Discriminant Filters For Object Recognition

Owen Carmichael, Shyjan Mahamud, Martial Hebert

Technical Report CMU-RI-TR-02-09
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper presents a technique for using training data to design image filters for appearance-based object recognition. Rather than scanning the image with a single set of filters and using the results to test for the existence of objects, we use many sets of filters and take linear combinations of their outputs. The combining coefficients are optimized in a training phase to encourage discriminability between the filter responses for distinct parts of the object and clutter. Our experiments on three popular filter types show that by using this approach to combine sets of filters whose design parameters vary over a wide range, we can achieve detection performance competitive with that of any individual filter set. This in turn can ease the task of fine-tuning the settings for both the filters and the mechanisms that analyze their outputs.[1]

1. Introduction

In recent years, local-appearance-based approaches to object recognition have shown great potential to solve detection and pose estimation problems because they combine the modelling simplicity of global-appearance-based methods [14] with the robustness they lack. An object is represented by the appearances of its parts[2] in a set of labelled training images (Figure 1); finding the object in a novel view begins with search for image patches containing the component parts (Figure 2). After these sections of object have been found, their relative positions in the image gives evidence to the pose of the object [21] or the likelihood that the part detections were spurious [3].

In many approaches, component appearance is represented by the responses of image windows containing it to one or more filters. A common paradigm is to first gather up example image patches containing the parts and run them through the filters; the filters can then be applied to a novel image and their responses compared to those for the training views to check for part matches. Filter sets appropriated for this purpose include convolutional kernels like Gabor wavelets [2] [13][18] and Gaussian derivatives[18][2][17], differential invariants based on combining the outputs of those kernels [19], local eigenspaces/PCA [5][10][15], and color statistics[22]. Each contains distinct characteristics that make them advantageous; for example, Gabor kernels are well-localized in space and frequency, invariants can tolerate transformations of the image, and local eigenspaces minimize reconstruction error of the training views.

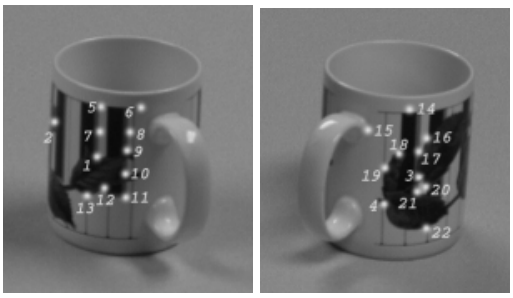

Figure 1: Example images with object parts labelled.

But none of these filters are designed from the beginning with component detection in mind. Those relying on pre-defined banks of kernels or statistics are not necessarily tuned to the appearances of specific objects and environments, so if the design parameters of the kernels are not properly set, it is possible that filter responses from distinct object parts will be indistinguishable from each other, or that outputs for background clutter will be the same as those for the object to be searched for. Local eigenspaces, although derived directly from training views, are not neccessarily tuned for discriminability either– they maximize the pooled covariance of all example patches of all object components, but do not necessarily encourage discriminability

[1]This work was supported by NSF Grant IIS-9907142.

[2]Throughout this paper, we use the terms "part," "component," and "section" to refer to any small piece of surface on the object, not its functional components (i.e. limbs, joints, etc.) For example, each of the labelled rectangles in Figure 2 frames the image of what we call an "object part." The term "patch" always applies to images, not 3D surfaces.

between the different components and background. As a result, implementations rely on two types of optimizations: tuning the design parameters of the filters so that outputs for distinct parts are distinct, and adjusting the settings of the classification mechanism that decides which filter outputs at run-time correspond to which sections of object. For example, suppose we want to recognize objects by using image responses to a set of Gaussian derivatives to tally votes for object parts in a hash table, as in [17]. For some recognition scenarios it may be unclear how to determine what Gaussian standard deviation (later referred to as the "width" or "scale" of the kernel) will result in responses that are well-clustered for the same component and well-separated for different components; it may also be difficult to determine how to size the bins in the hash table to minimize incorrect votes.

There are two common solutions to this problem. First is to discretize the range of reasonable filter parameter settings, run recognition experiments using each setting in turn, and select the setting which gives the best performance. Second is to generate filter responses over many parameter values for the same image patch at training time and/or run time. As an example of the second approach, Schiele et al [18] gather responses of training patches to Gaussian derivatives or Gabor kernels at several scales offline and compare these to the outputs for a single scale on a new image. Local eigenspace techniques, on the other hand, tend to take the first approach, generating filter responses for a single parameter setting for training and testing[5][10][15].

We show experimentally that by optimizing linear combinations of filter sets over a range of filter parameters we can achieve good part detection performance without requiring a suite of trial-and-error experiments or training a classifier with multiple distinct sets of responses per patch. Furthermore, we demonstrate that in some cases, the resulting filters can enhance classification to a degree that enables simpler classification mechanisms at run-time. We emphasize that this paper does not propose a new functional form for image filters; rather, we introduce a way to use training data to automatically combine sets of filters of any type in a way that enhances detection.

We call the resulting filters *discriminant filters* because the combining coefficients are optimized to discriminate between responses for distinct object parts and clutter. As an example, to design the Gaussian derivative kernels mentioned earlier we would synthesize many filter sets, one for each choice of Gaussian width over a range of plausible values. We then compute the responses of the training patches to each of the filter sets, and determine what the coefficients of a linear combination of those filter sets should be so that the responses for different object parts are discriminated from each other, while responses for the same part are tightly clustered. The discriminant filter in this case is the result of combining the Gaussian derivatives using those coefficients.

Figure 2: Detection of a subset of the parts from Figure 1 and clutter (marked "C") in a novel image. For display purposes, we only searched for six of the 11 labelled sections on this side of the mug. The number of rectangles around each component is proportional to the confidence in its classification. Note that the window on the upper left portion of the mug is *not* mislabelled as clutter; since none of the components of interest are on that portion of the object, it is technically "clutter" for parts detection purposes. Results for detection of the complete set of mug parts using the approach described in this paper are presented in Section 4.

A survey of related approaches to filter design is in Section 2, followed by the formulation of discriminant filters in Section 3. Detection experiments which apply discriminant filters to three previously reported image filters are described in Section 4 and discussed in Section 5.

2. Previous Work

To describe the problem setting more formally, we start with S sets of example image patches $C = \{C_1, C_2, \ldots, C_S\}$ corresponding to views of S different object sections. We denote a set of m image filters as a function Φ which maps a k-dimensional space of image patches of fixed size to the m-dimensional space of filter outputs. Our recognition paradigm is to compute $\Phi(x)$ for all $x \in C$ and use these outputs to train a classifier to correctly detect when novel responses $\Phi(x_t)$ corresponding to image patches x_t contain some object part. This section reviews previous designs for Φ and the classifier.

Several authors, including [5], [15], [20], and [10] propose the use of principal components analysis to model the local appearances of components, much the way earlier researchers [12] [14] modelled global appearances. An eigenspace decomposition is computed for the set of all training patches (or Fourier transforms of them as in [10]), and run-time parts detection consists of projecting novel image windows onto the first several significant eigenvectors. Since projecting patches into the eigenspace is a dot-product operation, the significant eigenvectors can equivalently be thought of as eigen-"filters" that are correlated

with the test image. In other words, PCA techniques which project data onto the first m eigenvectors can be formulated as an m-dimensional Φ. Local eigenspaces are an excellent way to model the appearances of components in a low-dimensional subspace since the first m principal components represent the best m-dimensional fit of all patches in an SSD sense; however for our application we are more interested in discriminating between sets of image patches than reconstructing them. In particular, while eigenspace techniques maximize the covariance of filter outputs over all classes of image patches, they do not necessarily encourage partitions between them. As a result, it is necessary to tune parameters of the PCA decomposition– namely, the image patch size and number of significant eigenvectors– to ensure that eigenfilter responses for different object parts and for clutter are not confused. Reasonable settings depend on the scale of object features and viewing conditions such as levels of occlusion, noise, clutter, and lighting.

A more prevalent approach to filter design is to construct banks of convolutional kernels based on criteria that do not depend on individual instances of training data. Gabor wavelets are an especially popular kernel choice [2] [13][18] since they are localized in frequency and space; it is easy to synthesize a set of Gabor kernels that regularly blankets spatial and frequency domains. Gaussian derivatives are also widely used [18][2][17] due in part to the fact that responses to them are equivariant to scale [18]; they have the added advantage that filter outputs for certain transformations of the image patches can be determined automatically by steering[7]. While these kernels form a mathematically sound way to represent the image signal present in the patches, responses from them are not necessarily sufficient for discrimination; in practice we will need to adjust the Gaussian widths of these filters, and the frequencies of Gabor kernels, to ensure that the information extracted from views of different object parts and clutter can be disambiguated from each other.

Invariants based on kernel responses are helpful since the outputs for the same object component will not vary at all when the image of the part undergoes certain transformations; for example the differential invariants in "jet" space computed by Schmid et al[19] will not change if the image of the part undergoes a rigid displacement. Still, there is no guarantee that for a particular set of kernel parameters these invariants will be distinguishable for different parts. Again, to ensure discriminability, the settings of the kernel bank must be tuned.

Other image filters, for example those based on local contour invariants [21], color invariants[22], and Laplacian zero-crossing images [11] could suffer the same limitation– the parameters for these transformations may need fine-tuning to reduce confusion between outputs for distinct components.

An expressive classification mechanism may accommodate image filters whose outputs are not necessarily tuned for discrimination. Mohan, for example[13], gets good parts detection results using Gabor kernel responses classified by a set of support vector machines with nonlinear Mercer kernels, while Nelson et al [21] achieve high performance using contour invariants and a hash table. The drawback is that the classification mechanisms are governed by their own set of parameters that must be tuned. In particular, the choice of Mercer kernel and penalty terms for SVMs affect detection and false alarm rate while bin size and policy for handing out votes are critical for indexing schemes to perform well. There is evidence that hashing schemes in particular are especially sensitive to parameter settings [8]. Worse, the effect of filter design and classifier design on performance is coupled– a change in the number of eigenvectors in PCA, for instance, may alter the design of k-nearest-neighbor distance functions which analyze the filter responses. Our results suggest that in some cases discriminant filter responses can cluster well enough that simple classifiers can perform well– for example, we see acceptable detection results by fitting Gaussian distributions to outputs.

While discrimination-centered techniques have not yet been applied to filter selection in local-to-global recognition, they have appeared in global approaches. In the Fisherfaces method[1], images of an entire object (faces in this case) are projected into a low-dimensional space using PCA, and a second linear transformation is determined by optimizing a Fisher ratio to encourage disparate outputs for different objects and similar outputs for the same object. Our approach differs in that we take many image transformations and combine their outputs, while Fisherfaces incorporate one eigenspace decomposition.

Meanwhile, in a number of texture segmentation papers[16][23], local-level filters are tuned for discrimination. Randen and Husøy[16], in particular, optimize linear FIR filters for a particular segmentation scenario so that maximizing the separation of their outputs to two textures becomes an eigenvalue problem similar to that found in the formulations for both Fisherfaces and discriminant filters. However, some aspects of the segmentation scenario and assumptions made by the authors restrict the applicability of this approach to component detection. For example, Randen and Husøy assume that the texture patches are separable autoregressive fields, while Weldon and Higgins[23] consider patches that are well-modelled as a dominant sinusoid plus bandpass noise.

3. Approach

For notational simplicity we illustrate the approach for the case of discriminating between two parts with image

patches C_1 and C_2. Our goal is to derive a function Φ which maximizes the following criterion:

$$\mathcal{R} = \frac{\frac{1}{|C_1||C_2|}\sum_{x_1 \in C_1, x_2 \in C_2} \|\Phi(x_1) - \Phi(x_2)\|^2}{\sum_{C_p \in \{C_1, C_2\}} \binom{|C_p|}{2} \sum_{x_1, x_2 \in C_p} \|\Phi(x_1) - \Phi(x_2)\|^2} \quad (1)$$

The numerator summarizes the distances between projected patches in C_1 and projected patches in C_2 and is analogous to the between-class scatter of Fisher discriminants[6]. The denominator summarizes the distances between projected patches in the same set and is analogous to within-class scatter. We assume that two sets whose patches are well-separated from each other will be the easiest to discriminate, so we seek a Φ which maximizes the numerator; at the same time, we assume that well-clustered sets of features require simpler representations for discrimination so we want Φ to minimize the denominator.

We express Φ as a linear combination of m-dimensional basis functions $\boldsymbol{\phi_j}$:

$$\Phi(x) = \sum_j \alpha_j \cdot \boldsymbol{\phi_j}(\boldsymbol{x}), \qquad \boldsymbol{\phi_j}(x) = \begin{bmatrix} \phi_{1j}(x) \\ \phi_{2j}(x) \\ \vdots \\ \phi_{mj}(x) \end{bmatrix}$$

This representation is not a restriction; we can represent arbitrarily complex functions Φ provided that we have a sufficient number of unique basis functions. Each $\boldsymbol{\phi_j}$ represents one set of filters for a particular parameter setting; returning to the Gabor kernel example, each $\boldsymbol{\phi_j}$ could correspond to a different choice of envelope width, with each ϕ_{ij} being a Gabor kernel with that width and some choice of orientation and frequency. Given a set of n basis functions $\boldsymbol{\phi_j}$ for n different parameter settings, we seek to find the set of coefficients $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_n]^T$ which maximizes $\mathcal{R}$. Substituting $\sum_j \alpha_j \cdot \boldsymbol{\phi_j}(\boldsymbol{x})$ for Φ in (1) and rearranging terms, we see that the numerator is equal to

$$\frac{1}{|C_1||C_2|}\sum_j \sum_k \alpha_j \alpha_k A_{1jk} - 2 \cdot \sum_j \sum_k \alpha_j \alpha_k B_{12jk} + \sum_j \sum_k \alpha_j \alpha_k A_{2jk}$$

and the denominator is

$$\binom{|C_1|}{2}(2 \cdot \sum_j \sum_k \alpha_j \alpha_k A_{1jk} - 2 \cdot \sum_j \sum_k \alpha_j \alpha_k B_{11jk}) + \binom{|C_2|}{2}(2 \cdot \sum_j \sum_k \alpha_j \alpha_k A_{2jk} - 2 \cdot \sum_j \sum_k \alpha_j \alpha_k B_{22jk})$$

where

$$A_{pjk} = \sum_{i=1}^{m} \sum_{x_1 \in C_p} \phi_{ij}(x_1)\phi_{ik}(x_1)$$

and

$$B_{pqjk} = \sum_{i=1}^{m} \sum_{x_1 \in C_p, x_2 \in C_q} \phi_{ij}(x_1)\phi_{ik}(x_2)$$

The ratio may be expressed equivalently as

$$\mathcal{R} = \frac{\boldsymbol{\alpha}^T N \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T D \boldsymbol{\alpha}} \quad (2)$$

where N and D are n-by-n matrices such that

$$N(j,k) = \frac{1}{|C_1||C_2|}(A_{1jk} - 2B_{12jk} + A_{2jk})$$

and

$$D(j,k) = \binom{|C_1|}{2}(2A_{1jk} - 2B_{11jk}) + \binom{|C_2|}{2}(2A_{2jk} - 2B_{22jk})$$

The $\boldsymbol{\alpha}$ which maximizes (2) is the eigenvector corresponding to the maximum generalized eigenvalue of N and D. Note that the coefficients A_{pjk} and B_{pqjk} which comprise N and D are readily computed by evaluating the basis functions $\boldsymbol{\phi_j}(\boldsymbol{x})$ over all elements of both sets C_1 and C_2 and taking various dot products and sums. The generalized eigenvalues of N and D may then be recovered using well-established numerical techniques.

This formulation is not restricted to two-component discrimination. In the general case, the within-class distances in the denominator will be summed over all sets and the across-class distances in the numerator will be summed over all pairs of sets, thus

$$N(j,k) = \sum_{C_p \neq C_q} \frac{1}{|C_p||C_q|}(A_{pjk} - 2B_{pqjk} + A_{qjk})$$

and

$$D(j,k) = \sum_{C_p} \binom{|C_p|}{2} 2A_{pjk} - 2B_{ppjk}$$

To summarize, our problem of combining basis functions to maximize distances across sets of image patches while minimizing distances within the sets reduces to evaluating the basis filters on the patches in the sets and finding generalized eigenvalues. As a concrete example, suppose we would like to use a vector of differential invariants based on derivatives of a Gaussian (as in [19]) to detect components, but it is unclear how to choose one or more Gaussian widths for the filters. We let each $\boldsymbol{\phi_j}$ correspond to one such vector of invariants for a particular choice of σ; by varying σ discretely over a range we arrive at a set of $\boldsymbol{\phi_j}$ functions which are combined using the derived coefficients α.

Figure 3: A sample of cluttered scenes containing the mug.

4. Experiments

We collected images of a common object (Figure 3) in varying poses and labelled the locations of selected object parts as in Figure 1. For each recognition experiment, we selected standard image filters from the literature and instantiated basis functions ϕ_j corresponding to a range of parameter settings for it. Given a subset of the labelled images as training data, we used the basis functions to derive discriminant filter coefficients as in Section 3, and used the remaining images for evaluation. To place the experiments in the context of previously reported end-to-end algorithms, we trained a nearest-neighbor classifier based on filter outputs as in [2][19][10][4][14][15][1]. For comparison, we also trained Gaussian clusters for classification as well. This section describes the data and experiments in detail.

4.1. Data

We took 60 images of a coffee mug with a hand-held camera. Of these photos, 12 featured the mug against a flat grey background and in the rest it was surrounded by a selection of clutter objects (Figure 3). For each of these shots, the camera was at roughly the same distance and elevation from the object, but there were still slight variations in object scale and in all components of rotation since the camera positions were not carefully controlled. The clutter objects maintained the same spatial arrangement with respect to each other in 12 of the pictures; for the other 36 we moved the pieces around between frames. We labelled 18 components on the mug in every view (Figure 1). Each of them appeared in at least 20 images.

4.2. Experimental Procedure

For each filter basis and classifier, we ran trials consisting of the following steps:

- *Data Collection*

    1. For each part, randomly select 20 views of it and partition them so that 75% (15 views) are used for training, and testing is done on the remaining 25% (5 views).
    2. Select 100 image patches of clutter at random and partition these 75-25 into train and test sets.

- *Training*

    1. Solve the eigenvalue problem for discriminant filter coefficients over the parts and clutter training sets, treating the clutter as though it were another object "part."
    2. Store the discriminant filter responses for the training views and train a classifier based on them.
    3. Store the filter outputs for each ϕ_j on training patches and train a separate classifier for the responses to each ϕ_j.
    4. Gather up all responses to each separate ϕ_j and train one classifier using all of them together as example data.

- *Testing*

    1. Run test set patches through each of the ϕ_j filters, compute the discriminant filters response from them, and process the results through each of the classifiers. In the case of the "all-ϕ_js-at-once" classifier, we follow a strategy found in earlier approaches[19][18]: pick one ϕ_j and compute responses for it at run time.

We ran 25 trials of this sort for each classifier and filter basis. Specific characteristics of our classifiers are described next.

4.3. Classifiers

We represent our set of Nc classifiers for Nc object parts as functions $\{f_1, f_2, \cdots f_{Nc}\}$. To train, we optimize these functions so that f_c is high for images of component c. At run time we gather up all scores $\{f_1(x), f_2(x), \cdots f_{Nc}(x)\}$ for all test cases x and count how many correct class scores are above a certain threshold versus how many incorrect scores are above the same threshold. In other words, this assessment is somewhat "pessimistic" as one test example can account for many false alarms.

For k-nearest-neighbors, we compute the filter response $\Phi(x)$ for each test patch x and find the k training examples $X^c = \{x_1^c, x_2^c, \cdots x_k^c\}$ from each class c whose responses are closest to x. The class score $f_c(x)$ for x belonging to class c is then

$$f_c(x) = \sum_{x^c \in X^c} exp(-C * \|x^c - x\|^2)$$

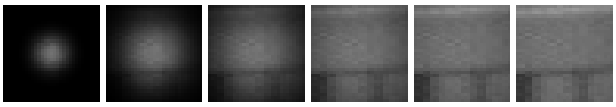

Figure 4: Image patch modulated by a range of Gaussian envelopes. For each Gaussian, a PCA decomposition is computed on the set of all training patches modulated by that Gaussian.

We estimate the parameter C for each class by brute force at training time; for each x in class c, we compute $f_c(x) - max_{c_2 \neq c}(f_{c_2}(x))$ for 50 different settings of C ranging from 10^{-6} to 10^6. In the end we pick the C for which the median of these values is highest. We emphasize that while this optimization is time-consuming, it is exactly the sort of optimization that in some cases a nearest-neighbor classifier may require to ensure good performance; there are no theoretical reasons for the exponential in f_c to take on one value or another. In all experiments we set k to 5.

To compare performance with a less-flexible, easily-trained classifier, we ran trials in which we fit a Gaussian distribution to filter outputs for training data. In other words, the classifier for each part c is a Gaussian function:

$$f_c(x) = (1/(2\pi)^{d/2}\|\Sigma\|^d) exp((-1/2)(x-\mu)^T\Sigma^{-1}(x-\mu))$$

and at training time we use filter responses to estimate the mean μ and covariance Σ. Since the covariance matrices Σ are estimated using a small number of examples relative to the dimension of the filters, we restrict Σ to be diagonal as in covariance selection methods [9].

4.4 Local Eigenfilters

The first set of experiments employs local eigenspaces for the filters. As in [5] [15][10], we want to collect all training patches for the various object components and perform principal components analysis on them, but we would also like to automatically determine how to incorporate multiple patch sizes into our filters. To apply discriminant filters to this problem, we simulate smaller effective window sizes by multiplying the patches by a Gaussian envelope. Depending on the standard deviation of this Gaussian, more or less of the periphery of the patch is set to zero (Figure 4). We perform PCA on the set of Gaussian-modulated image patches at training time; at run time, a test patch is multiplied by the same Gaussian and projected onto the first few significant eigenvectors.

We assign each $\boldsymbol{\phi_j(x)}$ to a different width of Gaussian envelope, so that each $\boldsymbol{\phi_j(x)}$ corresponds to PCA on a different effective window size. More formally, let G_σ denote a Gaussian with zero mean and standard deviation σ, and let the set of all image patches be $\{x_1, x_2, \cdots\}$. If we write $\{v_1^\sigma, v_2^\sigma \cdots v_n^\sigma\}$ for the first n principal components of $\{G_\sigma x_1, G_\sigma x_2, \cdots\}$, then we set $\phi_{ij}(x) = v_i^{\sigma_j} \cdot G_{\sigma_j} x$.

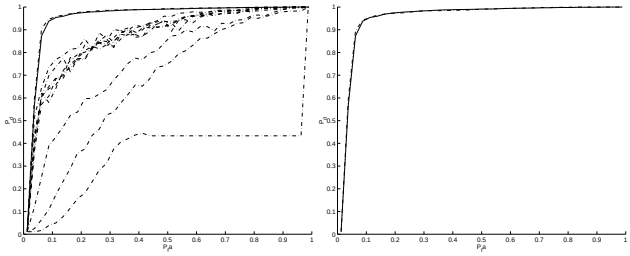

Figure 5: ROC curves using local eigenspace features and a Gaussian classifier. Solid curves on the left and right show performance of discriminant filters using eigenfilters as a basis. On the left, one dotted curve is plotted for each particular patch size. On the right, features for all patch sizes are combined at training time, and features for the median patch size are used at run time. Details in the text.

There are 10 different $\boldsymbol{\phi_j}$ filters, ranging from $\sigma = .25$ to $\sigma = 2.5$. Each has 10 principal components, resulting in a 10x10 basis.

Results using the Gaussian classifier for discriminant filters are plotted solid on Figure 5; results where responses are extracted using individual $\boldsymbol{\phi_j}$ filters are shown dotted. Figure 6 shows the same plot for the nearest-neighbor classifier. Comparing plots on Figure 6 to each other, we see that discriminant filters with a k-nearest neighbor classifier can be competitive with previous local eigenspace techniques which use k-nearest-neighbors, with the advantage that multiple window sizes were incorporated automatically. Comparing Figure 6 to Figure 5 suggests that the use of a Gaussian classifier does not degrade performance, even though its parameters are much easier to estimate than those of k-nearest-neighbors. Furthermore, discriminant filters performance is almost identical to that of classifiers trained on all patches of multiple sizes.

4.5 Gabor Filters

Next we consider banks of Gabor filters, used in a number of recognition methods[13][2][18]. Gabor filters are sinusoids modulated by a Gaussian envelope; as above, we would like to use discriminant filters to determine what frequencies and Gaussian widths lend themselves to effective parts detection.

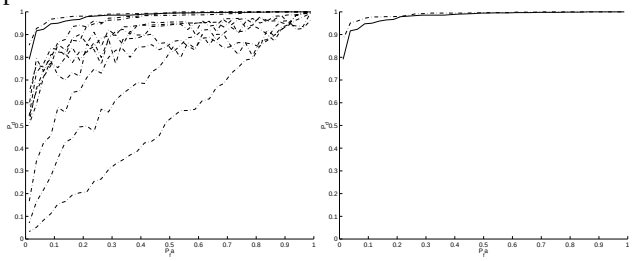

Figure 6: ROC curves using local eigenspace features and a k-nearest-neighbor classifier, as in Figure 5. Details in the text.

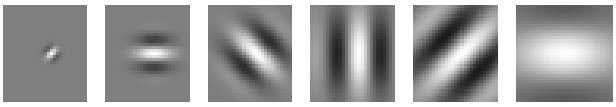

Figure 7: Examples of filters from the Gabor basis, varying by width of Gaussian envelope, frequency, and orientation.

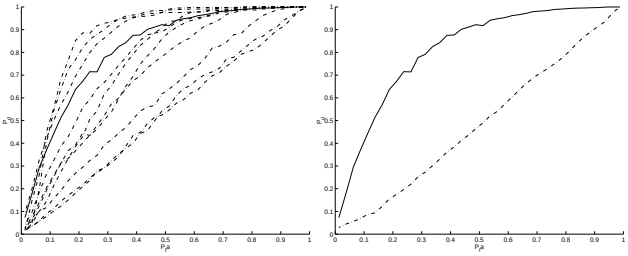

Figure 8: ROC curves using Gabor filters and a Gaussian classifier, displayed as in Figure 5. Details in the text.

For these experiments, each ϕ_j consisted of a set of 4 Gabor filters oriented at even intervals between 0 and π radians. The 25 different ϕ_js correspond to each possible combination of 5 frequencies ranging evenly from .2 to .5 and 5 Gaussian widths varying from .1 to .75(Figure 7). All filters have a 1:1 aspect ratio. As above, we used discriminant filters to derive 4-dimensional image features over the 4x25 basis, and trained Gaussian and k-nearest-neighbor classifiers to discriminate them for the 18 object parts and clutter. Results are shown in Figure 8 and Figure 9 (left). While discriminant filters do not perform as well as the best single filter set with a Gaussian classifier, the performance is comparable, and discriminant filters achieve better results than storing responses at multiple scales at training time. As in the previous section, the key point is that we were able to compute the discriminant filters directly, rather than running recognition experiments for each parameter setting in turn.

4.6 Differential Invariants

The next set of experiments is applied to differential invariants in "jet" space [19]. To compute an nth-order differential invariant for an image patch, we convolve it with all Gaussian derivatives up to order n and construct invariants by multiplying and adding the results together. As in [19], our experiments focus on the use of 3rd-order differential invariants under the rigid displacement group; there are 9 such unique invariants, so each ϕ_j will be 9-dimensional. These invariants are computed using derivatives of a single Gaussian, so we immediately arrive at the problem of determining what its standard deviation should be. In [19], Schmid et al compute the invariants over a range of discrete scales at training time and at a single scale at run time; here, we apply discriminant filters to the problem of selecting σ so that each patch is represented by a single vector of outputs during training. To do so, we select a set of values of σ and assign each ϕ_j to compute the differential invariants for a particular σ. We picked 10 values of σ ranging from .15 to .5, giving us a 10x9 filter basis (Figure 10). As above, we performed 25 trials using discriminant filters and 25 trials each for the individual settings of σ, using a Gaussian classifier. Figure 11 shows that in this case discriminant filters perform as well as the best setting of σ. Training a nearest-neighbor classifier using all invariants computed for all scales, and at run time using the invariants for the median scale, gives results that are comparable to those for discriminant filters (Figure 9,right). They are also comparable to those for the Gaussian classifier, suggesting again that it is possible to achieve acceptable recognition performance by combining invariants at different widths automatically, without training a classifier on responses for all possible

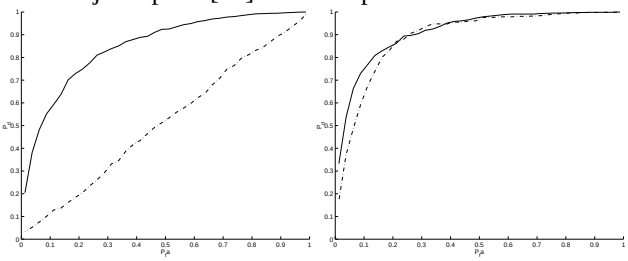

Figure 9: Left: ROC curves using Gabor filters and a nearest-neighbor classifier. The solid curve plots performance using discriminant filters; the dotted curve trains on all filter responses for all ϕ_j. Right: ROC curves using differential invariants and nearest-neighbor classifier, displayed as the graph on the left.

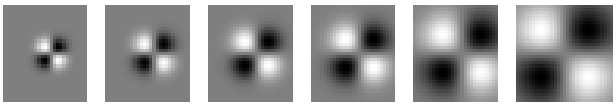

Figure 10: A filter basis is constructed using differential invariants computed from derivatives of Gaussians over a range of Gaussian variances. Shown is $\partial G/\partial x \partial y$ over that range of variances.

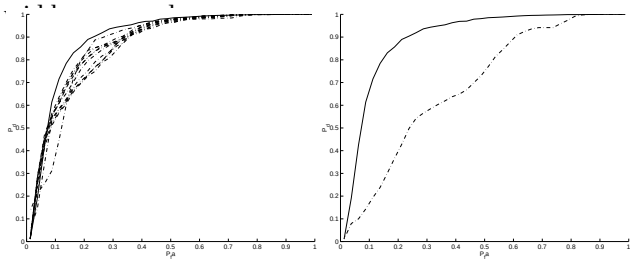

Figure 11: ROC curves using local differential invariant features and a Gaussian classifier. Solid curves show performance of discriminant filters using differential invariants of varying width as a basis. Left: One Dotted curve is plotted for each width setting. Right: Features for all widths are combined at training time, and features for the median width are used at run time. Details in the text.

5. Summary and Conclusions

This paper presents an approach to the detection of object parts based on tuned image filters. Given an arbitrary set of basis filters and sets of labelled patches containing the parts, we derive the combining coefficients needed to maximize discrimination between the filter outputs for the various classes. Initial detection results on real data and image filters commonly used in the recognition literature support the validity of this technique. As opposed to previous approaches, this paper suggests that it is possible to derive useful image information from a linear combination of sets of basis filters which span a plausible range of parameter settings, rather than selecting one setting or storing responses to all possible filters separately. The filters in turn can enable simple classification mechanisms that do not require much tuning.

References

[1] P. Belheumer, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces:recognition using class specific linear projection. In *Proceedings European Conference On Computer Vision*, 1996.

[2] J. Ben-Arie, Z. Wang, and R. Rao. Iconic recognition with affine-invariant spectral signatures. In *Proceedings IAPR-IEEE International Conference on Pattern Recognition*, volume 1, pages 672–676, 1996.

[3] M. Burl, M. Weber, and P. Perona. A probablistic approach to object recognition using local photometry and global geometry. In *Proceedings European Conference On Computer Vision*, pages 628–641, 1998.

[4] V. Colin de Verdiere and J. Crowley. Local appearance space for recognition of navigation landmarks. *Journal of Robotics and Autonomous Systems*, 1999. Special Issue.

[5] Vincent Colin de Verdiere and James L. Crowley. Visual recognition using local appearance. In *Proceedings European Conference On Computer Vision*, pages 640–654, 1998.

[6] Richard Duda, Peter Hart, and David Stork. *Pattern Classification.* Wiley-Interscience, 2 edition, 2001.

[7] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[8] W.E.L. Grimson and D. Huttenlocher. On the sensitivity of geometric hashing. In *Proceedings International Conference On Computer Vision*, pages 334–338, 1990.

[9] David Hand. *Construction and Assessment of Classification Rules*. Wiley, 1997.

[10] D. Jugessur and G. Dudek. Local appearance for robust object recognition. In *Proceedings IEEE Conference On Computer Vision And Pattern Recognition*, 2000.

[11] John Krumm. Object detection with vector quantized binary features. In *Proceedings IEEE Conference On Computer Vision And Pattern Recognition*, pages 179–185, June 1997.

[12] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *Proceedings International Conference On Computer Vision*, 1995.

[13] A. Mohan. Object detection in images by components. Technical Report AI Memo 1664, Massachusetts Institute of Technology, September 1999.

[14] H. Murase and Shree Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[15] K. Ohba and K. Ikeuchi. Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(9):1043–1048, 1997.

[16] T. Randen and J. Husøy. Texture segmentation using filters with optimized energy separation. *IEEE Transactions on Image Processing*, 8(4):571–582, 1999.

[17] R. Rao and D. Ballard. An active vision architecture based on iconic representations. *Artificial Intelligence*, 78(1-2):461–505, 1995.

[18] Bernt Schiele and James Crowley. Object recognition using multidimensional receptive field histograms. In *Proceedings European Conference On Computer Vision*, pages 610–619, 1996.

[19] C. Schmid and R. Mohr. Combining greyvalue invariants with local constraints for object recognition. In *Proceedings IEEE Conference On Computer Vision And Pattern Recognition*, 1996.

[20] H. Schneiderman and T. Kanade. Probabilistic modeling of local appearance and spatial relationships for object recognition. In *Proceedings IEEE Conference On Computer Vision And Pattern Recognition*, 1998.

[21] A. Selinger and R. C. Nelson. A perceptual grouping hierarchy for appearance-based 3d object recognition. Technical Report 690, University of Rochester Computer Science Department, May 1998.

[22] D. Slater and G. Healey. The illumination-invariant recognition of 3d objects using local color invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(2):206–210, 1996.

[23] T. Weldon and W. Higgins. Designing multiple gabor filters for multitexture image segmentation. *Optical Engineering*, 38(9):1478–1489, September 1999.