A Unified System For Object Detection, Texture Recognition, and Context Analysis Based on the Standard Model Feature Set

Stanley Bileschi and Lior Wolf
The Center for Biological and Computational Learning,
Massachusetts Institute of Technology,
Cambridge, MA 02138.
Email: bileschi,liorwolf@mit.edu.

Abstract

Recently, a neuroscience inspired set of visual features was introduced. It was shown that this representation facilitates better performance than state-of-the-art vision systems for object recognition in cluttered and unsegmented images.

In this paper, we investigate whether these features can be applied outside the scope of unsegmented object detection. We show that this outstanding performance extends to shape-based object detection in the usual windowing framework, to amorphous object detection as a texture classification task, and finally to context understanding

These tasks are performed on a large set of images which were collected as a benchmark for the problem of scene understanding. The final system is able to reliably identify cars, pedestrians, bikes, sky, road, buildings and trees in a diverse set of images.

1. Introduction

The Standard Model Features (SMF), recently introduced in [23], are based on combining the output of Gabor filters over scale and position. This combination is done using a max operation[1], resulting in a set of features which are position- and scale-tolerant edge-pattern detectors. The SMF were introduced as an implementation of the feed-forward model in neuroscience, and are successors to previous quantifications of it [19].

The goal of this paper is to reclaim these SMF features from neuroscience by putting them within the context of other common computer vision tasks, and to present a simple and complete system for object detection of a wide variety of object types.

Object recognition using an unsegmented training set is becoming a popular research field, with several benchmark data sets and many published contributions. The first methods [29, 10, 9] used generative models that recognize highly informative object components and their spatial relations, but there have been discriminative approaches [6, 23] and registration based approaches [1] applied to this problem as well.

[1] similar to the morphological dilation operation, but the maximum is taken over scale as well as position

The ability to learn from unsegmented images is impressive, but the performance is still behind that of those systems which do not train in clutter. To detect objects in natural images requires high recall at very low false positive rates. Today there exist several systems which perform well on this difficult benchmark for faces [24, 22, 27], cars [22] and pedestrians [17, 28]. Typically, these systems require large sets of training data.

There is an open question whether the systems that were designed to work on unsegmented images can become successful object detection systems when trained on images with no clutter. If so, then a large gain can be made if they could transform their ability to learn from few training examples, as well as their suitability to a large number of objects, into this new domain. In this paper, we will show that SMF features can be used successfully for object detection in the segmented object framework.

Another recognition task to which we extend the SMF set is the detection of non-shape-based objects, i.e., trees and buildings. This is essentially a texture recognition task: after segmenting the images, we recognize the texture of each segment. We demonstrate that the SMF features outperform other state-of-the-art algorithms. Finally, we offer a platform for context computation inside the same unified framework.

The three capabilities, shape-based object detection, texture-based object detection and context computation, form a complete system that serves as a robust base for scene understanding architectures.

2. The SMF features

The Standard Model feature set is composed of two sets of features: an intermediate set of features (C1), and the position invariant set of features (C2). It is believed that the biological counterparts of both sets play a role in object recognition in the brain.

The set of intermediate features: This set corresponds to the first cortical stages of V1. It is implemented as the output of a hierarchical process containing two layers termed S1 and C1. The first layer (S1) is obtained by applying a battery of Gabor filters to the image. The parameters of the filters were adjusted so that the S1 units' tuning profiles match those of V1 parafoveal simple cells. This was done by first sampling the space of the parameters and then generating a large number of filters. These filters were applied to stimuli which are commonly used to assess V1 neurons' tuning properties [14] (i.e., gratings, bars and edges). After removing filters that were incompatible with biological cells [14], we were left with a final set of 16 filters at 4 orientations (see table 1). The S1 layer therefore contains 16×4 filter output images.

The next layer, C1, corresponds to complex cells which show some tolerance to shift and size. This tolerance is obtained by taking a maximum across neighboring scales and nearby pixels. For this purpose, the 16 filters were divided into into 8 *bands*. The output of each band Σ is determined by max-filtering each filter-response over a region of size $N^{\Sigma} \times N^{\Sigma}$, and taking the maximum again over the scales within the band. This process is done separately for every orientation. The output of the C1 layer therefore contains 4 orientations times 8 bands for a total of 32 different images of combined filter outputs.

The position- and scale-invariant set of features: This computation can also be conceptualized as two layers, the S2 layer and the C2 layer. The S2 layer employs a patch based approach, wherein each band of the C1 output is filtered with a set of prototypes. These prototype patches are themselves crops of images represented in C1 space. This process can be described as a template matching process where each prototype is com-

Band Σ	1	2	3	4	5	6	7	8
filter scale s	7 & 9	11 & 13	15 & 17	19 & 21	23 & 25	27 & 29	31 & 33	35 & 37
Gabor width σ	2.8 & 3.6	4.5 & 5.4	6.3 & 7.3	8.2 & 9.2	10.2 & 11.3	12.3 & 13.4	14.6 & 15.8	17.0 & 18.2
Gabor wavelength λ	3.5 & 4.6	5.6 & 6.8	7.9 & 9.1	10.3 & 11.5	12.7 & 14.1	15.4 & 16.8	18.2 & 19.7	21.2 & 22.8
position pooling size N^{Σ}	8	10	12	14	16	18	20	22
orientation θ	$0; \frac{\pi}{4}; \frac{\pi}{2}; \frac{3\pi}{4}$							
patch size n_i	$4 \times 4; 8 \times 8; 12 \times 12; 16 \times 16$ ($\times 4$ orientations)							

Table 1: Summary of parameters used in our implementation.

pared to every window of matching size in each band. Note that each $n_i \times n_i \times 4$ prototype is originally extracted from one band, but it is compared across bands for scale invariance.

The final set of shift and scale invariant SMFs (C2) contains the global max over all bands and positions of elements in the S2 layer. This is done separately for each prototype, hence the set of C2 features has as many elements as the number of prototypes.

3. C2 features from a computer vision perspective

The first layer, S1, is just an application of Gabor filters [12, 7] to the input image, which is fairly standard and has been used for many computer vision problems [3, 21, 18]. The S2 and C2 layers are an application of a patch based approach, which uses correlation with smaller image crops as its basic building block. Such systems are gaining a lot of popularity, and have been used successfully for texture synthesis [8], super resolution [11], object detection [26, 25] and object-specific image segmentation [2].

The only layer which might seem unorthodox from a computer vision perspective is the C1 layer, in which the outputs of the S1 layer are being maximized locally. While many systems maximize the output of a detector over the entire image, this has been done locally only recently. For part based object detection [26, 25], detectors of each part are learned independently and then applied to regions where the parts are expected to appear. The SMF seem unique in that *general purpose filters* are being maximized over local regions in the image.

In order to explain the utility of C1, we invoke a scale space terminology (see [15] for an overview). Scale space theory was mostly concerned at first with the Gaussian scale space. This scale space has many desirable properties such as separability, linearity, shift invariance, isotropy, homogeneity, and causality. The last property is an important one: causality means that no new level sets are generated by going into coarser scales. A related property is to demand the non-creation of local extrema in coarser scales.

In our application, local maximization is used to move from a fine scale to a coarser scale in order to make the C1 layer invariant to local translations of the edge. As a pseudo scale space, local maximization has some desirable properties: it is separable (one can apply it over the rows and then over the columns), it is shift invariant, and it is homogeneous (applying it repeatedly corresponds to moving into coarser and coarser scales). However, in general, it is not an appropriate scale space. Among other problems, applying it to an image may create new local extrema.

However, in the SMF framework, the local maximum operator is applied to a set of Gabor filtered images, which are a *sparse* representation of the original image. The max scale space is successful in preserving the amplitude of the sparse maxima, whereas the Gaussian scale space smooths them out.

Object	car	pedestrian	bicycle	building	tree	road	sky
Type	shape-based			texture-based			
# of Labeled Examples	5799	1449	209	5067	4932	3400	2562

Table 2: Summary of some of the labeled objects in the StreetScenes Database.

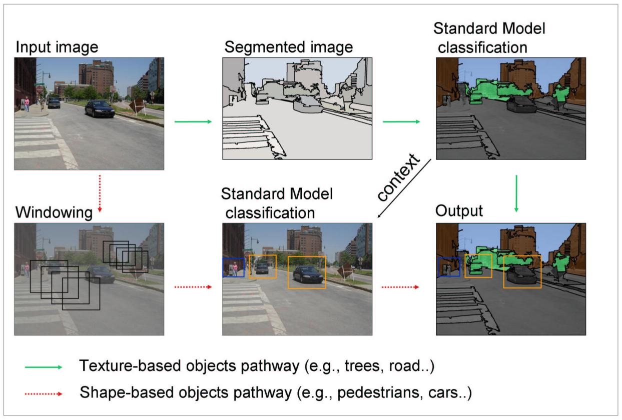
Input image
Segmented image
Standard Model classification
Windowing
Standard Model classification
context
Output
Texture-based objects pathway (e.g., trees, road..)
Shape-based objects pathway (e.g., pedestrians, cars..)

Figure 1: An illustration of the data flow diagram of our system.

4. Scene-Understanding System Architecture

We present the current implementation of a multi-year scene-understanding project. Every detector within this system relies upon the same SMFs, even though the detected objects themselves are qualitatively different. The currently detected objects are listed in table 2.

The objects are divided into two distinct sets, *texture-based* objects and *shape-based* objects, and two classes are handled using different learning strategies. Fig. 1 illustrates the data flow diagram for this architecture, specifically the pathways for detection of the texture-based and shape-based objects. Additionally, the arrow labeled 'context' symbolizes that detections of the texture-based objects are used to aid in the detections of the shape-based objects. Detailed descriptions of the algorithms for texture-based object detection, shape-based object detection, and contextual influence are offered below.

4.1. The Street Scene Dataset

Outdoor images of cities and suburbs was selected as an appropriate setting for the scene-understanding system. A database of nearly $10,000$ high-resolution images has been collected, $3,000$ of which have been hand labeled for 9 object categories. Sample images, their hand labellings, and some empirical results are illustrated in Fig 2. Note that the accurate detection of many of these object categories is made difficult by the wide internal variability in their appearance. For example the object class "cars" includes examples of many diverse models, at many poses, and in various amounts of occlusion and clutter, "trees" appear very different in summer and winter, and the class of "buildings" includes sky-scrapers as well as suburban houses. Capturing this wide variability while

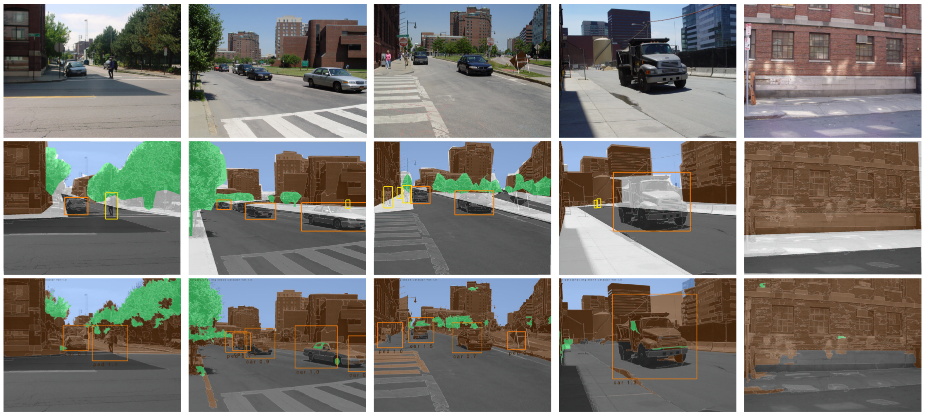

Figure 2: Top Row: StreetScenes examples. Middle Row: True hand-labeling; color overlay indicates texture-based objects and bounding rectangles indicate shape-based objects. Note that pixels may have multiple labels due to overlapping objects. Bottom Row: Empirical performance of the current object detectors.

maintaining high accuracy is part of the challenge of the scene-understanding problem.

4.2. Shape-Based Object Detection

In our system, shape-based objects are those objects for which there exists a strong part-to-part correspondence between examples, including things like pedestrians, cars, and bicycles. In order to detect shape-based objects, the system presented here uses the C1 features from the SMF set in combination with the well-known windowing technique. Windowing is used to enable the detector to recognize objects at all positions and scales, given that C1 features have only limited position and scale invariance.

The training data for these detectors is extracted by cropping examples from a subset of the database set aside for training. These crops are converted into C1 SMF space as detailed in Sec. 2. Briefly, each crop is resized to a common resolution, filtered with directional Gabor wavelets at multiple scales, max-filtered, and finally decimated. In this way, each training example is converted into a $1,024$ dimensional vector, representing a 16×16 square array of C1 level features, each of which is itself a 4 dimensional vector representing 4 different orientations. After both positive and negative examples are extracted, the data are used to train a boosting classifier.

In test images, every square window of the image is converted into C1 space and fed into the object detectors, resulting in a real-valued detection strength at every possible location and scale. The final system output is drawn by thresholding this response and using a local neighborhood suppression technique. In Fig. 2 we presented some typical results of this type of detection.

4.3. Texture-Based Object Detection

Texture-based objects are those objects for which, unlike shape-based objects, there is no obvious visible inter-object part-wise correspondence. These objects are better described by their texture than the geometric structure of reliably detectable parts. For the

StreetScenes database these currently include buildings, roads, trees, and skies.

The detection of the texture-based objects begins with the segmentation of the input-image. For this we employ the freely-available segmentation software "Edison" [5]. Segments are assigned labels by calculating C2 SMFs within each segment, and inputting this vector into a suitably trained boosting classifier. One classifier is trained for each object type using examples from the training database. Note that training samples for the texture objects are only drawn from locations nearer to the center of these objects so as to prevent the classifier from learning anomalous texture responses due to the boundaries between objects.

In our experiments, 444 C2 features are used to represent each texture segment, 111 each from the four possible patch sizes, n_i (see table 1). The associated prototypes are extracted from random locations in the training image database. In order to learn the mapping from this vector of C2 responses to the correct object label, a boosting classifier is employed. Only 150 rounds of boosting are used to learn each model, meaning that for each object, even though 444 C2 features are available, only a maximum of 150 features are actually used.

4.4. Context Detection

In the data flow diagram in Fig. 1, an arrow labeled "context" points from the texture-based object detection unit to the shape-based object detection unit. This arrow indicates that it is possible to use the detection of the texture objects as useful feature inputs to the shape-based objects. The intuition is that, for instance, the detection of roads can and should bias the detection of cars.

In our system, context at a point is defined as a function of the nature of the surrounding objects. The context feature at point x is constructed by sampling the texture-based object detector responses at a set of locations measured relative to point x. These relative locations are spaced such that they well sample the surrounding region while avoiding sampling from any locations which might intersect the actual shape-based object we are building a context model for. Please see Fig. 3 for an illustration of these relative sampling locations in comparison to the average sizes of some of the shape-based objects. A total of 24 such relative locations have been selected, meaning that the feature vector associated with a context is 4×24 dimensional, where 4 is the number of texture-based objects detectable by the system.

In order to train the context detection system, the context feature is sampled from a number of locations of positive and negative object context. A pixel with positive context is defined as a pixel which is within a labeled example of the target object. In the training stage, context feature samples are taken using the true-hand labeled locations of the texture-objects. This training data is used to train a boosting classifier for the context of each shape-based object.

In order to apply the context classifier to a test image, the context feature is first calculated at every pixel. In this case, since true texture-based object locations are unavailable, the empirical detections are used instead. Applying the contextual classifiers to the pixel-wise feature vectors results in one map of contextual support for each of the shape-based object classes. These maps of contextual support are used in a rejection cascade framework, wherein if the support at a particular location is below some threshold, then the window is labeled as a negative before it is even passed to the shape-based object classifier. The appropriate context threshold for the rejection cascade is learned using cross

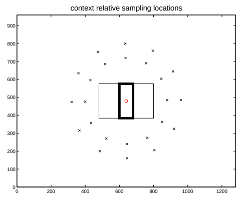

Figure 3: An illustration of the 24 relative sampling locations (black ×'s). The thin black rectangle represents the average size of the cars in the database, and the thick black rectangle represents the average size of pedestrians.

validation on the training set.

5. Experiments

Three experiments are described below, each of which is intended to test the fidelity of different subsystems of the SMF-based scene understanding architecture. Comparisons are made to other state of the art object detectors for these subsystems.

5.1. Experiment 1: Fidelity of the Shape-Based Object Detector

For the three shape-based objects car, pedestrian, and bicycle, we train C1 based detectors as described in Sec. 4.2. For comparison, we also train classifiers for these objects using three other well known object detection techniques. The ROC results of this experiment are illustrated in Fig. 4. The thick dashed curve labeled “Grayscale” indicates the results of training a system using a simple grayscale feature vector instead of the C1 values. In this system, each example is normalized in size and histogram equalized to build the feature vector.

Another base-line detector is built using patch-based features similar to those described in [25]. Each patch-based feature f_i is associated with a particular patch p_i, extracted randomly from the training set. The value of f_i is the maximum of the normalized cross correlation of p_i within a window of the image. This window of support is equal to a rectangle three times the size of p_i and centered in the image at the same relative location from which p_i was originally extracted. The advantage of these types of features over the gray-scale features is that the patch features can be highly object-specific while maintaining a degree of position invariance. For the results illustrated in Fig. 4, the system is implemented with $1,024$ such patch features with patches of size 12×12 in images of size 128×128.

For the SMF classifier, the ”grayscale” classifier and the “local patch correlation” classifier, the statistical learning machine is a boosting classifier with 150 rounds. Performance is no better using a linear or polynomial kernel SVM.

The final baseline system compared to in Fig. 4 is a part-based system as described in [16]. Briefly, object parts and a geometric model are learned via image patch clustering, and detection is performed by re-detecting these parts and allowing them to vote for objects-at-poses in a generalized Hough transform framework. While good results for cars were reported in the original work, we see that for the *pose-independent* learning problem, this patch-based constellation model is outperformed by the SMF system. Over-

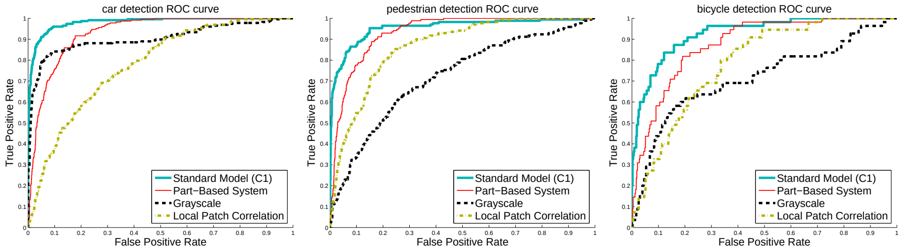

Figure 4: ROC curves illustrating the performance our the shape-based object detector and three baseline systems.

all, for all the three object categories tested, the standard model feature based classifiers were dominant.

5.2. Experiment 2: Fidelity of the Texture-Based Object Detector

In order to measure the fidelity of our object detectors, we compare performance to other state-of-the-art texture recognition systems using ROC curves. However, quantification of the performance of the texture-based object detectors is made complicated by the nature of the database itself. First, due to object occlusions, some pixels in the StreetScenes database are labeled as one object, i.e., “building”, but their actual appearance is due to another object, i.e., “tree.” We address this by removing pixels with multiple labels, or no label, from the test. Second, the detector output when the receptive field overlaps a texture-boundary is unreliable. This issue is addressed by segmenting the input image and averaging the detectors’ responses over each segment. As a result, uncertain responses at object borders are offset by the responses completely within the object boundaries.

In Fig. 5 we compare the results of the SMF texture-based object detector against four other texture classification systems. The “Blobworld” system is constructed using the Blobworld features described in [4]. Briefly, the Blobworld feature is a six dimensional vector at each pixel; 3 dimensions encode the color in the well-known Lab color space, and 3 dimensions encode the texture using the local spectrum of gradient responses. The curves labeled “Texton 1” and “Texton 2” are are the results of a system based on [20]. The texton feature is extracted by first processing the test image with a number of predefined filters. Texton 1 uses 36 oriented edge filters arranged in 5° increments from 0° to 180°. Texton 2 follows [20] exactly by using 36 gabor wavelet filters at 6 orientations and 3 scales. For both of these systems independently, a large number of random samples of the 36 dimensional edge response images are taken and subsequently clustered using k-means to find 100 cluster centroids. Each of these centroids is called a ’texton.’ The ’texton image’ is calculated by finding the index of the nearest texton for each pixel in the edge response images. The feature vector used for learning the texture-based object model is built by calculating the local 10×10 histogram of texton values. The texton feature is thus 100 dimensional, one dimension for each histogram bin. Finally, the “Histogram of Edges” system is built by simply using the same type of histogram framework, but over the 36 dimensional directional edge response of “Texton 1” rather than the texton identity. Learning is done with 150 rounds of boosting over regression stumps.

From Fig. 5 we see that, while different methods have particular strengths for some

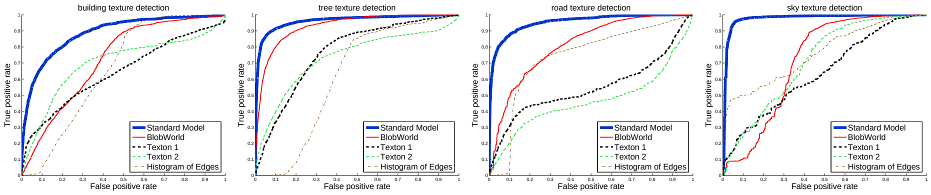

Figure 5: ROC curves of the performance of five texture classification algorithms on four classification tasks; the detection of buildings, trees, skies, and roads.

Context Classifier	car	pedestrian	bicycle
no context	.9809	.9551	.9275
position only	.9832	.9585	.9384
using true texture-object locations	.9865	.9672	.9521
using texture-object detections	.9868	.9606	.9554

Table 3: Area under ROC curve for object detection using both appearance and contextual cues in a rejection cascade.

objects, the SMF based texture system has superior performance on every texture-based object class. Changing the type of classifier or removing the smoothing over segments step does not change the order of the performances.

5.3. Experiment 3: Fidelity of the Contextual Modulation

The context system described in Sec. 4.4 is designed to augment the detection power of the shape-based object detectors by automatically removing the false positives that are out of context. The best way to measure this type of system is to show how much performance is gained by using the context as filter. In table 3 we document the area under the ROC curve for the shape-based detectors both with and without contextual assistance. In addition, we present two alternative contextual systems. One baseline system treats context as a learned position prior. This works because some locations are more likely to contain the shape-based objects than others. The other comparison system is given access to the true locations of all the texture-based objects in the image, rather than the detection scores. In two cases the system relying upon the estimated texture-based object locations outperforms even the system with access to the true locations of the texture-based objects. The difference may be due to the greater consistency in the labeling from the empirical detections.

6. Summary and Conclusions

The standard model feature set, a feature set designed to closely model the early visual computation in the brain-area V1, has been successfully employed previously for the task of unsegmented object recognition. In this work we have shown that these features also excel in three other areas important to computer vision, specifically, segmented object-detection, texture-recognition, and context understanding. By tying these three tasks together within the common SMF framework we have built a system capable of rudimentary image understanding in a challenging domain.

References

[1] A. C. Berg, T L. Berg, J. Malik. Shape Matching and Object Recognition using Low Distortion Correspondence. U.C. Berkeley Technical Report, 2004.

[2] E. Borenstein, and S. Ullman. Class-specific, top-down segmentation. In *ECCV*, pages 109–122, 2002.

[3] A.C. Bovik, M. Clark, and W.S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:55–73, 1990.

[4] C. Carson, M. Thomas, S. Belongie, J. Hellerstein and J. Malik", Blobworld: A system for region-based image indexing and retrieval. In "Third International Conference on Visual Information Systems", "Springer", "1999"

[5] C. M. Christoudias, B. Georgescu, and P. Meer. Synergism in low level vision. in *ICCV*, vol. IV, pages 150–155, August 2002.

[6] G. Csurka, C. Bray, C. Dance, L. Fan. Visual categorization with bags of keypoints. The 8th European Conference on Computer Vision - ECCV, 2004.

[7] J.G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Optical Society of America A*, 2:1160–1169, 1985.

[8] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001*, pages 341–346, August 2001.

[9] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR, Workshop on Generative-Model Based Vision*, 2004.

[10] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, volume 2, pages 264–271, 2003.

[11] W.T. Freeman, E.C. Pasztor, and O.T. Carmichael. Learning low-level vision. *Intl. Journal of Computer Vision*, 40:25–47, 2000.

[12] D. Gabor. Theory of communication. *Journal of the IEE*, 93:429–457, 1946.

[13] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio. Categorization by learning and combining object parts. In *NIPS*, Vancouver, 2001.

[14] D. Hubel and T. Wiesel. Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *J. Neurophys.*, 28:229–89, 1965.

[15] T. Lindeberg. Scale-space theory: A framework for handling image structures at multiple scales. In *Proc. CERN School of Computing, Egmond aan Zee, The Netherlands.*, 1996.

[16] B. Leibe, A. Leonardis, and B. Schiele. Combined Object Categorization and Segmentation with an Implicit Shape Model. In *SLCP '04 Workshop on Statistical Learning in Computer Vision*, Prague, May 2004.

[17] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. In *PAMI*, volume 23, pages 349–361, 2001.

[18] K. Okada, J. Steffens, T. Maurer, H. Hong, E. Elagin, H. Neven, , and C. von der Malsburg. The bochum/usc face recognition system and how it fared in the feret phase iii test. In *Face Recognition: From Theory to Applications*, pages 186–205, 1998.

[19] M. Riesenhuber and T. Poggio. How visual cortex recognizes objects: The tale of the standard model. *The Visual Neurosciences*, 2:1640–1653, 2003.

[20] L. W. Renninger, and J. Malik When is scene recognition just texture recognition? Vision Research, 44, 2301-2311.

[21] T.D. Sanger. Stereo disparity computation using gabor filters. *Biological Cybernetics*, 59:405–418, 1988.

[22] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *CVPR*, pages 746–751, 2000.

[23] Serre, T., L. Wolf and T. Poggio. Object Recognition with Features Inspired by Visual Cortex. In *CVPR*, 2005, San Diego, June 2005. -

[24] K.-K. Sung. *Learning and Example Selection for Object and Pattern Recognition*. PhD thesis, MIT, Artificial Intelligence Laboratory and Center for Biological and Computational Learning, Cambridge, MA, 1996.

[25] A. Torralba, K.P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multi-class object detection. In *CVPR*, 2004.

[26] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual features of intermdediate complexity and their use in classification. *Nature Neuroscience*, 5(7):682–687, 2002.

[27] P. Viola and M. Jones. Robust real-time face detection. In *ICCV*, volume 20(11), pages 1254–1259, 2001.

[28] P. Viola, M. Jones, and J.Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV*, volume 2, pages 734–741, 2003.

[29] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *ECCV*, Ireland, 2000.