Learning Appearance Based Models: Hierarchical Mixtures of Experts Approach based on Generalized Second Moments

Christoph Bregler and Jitendra Malik

Computer Science Division
University of California at Berkeley
Berkeley, CA 94720
email: bregler@cs.berkeley.edu, malik@cs.berkeley.edu

Abstract

This paper describes a new technique for object recognition based on learning appearance models. The image is decomposed into local regions which are described by a new texture representation derived from the output of multiscale, multiorientation filter banks. We call this representation "Generalized Second Moments" as it can be viewed as a generalization of the windowed second moment matrix representation used by Garding & Lindeberg. Class-characteristic local texture features and their global composition is learned by a hierarchical mixture of experts architecture (HME by Jordan & Jacobs). The technique is applied to a vehicle database consisting of 5 general car categories (Sedan, Van with back-doors, Van without back-doors, old Sedan, and Volkswagen Bug). This is a difficult problem with considerable in-class variation. The new technique has a 6.5% misclassification rate, compared to eigen-images which give 17.4% misclassification rate, and nearest neighbors which give 15.7% misclassification rate.

Keywords: Learning, Appearance Based Object Recognition, Filter Banks, Texture Statistics, Second Moment Matrix, Mixtures of Experts, Car Classification.

1 Introduction

Machine learning offers potentially a very attractive methodology for the problem of acquiring object models for recognition:

1. The painstaking process of hand-crafting 3D object models is avoided.
2. Essential vs inessential variation among different objects in a class can be treated simply by looking at the variation in the examples in the training data.
3. Selection of the most useful i.e. most discriminating features for classification can be driven by the data instead of the programmer's a priori conception of what are likely to be the most salient and discriminating features.

Speech recognition is a prime example of the success of the learning methodology–Hidden Markov Models are learned using training data and then used to drive recognition. In computer vision, learning methods are much less popular and until a few years ago their application was limited to non-mainstream problems such as handwritten digit recognition.

A significant shift has occurred recently with the successful application of machine learning techniques based on acquiring (and subsequently using) appearance-based or viewer-centered techniques in contrast to object-centered 3D representations. Appearance-based schemes rely on collections of images of the object and implicit structural information. These techniques may be classified according to the degree of structure that is extracted from the image data and what kind of additional photometric, color, or texture information is explicitly used.

The most extreme solutions start with raw unprocessed images and show how task based feature detectors can be learned from a large image database automatically [26, 16, 20, 3, 25]. Neglecting any preprocessing is mainly motivated by the concern not to loose any useful information for the final classification.

To the other extreme belong approaches that assume that mid-level representations like outlines or contour images can be estimated robustly using "hard"-decision heuristics. Learning starts at this level and algorithms are proposed that deal with inducing models for geometrical configurations [21, 23].

Intermediate solutions impose explicit structure to some extent, but allow for data driven estimation of the global structural and local textural features. The domain of face recognition is a popular application for such decompositions [17, 2, 12, 11].

We propose a domain independent part decomposition using a 2D grid representation of overlapping local image regions. The image features of each local patch are represented using a new texture descriptor that we call "Generalized Second Moments". Based on this representation we learn class-based local features and their global relationships using a "Hierarchical Mixtures of Experts" Architecture (HME) [9]. Multiple experts are trained to classify object categories. Each expert is a generalized linear model (GLIM) that maps the grid based feature representation into a class probability vector. Potentially different experts are "responsible" for different object poses or sub-categories. A hierarchical set of "gating" functions compute the mixture coefficients for all experts.

We apply this technique to the domain of vehicle classification. We urge the reader to examine Figure 7 to see examples of the 5 different categories. Our technique could classify five broader categories with an error of as low as 6.5% misclassification. The vehicles are shown from behind covering a small set of poses. We compare the learning architecture to the nearest neighbor technique and the feature representation to the eigen-image technique. Across a large set of experiments our technique performed significantly better. The best results for eigen-image were 17.4% misclassification and the best results for nearest neighbor were 15.7%.

This paper is organized as follows. The next section gives an overview of other appearance based representations and our own solution. Section 3 explains Jordan & Jacobs's Hierarchical Mixture of Experts Architecture. In Section 4 comparative recognition results are presented on the vehicle classification task.

2 Representation

An appearance based representation should be able to capture features that discriminate the different object categories. It should capture both local textural and global structural information. This corresponds roughly to the notion in 3D object models of (i) parts (ii) relationship between parts.

2.1 Structural Description

Examples of representations that implicitly capture the global and local structure are Eigen-Images [26], and nonlinear subspace-representations [16] of graylevel images. To span all object configurations, pose, and lighting conditions, large training image databases are crucial to the success of these representations.

Objects usually can be decomposed into parts. A face consists of eyes, nose, and mouth. Cars are made out of window screens, tail lights, license plates etc. The question is what granularity is appropriate and how much domain knowledge should be exploited. A car could be a single part in a scene, a license plate could be a part, or the letters in the license plate could be the decomposed parts. Eyes, nose, and mouth could be the most important parts of a face for recognition, but maybe other parts are important as well.

It would be advantageous if each part could be described in a decoupled way using a representation that was most appropriate for it. Object classification should be based on these local part descriptions and the relationship between the parts. The partitioning reduces the complexity greatly and invariance to the precise relation between the parts could be achieved.

Examples of learning architectures that use domain specific part decompositions may be found in [2, 17, 12, 11] for face recognition. A more general decomposition based on line and circular edge segments is proposed by [21].

For our domain of vehicle classification we don't believe it is appropriate to explicitly code any part decomposition. The kind and number of useful parts might vary across different car makes. The resolution of the images (100x100 pixel) restricts us to a certain degree of granularity. We decided to decompose the image using a 2D grid of overlapping tiles or Gaussian windows[1]. The content of each local tile is represented by a feature vector (next section). The generic grid representation allows the learning architecture to induce class-based part decomposition, and extract local texture and global shape features. For example the outline of a face could be represented as

[1] A similar grid representation is used by [19] but only local classification for each tile region is done.

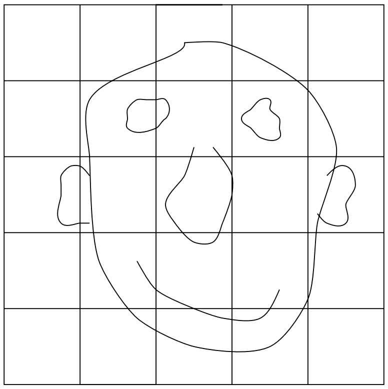

Figure 1: Generic grid decomposition of input image.

certain orientation dominance in the local tiles at positions of the face boundary (Figure 1). The eyes are other characteristic features in the tiles.

2.2 Local Features

We like to extract from each local tile characteristic features. Certainly images of cars contain lines and curves. A representation based on straight and circular edge segments is proposed by [21]. This requires hard decision thresholds. We like to avoid such heuristics and we like to capture a much richer set of geometric and non-geometrical information.

Convolving image regions with a large number of spatial filters, at various orientations, phases, and scales is the first step of robust edge detection [13]. Besides edge information the response values of such filters contain much more general information about the local neighborhood. It is a mature representation used in other early vision tasks like stereopsis [8], motion, [28, 7, 27], and texture discrimination [14]. Although this approach is loosely inspired by the current understanding of processing in the early stages of the primate visual system, the use of spatial filters has many advantages from a pure analytical viewpoint. Usually the filter kernels are highly orientation selective elongated Gaussian derivatives. Orientation specific features are very useful (i.e. invariant to illumination). If more than one orientation is present at a single point (e.g. junctions) there is more than one local maxima across the orientation selective responses.

The computation time of such convolutions can be significantly reduced with efficient methods proposed by [18, 4]. Instead of convolving the image with a rich family of orientation and scales, a small set of simple X-Y separable, steerable, and scalable basis functions can be found.

How could a local image region be described with just a few numbers? Representing the region with all filter responses at all locations is not feasible because of the high dimensionality. [12] investigated heuristics how to find interesting locations that are representative for the local neighborhood. Locations with local magnitude maxima or high responses across several directions could be interesting points. There might be many such interesting points in an image region. It is unclear how to pick the right number of points and how to order them.

Another way of representing the texture in a local region is done by [6] in calculating a windowed second moment matrix. Instead of finding maximum filter responses, the second moments of brightness gradients in the local neighborhood are weighted and averaged with a

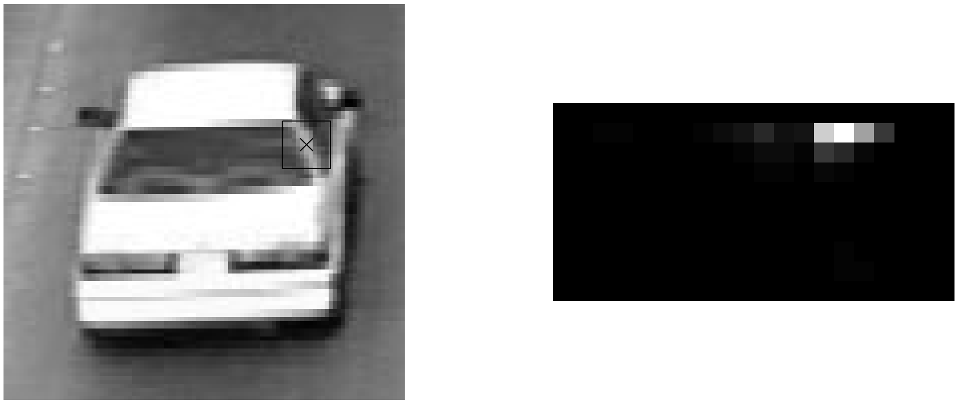

Figure 2: Left image: Input image to the filter bank convolving. The cross shows the selected location. Right image: The angle and scale distribution of the filter response at the selected location. The horizontal axis are angles from 0 to 180 degrees and the vertical axis are different scales.

circular Gaussian window. The gradient is a special case of Gaussian oriented filter banks. The windowed second moment matrix takes into account the response of all filters in this neighborhood. The disadvantage is that gradients are not very orientation selective and a certain scale has to be selected beforehand. Averaging the gradients "washes" out the detailed orientation information in complex texture regions.

Orientation histograms would avoid this effect [5]. Elongated families of oriented and scaled kernels could be used to estimate the orientation at each point. But as pointed out already there might be more than one orientation at each point, and significant information is lost.

2.2.1 Generalized Second Moments

We propose a new way to represent the texture in a local image patch by combining the filter bank approach with the idea of second moment matrices.

The goal is to compute a feature vector for a local image patch that contains information about the orientation and scale distribution. We avoid "hard" decisions such as finding locations of local magnitude maxima or local response maxima in the orientation/scale space.

We compute at each point (x, y) the filter response of a finite set of R basis kernels ($F_1(x, y), ..., F_R(x, y)$). These kernels are generated using the technique of X-Y separable steerable scalable approximations of filter kernels by [24]. It is possible to reconstruct for any desired oriented and scaled version of the filter family the response $F_{\theta,\sigma}(x, y)$ in interpolating the basis kernel responses using the interpolation function $h_r(\theta, \sigma)$:

$$F_{\theta,\sigma}(x, y) = \sum_{r=1}^{R} h_r(\theta, \sigma) F_r(x, y) \tag{1}$$

We can draw for each R-dimensional vector $(F_1, ..., F_R)$ the energies (squares) of the corresponding orientation/scale responses in a 2D coordinate system ($[F_{\theta,\sigma}(x, y)]^2$). Figure 2 shows an example of the reconstructed filter responses at a local point of the back screen frame. The kernel

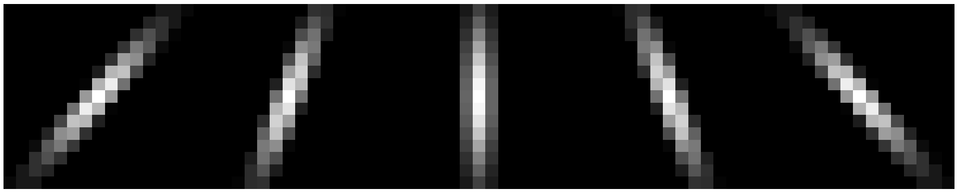

Figure 3: Elongated second order Gaussian derivatives at different orientations.

family are second order Gaussian derivatives with an elongation of 1:3 (Figure 3). The elongation produces a high peak in the orientation/scale space graph. In order to describe a whole image region instead of a single point we add all energies of the orientation/scale space responses of all points[2]. We weight each location by a spatial window function W (for example circular Gaussian):

$$E(\theta, \sigma) = \sum_{x,y} W(x,y)[F_{\theta,\sigma}(x,y)]^2 \tag{2}$$

Using elongated kernels produces orientation/scale peaks, therefor the sum of all orientation/scale responses doesn't "wash" out high peaks. The height of each individual peak corresponds to the intensity in the image. Little noisy orientations have no high energy responses in the sum. $E(\theta, \sigma)$ is somehow a "soft" orientation/scale histogram of the local image patch.

We can represent $E(\theta, \sigma)$ with just a few numbers in a way, similar to representing $F_{\theta,\sigma}(x,y)$ with just R numbers. Substituting (2) into (1):

$$\begin{aligned} E(\theta, \sigma) &= \sum_{x,y} W(x,y) \left[\sum_{r=1}^{R} h_r(\theta,\sigma) F_r(x,y) \right]^2 && (3) \\ &= \sum_{x,y} W(x,y) \sum_{r=1}^{R} h_r(\theta,\sigma) F_r(x,y) \sum_{r'=1}^{R} h_{r'}(\theta,\sigma) F_{r'}(x,y) && (4) \\ &= \sum_{r=1}^{R} \sum_{r'=1}^{R} h_r(\theta,\sigma) h_{r'}(\theta,\sigma) \sum_{x,y} W(x,y) F_r(x,y) F_{r'}(x,y) && (5) \end{aligned}$$

In equation (5) $h_r(\theta,\sigma)h_{r'}(\theta,\sigma)$ only varies with θ and σ but not with the image. The terms that depend on the local image patch of the window W are the $\frac{(R+1)R}{2}$ numbers $F_{r,r'} = \sum_{x,y} W(x,y)F_r(x,y)F_{r'}(x,y)$. This is similar to a $R \times R$ second moment matrix. The orientation/scale responses covering a local image patch can be reconstructed based on these $\frac{(R+1)R}{2}$-dimensional vector which we call "Generalized Second Moments".

Figure 4 shows an example of such an reconstruction. As you can see there a three peaks representing the edge lines along three directions and scales in the local image patch.

This representation greatly reduces the dimensionality without being domain specific or applying any hard decisions. It is shift invariant and decouples scale in a nice way. Dividing the $R \times R$ second moment matrix by its trace makes this representation also illumination invariant.

[2] If we just would add the original filter responses (not energies), this is equal to convoluting a constant image, where the constant factor is the sum of all image pixels (for odd symmetric filters this is equal to 0).

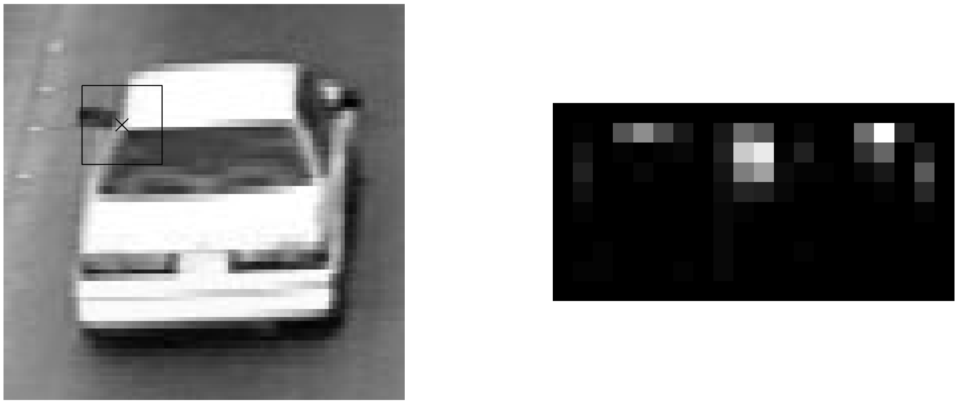

Figure 4: Left image: The black rectangle outlines the selected area of interest. Right image: The reconstructed scale and rotation distribution of the Generalized Second Moments. The horizontal axis are angles between 0 and 180 degrees and the vertical axis are different scales.

Using a 10x10 grid and a kernel basis of 5 first Gaussian derivatives and 5 second Gaussian derivatives represents each input image as an $10 \cdot 10 \cdot (5 + 1) \cdot 5 = 3000$ dimensional vector. Potentially we could represent the full image with one generalized second moment matrix of dimension 20 if we don't care about capturing the part decomposition.

3 Pattern Classification

Based on this representation we would like to *learn* an object classification.

The simplest techniques are so called nearest-neighbor or k-nearest-neighbor algorithms. An example database of labeled feature vectors is kept in storage. A new test feature vector is classified in finding the closest example vector in the database. Distance metrics could be the Euclidean distance. A more complex distance metric is the "tangent-distance" [25] which is invariant to slight shifting, rotation, and scaling if the feature vectors would be raw pixel images. In our case this is unnecessary, because the generalized second moments provide this invariance already. These techniques are very storage intensive and for that very slow in recall.

More powerful techniques for object classification are multi-layer-perceptrons [22]. [3] showed successfully how appearance-based recognition can be done using such techniques. Another version of neural networks are so called Radial-Basis-Networks. Radial-basis-functions can be circular Gaussian modeling receptive fields. [1, 15] demonstrated how such networks can be used for classification and interpolation.

3.1 Mixtures of Experts

The idea is the following: Experts are classifiers that are specialized on certain sub-domains. A gating function or mixture function weights the different experts and builds a combined classification hypothesis. Potentially each expert could be a specialist for a certain object pose or sub-category.

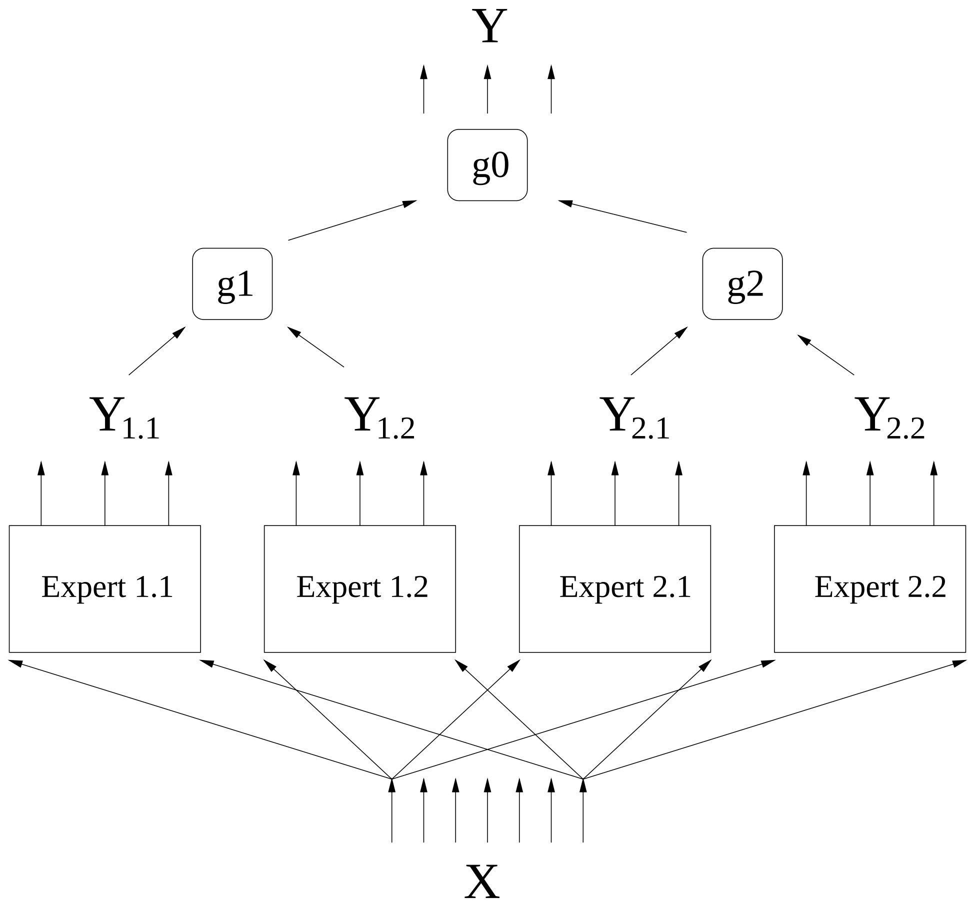

Figure 5: A two-level hierarchical mixtures of expert architecture.

Hierarchical Mixtures of Experts (HME [9]) are tree-structured mixture models in which both the mixture coefficients and the mixture components are generalized linear models (see below).

The mixture components or gating functions divide the feature vector space into a nested set of regions. In each region a certain expert fits a simple surface or liner mapping to the data that falls into these regions. Regions have "soft" boundaries, meaning that data points may lie simultaneously in multiple regions.

Figure 5 illustrates a two-level hierarchical mixture of experts. x is the input vector. In our case it will be the extracted feature vector. Each expert i computes a function $f : X \to Y_i$ from the input vector x to an output vector y_i:

$$y_i = f(U_i x) \tag{6}$$

U_i is a weight matrix from the input dimension to the N-dimensional output dimension (in case there are N categories to classify). $f(\cdot)$ is the logistic function. The expert outputs are interpreted as the log odds of "success" under a Bernoulli probability model.

The gating function j computes a function $f : X \to g_j$:

$$g_j = \frac{1}{1 + e^{v_j^T x}} \tag{7}$$

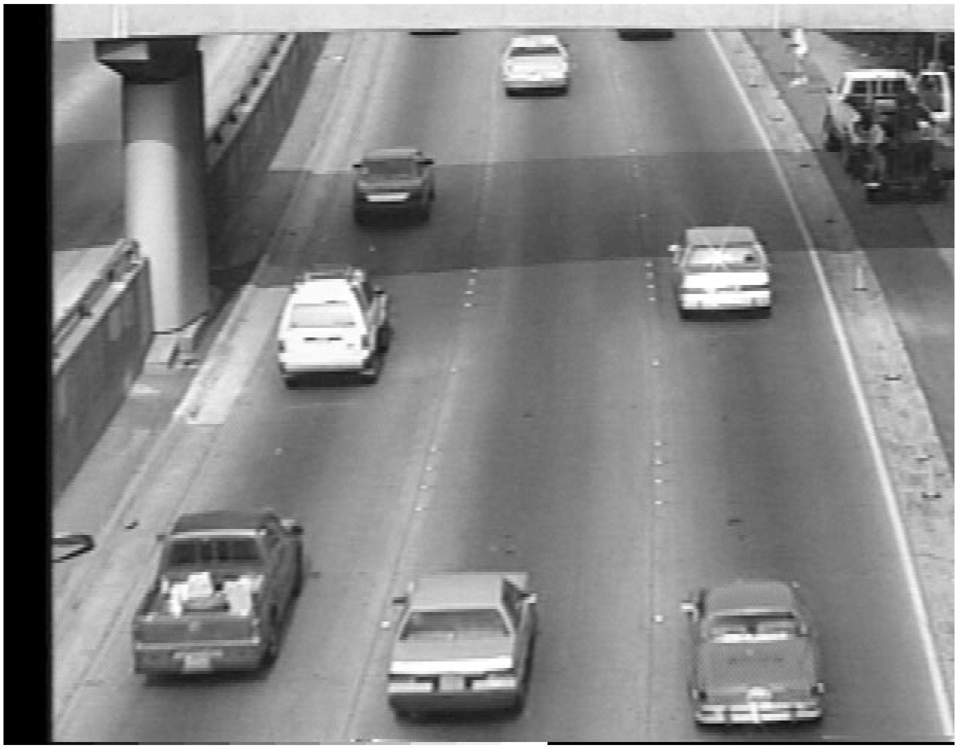

Figure 6: Typical shot of the freeway segment

v_j is a vector of the input dimension and g_j is a value between 0 and 1. Each gating function in the binary tree (Figure 5) weights the results of all experts on the left side of the gating node with g_j and on the right side of the gating node with $1 - g_j$. The final classification result is the weighted average of all local expert classifications y_i.

Given the training data and output labels, the gating functions and expert functions can be estimated using an iterative version of the EM-algorithm. For more detail see [9].

3.1.1 Linear subspace projection

As mentioned earlier our feature representation consists of 3000 dimensional vectors. For a 5 category classification problem this would mean we have to estimate for each expert a 3000 × 5 dimension matrix and for each gating function a 3000 dimensional vector. In order to reduce training time and storage requirements we project the input vectors into a 64 dimensional subspace that was estimated from the training data using principal components analysis.

4 Experiments

We experimented with a database consisting of images taken from a surveillance camera on a bridge covering normal daylight traffic on a freeway segment (Figure 6). The goal is to classify different types of vehicles. This might be useful for a traffic surveillance task, where cars have to be tracked over a longer freeway segment. Surveillance cameras are far apart from each other

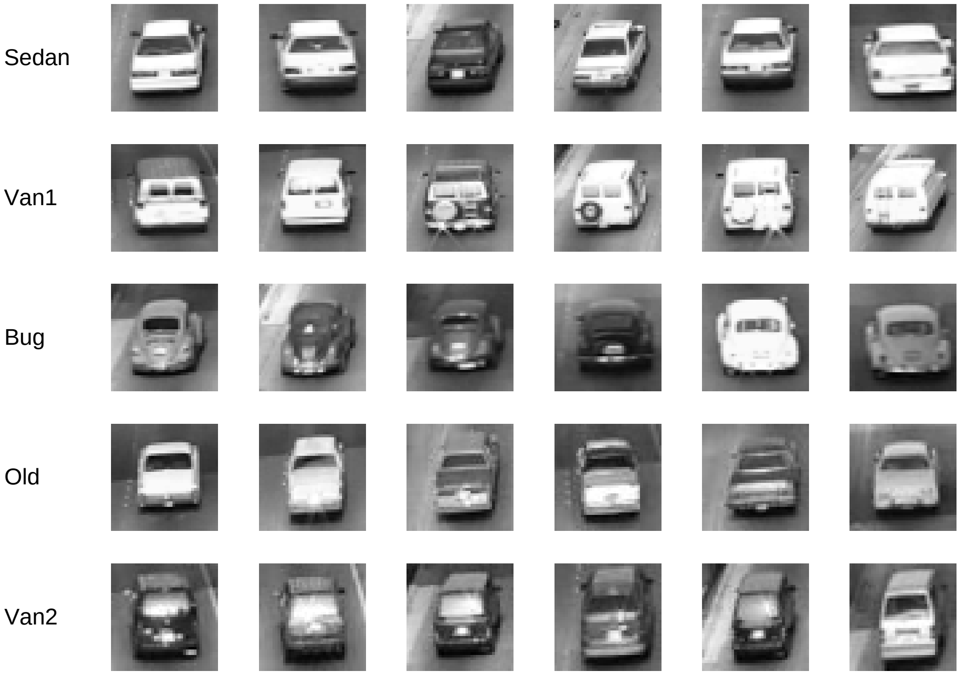

Figure 7: Example images of the vehicle database.

which require one to match cars seen in one segment to cars seen in another segment. We are able to segment each moving object based on motion cues [10]. Given the car segmentation 5 broader car categories should be classified: Modern Sedan, Old Sedan, Van with back-doors, Van without back-door, and Volkswagen Bug.

The images show the rear of the car across a small set of poses. All images are normalized to 100x100 pixel using bilinear interpolation. For this reason the size or aspect ratio can not be used as a feature. Figure 7 shows example images of all five categories.

4.1 Classification Performance

We experimented with grid sizes between 6x6 to 16x16 without getting significant different performance results. Figure 8 shows the classification performance using a 10×10 grid and training sizes between 57 and 228 examples. Each experiment is run 5 times with different random distribution of training and test images (Jack-Knifing). The database consists of 285 example images. Therefore the number of test images are (285 - number of training images). The generalized second moments were computed using a window of $\sigma = 6$ pixel, and 5 filter bases of 3:1 elongated first and second Gaussian derivatives on a scale range between 0.25 and 1.0. (We experimented also with 8 filter bases and a rectangle window for the second moment statistics without getting significant improvement). The eigen-image representation was done using the first 64 eigen-images. Both representations, the generalized second moments and eigen-images were evaluated with a HME architecture of 8 local experts, and the nearest neighbor technique. Across all experiments the HME architecture based on Generalized Moments was superior to all other techniques. The best performance with a misclassification of 6.5% was achieved using 228 training images. Using less

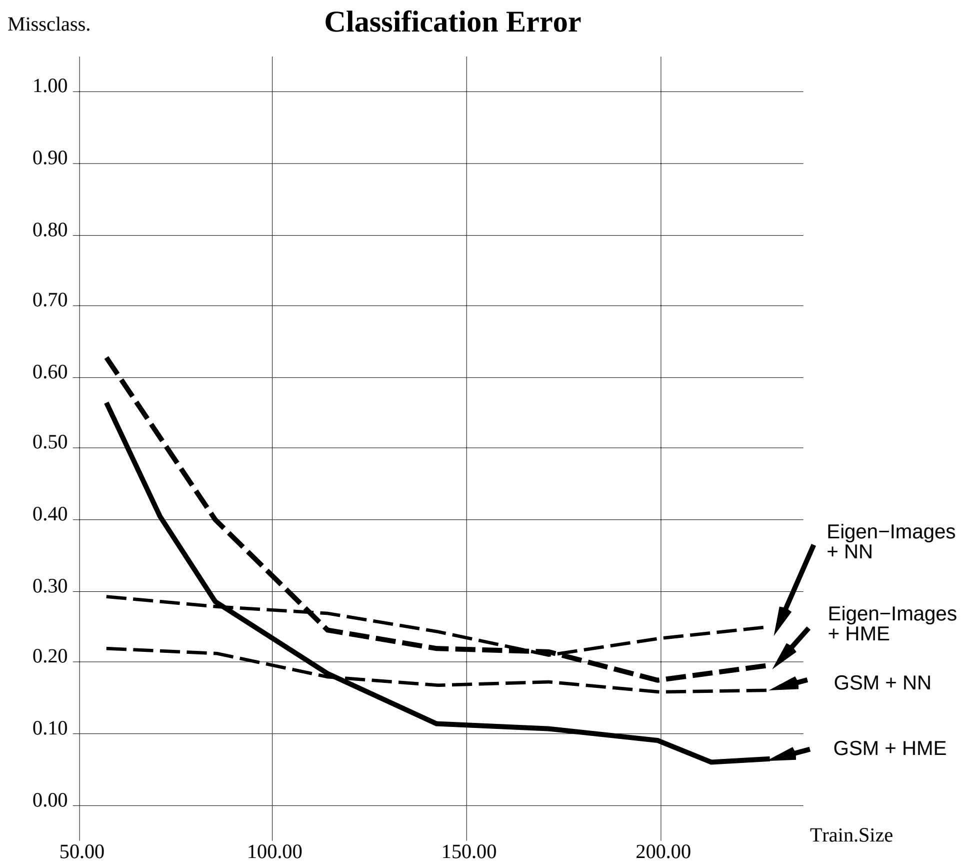

Figure 8: The classification errors of four different techniques. The X-axis shows the training size, and the Y-axis shows the percentage of misclassified test images. HME stands for Hierarchical Mixtures of Experts, GSM stands for Generalized Second Moments, and NN stands for Nearest Neighbors.

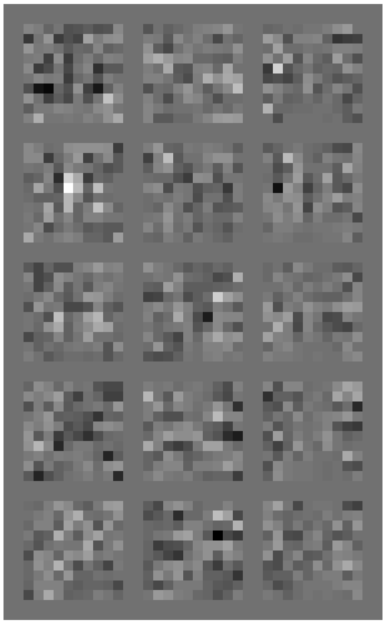

Figure 9: Example weights of one local expert.

than 120 training images, the HME architecture performed worse than nearest neighbors. This is the minimum of training samples to achieve generalization.

On analysis of the misclassified examples we found the most common confusion between sedans and “old” sedans. The second most confusion was done between vans with back-doors, vans without back-doors, and old sedans.

4.2 What has been learned

Another interesting question is what representation has been learned in the HME architecture. We were not able to “read” any obvious meaningful features out of the gating vectors. In general it is very hard to interpret the complex interplay of these high-dimensional weights. Looking at the weight matrices of the experts trained on simpler second moment matrices we found some interesting constellations. We used just two basis kernels, a vertical and a horizontal first Gaussian derivative, which results in 3 numbers per local tile (similar to [6]). This allows us to interpret the learned weights better. Figure 9 shows the weights back-projected into the $10 \times 10 \times 3$ grid space. Each of the five rows represents the weights used for each of the five categories. The first column is the part of the weight vector that is multiplied with the vertical second moments (F_{yy}) in each of the 10x10 tiles. The second column represents the horizontal second moments (F_{xx}), the third column is F_{xy}. The second row represents the category of vans with back-doors. One significant feature for such vans is a vertical bar in the middle of the car (separation of the two doors). Indeed the value that is multiplied with this region is significant larger than all other values, which indicates that the HME architecture uses this value as the most significant value to discriminate such vans from other categories. Unfortunately the remaining values are hard to interpret. The final classification is a complex interplay of the multiple gating and expert weights.

5 Conclusion

We have demonstrated a new technique for appearance-based object recognition based on a 2D grid representation, generalized second moments, and hierarchical mixtures of experts. Experiments have shown that this technique has significant better performance than other representation techniques like eigen-images and other classification techniques like nearest neighbors.

We believe that learning such appearance-based representations offers a very attractive methodology. Hand-coding features that could discriminant object categories like the different car types in our database seems to be a nearly impossible task. The only choice in such domains is to estimate discriminating features from a set of example images automatically.

The proposed technique can be applied to other domains as well. We are planning to experiment with face databases, as well as larger car databases and categories to further investigate the utility of hierarchical mixtures of experts and generalized second moments.

Acknowledgments

We would like to thank Leo Breiman, Jerry Feldman, Thomas Leung, Stuart Russell, and Jianbo Shi for helpful discussions and Michael Jordan, Lawrence Saul, and Doug Shy for providing code.

References

[1] D. Beymer, A. Shahsua, and T. Poggio. Example based image analysis and synthesis. *M.I.T. A.I. Memo No. 1431*, Nov 1993.

[2] R. Brunelli and T. Poggio. “Face Recognition: Features versus Templates”. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(10):1042–1052, October 1993.

[3] C.J.C. Burges, J.I. Ben, J.S. Denker, and Y. et al. Lecun. Off line recognition of handwritten postal words using neural networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4):689–704, 1993.

[4] W. Freeman and E Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:891–906, 1991.

[5] W. Freeman and M. Roth. Orientation histogrmas for hand gesture recognition. In *International Workshop on Automatic Face- and Gesture-Recognition*, 1995.

[6] J. Garding and T. Lindeberg. Direct computation of shape cues using scale-adapted spatial derivative operators. *to appear in Int. J. of Computer Vision*, 1995.

[7] D.J. Heeger. Optical flow using spatiotemporal filters. *Int. J. of Computer Vision*, 1, 1988.

[8] D. Jones and J. Malik. Computational framework for determining stereo correspondence from a set of linear spatial filters. *Image and Vision Computing*, 10(10), 1992.

[9] M.I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 6(2), March 1994.

[10] D. Koller, J. Weber, and J. Malik. Towards realtime visual based tracking in cluttered traffic scenes. In *Proceedings of the Intelligent Vehicles Symposium*, Paris, 1994.

[11] A. Lanitis, Taylor C.J., Cootes T.F., and Ahmed T. Automatic interpretation of human faces and hand gestures using flexible models. In *International Workshop on Automatic Face- and Gesture-Recognition*, 1995.

[12] T. Leung, Burl M.C., and Perona P. Finding face in cluttered scenes using random labelled graph matching. In *Proc. Int. Conf. Computer Vision*, 1995.

[13] J. Malik and P. Perona. Detecting edges composed of steps, peaks and roofs. In *Proc. 3rd Int. Conf. Computer Vision*, 1990.

[14] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *J. Opt. Soc. Am. A*, 7(5):923–932, 1990. Also appeared as a chapter in *Computer vision: advances and applications*, Ed. R. Kasturi and R. Jain, IEEE Computer Society Press, 1991.

[15] S. Mukherjee and S. K. Nayar. Automatic generation of grbf networks for visual learning. In *Proc. Int. Conf. Computer Vision*, 1995.

[16] H. Murase and S.K. Nayar. Learning and recognition of 3-d objects from brightness images. In *Proc. AAAI Fall Symposium: Machine Learning in Computer Vision: What, Why, and How?*, 1993.

[17] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, 1994.

[18] P. Perona. Deformable kernels for early vision. *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 222–227, Maui, June 1991.

[19] R. W. Picard and T. P. Minka. Vision texture for annotation. *ACM/Springer Journal of Multimedia Systems*, 3, 1995.

[20] D. Pomerleau. *Neural Network Perception for Mobile Robot Guidance*. Kluwer Academic Pubulishing, 1994.

[21] A. Pope and Lowe D. Learning 3d object recognition models from 2d images. In *AAAI Fall Workshop on Machine Learning in Computer Vision*, 1993.

[22] D. E. Rummelhart, G. E. Hinton, and R. J. Williams. Learning Representations by Back-Propagating Errors. *Nature*, 323(9):533–536, October 1986.

[23] S. Sclaroff and A. Pentland. Modal matching for correspondence and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, June 1995.

[24] D. Shy and P. Perona. X-y separable steerable filters. Computation and Neural Systems Technical Report 33, California Institute of Technology, October 1993.

[25] P. Simard, Y. LeCun, and J. Denker. Efficient pattern recognition using a new transformation distance. In *Advances in Neural Information Processing Systems*, 1993.

[26] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[27] J. Weber and J. Malik. Robust computation of optical flow in a multi-scale differential framework. *International Journal of Computer Vision*, 14(1), 1995.

[28] Y. Xiong and S. A. Shafer. Hypergeometric filters for optical flow and affine matching. In *Proc. Int. Conf. Computer Vision*, 1995.