane at 5-degree resolution, resulting in 72 images per object. The second set was taken under variation of the illumination angle archived by either using different combinations of light positions (l1–l8) or camera positions (c1–c3), resulting in 21 images per object. The third set consists of a variation of the illumination color temperature from 2175K to 3075K, with 12 images per object.

For our tests we have cropped all images to the bounding box of the object and rescaled the cropped images to 32 × 32 pixel. We used the last 250 objects as model database. The gallery consists of a single image at 0° viewing angle for each of the remaining 750 objects. Probe images were drawn from all these images.

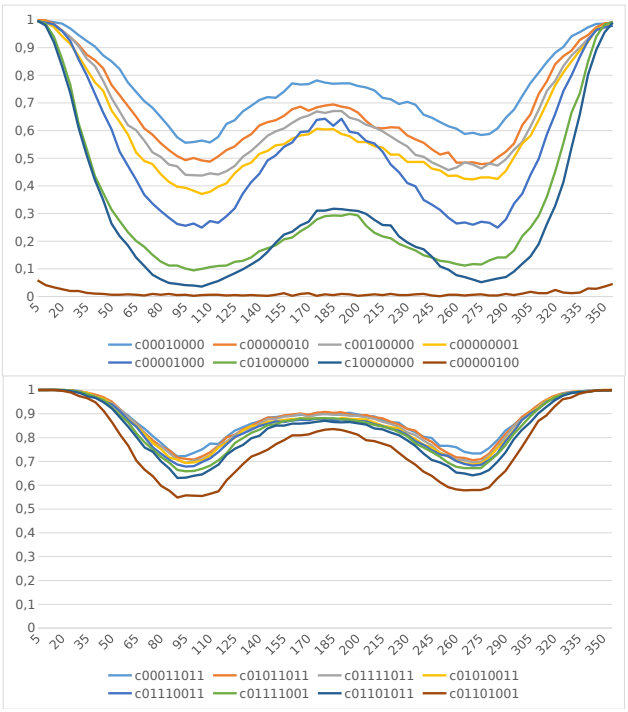

Fig. 2. Recognition rates on ALOI depending on rotation angle. Top: Single similarity functions, Bottom: Selected combinations of similarity functions. The overall recognition rate for c00011011 is 87.5%

Fig. 3. Recognition rates on ALOI depending on illumination direction (top) and illumination type (bottom). The cues for the color evaluation were HHI, LBP, SHI, and BHI), for greyscale PCC, CHI, LBP and BHI.

Method	view	ill. dir.	ill. col.
Proposed System: 750 objects			
Color KS	99.95	98.23	99.91
Greyscale KS	99.95	98.78	**100**
Color US	99.85	**99.75**	99.69
Greyscale US	99.84	99.58	99.96
Color SSR-US	**100**	N/A	N/A
Greyscale SSR-US	99.92	N/A	N/A
Proposed System: 1000 Objects			
Color KS	98.99	—	—
Greyscale KS	95.03	—	—
Other Methods: 1000 Objects			
TCG [9]	98.06	98.73	N/A
HMAX [2]	80.76	83.13	99.04
SalBayes [2]	89.71	75.50	64.79
SIFT [2]	70.95	71.47	89.41

TABLE II
RECOGNITION RATES ON THE ALOI DATABASE WITH A 25% OF VIEWS IN THE GALLERY. THE TESTS OF OUR SYSTEM INCLUDE KNOWN (KS) OR UNKNOWN (US) SITUATION, COLOR OR GREYSCALE, AND RANK LIST BUILDING WITH AND WITHOUT SUBSQUARE RANKING (SSR). IN THE FIRST BLOCK, WE USE 250 ALOI OBJECTS AS MODEL. IN THE SECOND BLOCK, THE MODEL IS REPLACED BY THE COIL-100 DATASET. THE THIRD ONE SHOWS THE RECOGNITION RESULTS OF OTHER SYSTEMS.

The results are shown in fig. 2. Recognition rates are at 100% under 10°, decline slowly until 90°, then go back up to 180°, then the behavior is practically symmetric. It can be seen that the HHI-cue performs best, while the GLCM-cue is worthless. The second part of fig. 2 shows results for combinations of four cues, all of which are better than the best single cue.

Fig. 3 shows that illumination situations are handled very well. Table III presents the orientation-invariant recognition results for different sizes of gallery and model, respectively. It can be concluded that moderate model sizes already lead to good recognition.

Finally, we compare the recognition system with others on the same database (table II). The gallery contains 25% of all views for each object, as this is the setup that the other authors used for testing. As we have used 250 images as model database, we first restricted the gallery to 750 objects. Then we changed the model database to the 100 objects in the COIL-100 database [14] and used a gallery of 1000 objects. Although this is a small model database, results are still higher than the competitors'.

IV. Processing of Cluttered Images

A. Subsquare-Rankings (SSR)

To increase the performance in unknown situations we propose a new version of rank list matching using rankings of subsquares of variable size. Here, both probe and gallery images are divided into multiple equally sized squares, which are then compared only to equivalent counterparts of the model image. Independent from each other, each subsquare votes for an object, and the object with the highest vote count becomes matched. This method has although been applied to the segmented images (table II).

To address the challenges of cluttered images, this needs to be accompanied by decision integration across different

Model objects	50	100	150	200	250	300	350	400	450	500
Gallery objects										
50	0.93	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94
100	0.90	0.93	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94
150	0.87	0.91	0.91	0.92	0.92	0.92	0.93	0.93	0.93	0.93
200	0.86	0.90	0.90	0.91	0.91	0.91	0.92	0.92	0.92	0.93
250	0.84	0.88	0.90	0.90	0.90	0.90	0.90	0.90	0.90	0.92
300	0.83	0.86	0.88	0.89	0.89	0.90	0.90	0.91	0.91	0.92
350	0.82	0.85	0.87	0.88	0.88	0.90	0.90	0.90	0.91	0.91
400	0.81	0.85	0.87	0.88	0.88	0.90	0.90	0.90	0.90	0.91
450	0.81	0.85	0.88	0.89	0.89	0.90	0.90	0.90	0.90	0.90
500	0.81	0.85	0.87	0.87	0.88	0.89	0.90	0.90	0.90	0.90

TABLE III
RECOGNITION RATES ON A SINGLE VIEW OVER VARYING ORIENTATION ON THE ALOI DATABASE DEPENDING ON THE NUMBER OF OBJECTS IN MODEL AND GALLERY SETS, RESPECTIVELY.

Fig. 4. An overview of all objects in the RUBJECTS-100 database

scales. In a first step the input (natural) image is resized so that the longest side of the image measures 100 pixel. Then a multiscale sliding window exhaustive search is performed (algorithm 2) and the resulting probabilities at each pixel are saved for each scale (algorithm 1). At the end we have one probability map per object and per scale, which can be evaluated (algorithm 3) to estimate where the object may be on the image. This consists of the following steps:

1) **Selecting the points of interest** From the probability map the maximum value is determined and afterwards all pixels with superthreshold probability selected as points of interest for the current scale. The lowest scale is ignored in this step since it yields many false positives.
2) **Clustering the points of interest** The POI are then clustered using the Mean Shift Algorithm with a Gaussian Kernel [1] with given radius.
3) **Creating bounding boxes for each cluster** For each cluster a rectangle that envelops all points is calculated. Very small rectangles with an area smaller then a fixed threshold are discarded, since they are considered noise artifacts.
4) **Globally merging intersecting rectangles** The newly found rectangles are merged globally with all intersecting or close by rectangles located in previous iterations or scales. The results are the bounding box of the merged rectangles.

Algorithm 1 Object finding

```
procedure FINDOBJECTS(PScaleMaps,Objs)
 rectangles
 for all obj ∈ Objs do
  scale ← 0
  for all map ∈ PScaleMaps_obj do
   if scale > 0 then
    map ←MAXVALUES(map,0.00116)
    clusters ←GAUSSIANMEANSHIFT(map,3,3)
    tmpRcts ←BUILDRECTANGLES(clusters,0.96)
    mergedRcts ←MERGEINTERSECT(tmpRcts,rectangles)
    rectangles ←MERGENEIGHBOR(rectangles,mergedRcts)
   end if
   scale ← scale + 1
  end for
 end for
 return MERGELOWESTSCALE(rectangles,PScaleMaps)
end procedure
```

5) **Globally merging nearest neighbors** Rectangles whose minimal distance is below a fixed threshold are merged analogy to the previous step. In other words, they are

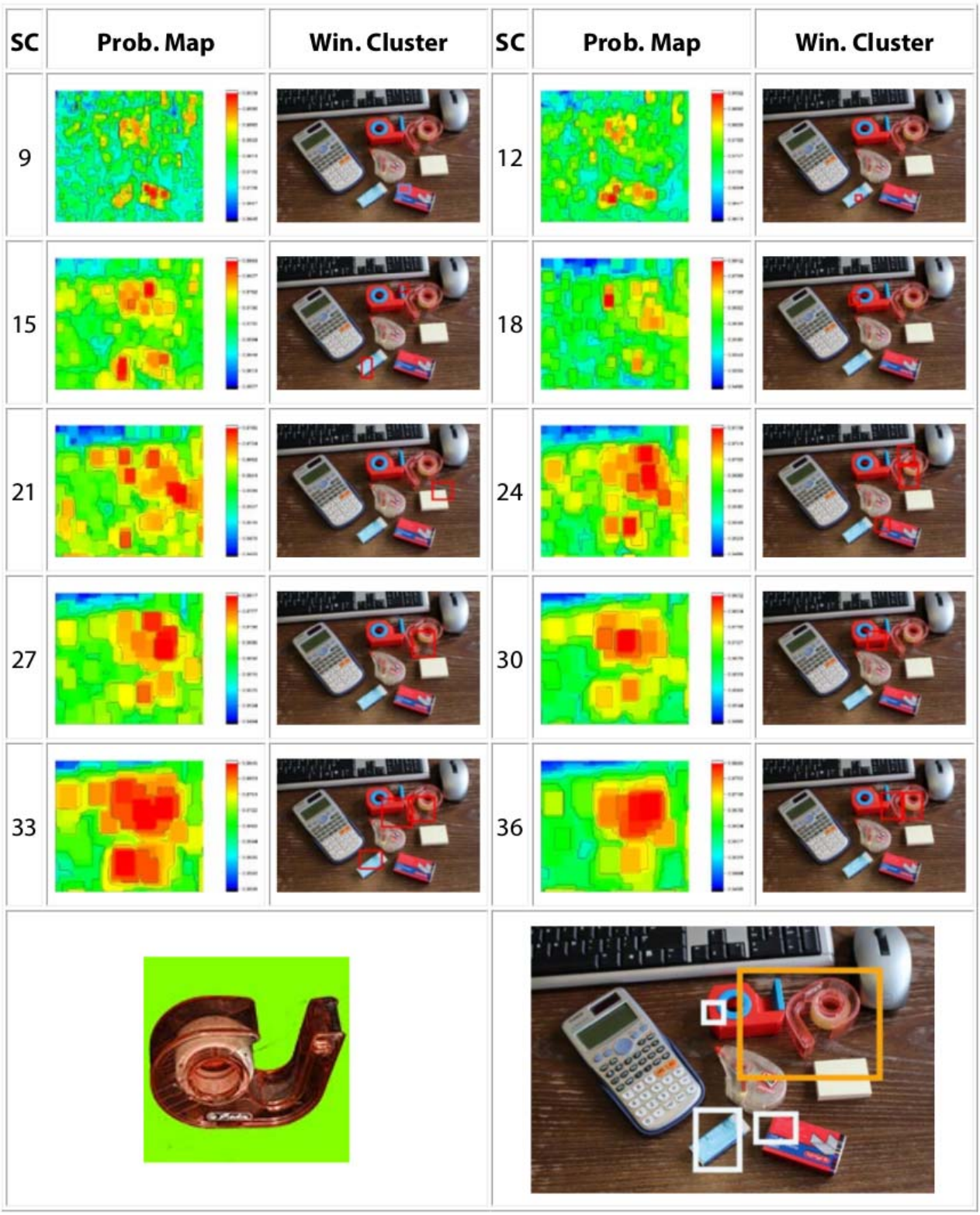

Fig. 5. Object recognition in a cluttered scene.

Algorithm 2 Image preprocessing

```
procedure ProcessImage(IMG,ScaleList,NrOfSquares)
 for all x ∈ IMG_WIDTH do
  for all y ∈ IMG_HEIGHT do
   for all scale ∈ ScaleList do
    WinSize ← scale/NrOfSquares
    WinIMG ← CropImage(IMG,x,y,WinSize,WinSize)
    for s_x ← 0; s_x < NrOfSquares; s_x ← s_x + 1 do
     for s_y ← 0; s_y < NrOfSquares; s_y ← s_y + 1 do
      ProcessWindow(WindowIMG,x,y,scale,s_x,s_y)
     end for
    end for
   end for
  end for
 end for
end procedure
```

Algorithm 3 Window evaluation

```
procedure ProcessWindow(WinIMG,x,y,scale,s_x,s_y)
 PositionList ← GetPositionList()
 FlipList ← GetFlipList()
 RotationList ← GetRotationList()
 for all pos ∈ PositionList do
  for all flip ∈ FlipList do
   for all rot ∈ RotationList do
    ModelList ← GetModelList(pos,flip,rot,scale,s_x,s_y)
    RLCol ← CreateRankListCol(ModelList,WinIMG)
    GalRankLists ← GetGalRankLists(flip,rot,scale,s_x,s_y)
    for all GalRLCol ∈ GalRankLists do
     GID ← GalRLCol_ID
     SYM ← CompareRankListCol(GalRLCol,RLCol)
     SYM_OLD ← GetSym(x,y,GID,scale)
     if SYM > SYM_OLD then
      SaveSym(x,y,GID,scale,SYM)
     end if
    end for
   end for
  end for
 end for
end procedure
```

threaded as intersecting rectangles.

6) **Merging on the lowest scale** Finally, the information of the previously ignored lowest scale is used to expand the final rectangles to their final dimensions.

The generated rectangles are drawn on the original picture in white and the winner rectangle is drawn in orange. Different Methods for deciding the winner rectangle have been tried. The most consistent one was to pick the rectangle with the highest number of merged rectangles. This version was used in the present results of this work.

We did not find realistic cluttered scenes in the available object databases. Therefore, we created our own, called *RUBJECTS100* after our university's acronym. The objects shown in figure 4 have been photographed with a black background and 72 views each.

The described algorithm has shown good results on several severely cluttered scenes containing these objects. We present one example complete with probability maps on all scales in fig. 5. The system also showed good generalization in the presence of noise and distortions. Many more examples can be found in [6].

V. Conclusion

We have presented an object recognition method that learns invariance under various imaging conditions from examples and is capable of reasonable recognition in cluttered real world scenes. Future work will include integrating this into a neuronal system equipped with a robot arm to enable grasping of desired objects [7].

References

[1] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[2] L. Elazary and L. Itti. A bayesian model for efficient visual search and recognition. *Vision Research*, 50(14):1338 – 1352, 2010. Visual Search and Selective Attention.

[3] W. Gao, B. Cao, S. Shan, D. Zhou, X. Zhang, and D. Zhao. The CAS-PEAL large-scale Chinese face database and baseline evaluations. Technical Report JDL-TR-04-FR-001, Joint Research & Development Laboratory for Face Recognition, Chinese Academy of Sciences, 2004.

[4] J. Geusebroek, G. Burghouts, and A. Smeulders. The Amsterdam Library of Object Images. *International Journal of Computer Vision*, 61:103–112, 1 2005.

[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[6] R. Grieben. Visual analysis of natural scenes. M.Sc. thesis, Applied Informatics, Univ. of Bochum, Germany, Sept. 2016.

[7] G. Knips, S. K. U. Zibner, H. Reimann, and G. Schöner. A neural dynamic architecture for reaching and grasping integrates perception and movement generation and enables on-line updating. *Frontiers in Neurorobotics*, 11(9):1–14, 2017.

[8] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[9] M. Lessmann and R. P. Würtz. Learning of invariant object recognition from temporal correlation in a hierarchical network. *Neural Networks*, 54:70–84, 2014.

[10] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016.

[11] M. K. Müller. *Lernen von Identitätserkennung unter Bildvariation*. PhD thesis, Physics Dept., Univ. of Bochum, Germany, July 2010.

[12] M. K. Müller, A. Heinrichs, A. H. Tewes, A. Schäfer, and R. P. Würtz. Similarity rank correlation for face recognition under unenrolled pose. In S.-W. Lee and S. Z. Li, editors, *Advances in Biometrics*, LNCS, pages 67–76. Springer, 2007.

[13] M. K. Müller, M. Tremer, C. Bodenstein, and R. P. Würtz. Learning invariant face recognition from examples. *Neural Networks*, 41:137–146, 2013.

[14] S. Nene, S. Nayar, and H. Murase. Columbia Object Image Library (COIL-100). Technical Report CUCS-006-96, Columbia University, 1996.

[15] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution Gray-Scale and Rotation Invariant Texture Classification With Local Binary Patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002.

[16] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[17] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013.

[18] W. Webber, A. Moffat, and J. Zobel. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):20, 2010.

[19] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.