European Conference on Computer Vision (ECCV), 2012

V1-Inspired Features Induce a Weighted Margin in SVMs

Hilton Bristow[1,2] and Simon Lucey[2]

[1] Queensland University of Technology, Australia
[2] Commonwealth Scientific and Industrial Research Organisation, Australia
{hilton.bristow, simon.lucey}@csiro.au

Abstract. Image representations derived from simplified models of the primary visual cortex (V1), such as HOG and SIFT, elicit good performance in a myriad of visual classification tasks including object recognition/detection, pedestrian detection and facial expression classification. A central question in the vision, learning and neuroscience communities regards *why* these architectures perform so well. In this paper, we offer a unique perspective to this question by subsuming the role of V1-inspired features directly within a linear support vector machine (SVM). We demonstrate that a specific class of such features in conjunction with a linear SVM can be reinterpreted as inducing a weighted margin on the Kronecker basis expansion of an image. This new viewpoint on the role of V1-inspired features allows us to answer fundamental questions on the uniqueness and redundancies of these features, and offer substantial improvements in terms of computational and storage efficiency.

1 Introduction

One of the fundamental unanswered questions in computer vision regards how to best represent object appearance in the face of geometric and photometric distortions. The computational neuroscience community faces a parallel challenge, trying to understand how the human visual system attains a high degree of invariance whilst maintaining high selectivity. Our understanding of invariant representations stems largely from the works on the mammalian primary visual cortex (V1) [7]. Here, local object appearance and shape can be well categorised by the distribution of local edge directions, without precise knowledge of their spatial location. This is the premise behind HOG [4] and SIFT [12], among other features inspired by V1.

Jarrett *et al.* [8] showed that many V1-inspired features follow a similar pipeline of filtering an image through a large filter bank, followed by a non-linear rectification step, and finally a blurring/histogramming step.[3] Canonical features such as HOG and SIFT employ filter banks with strong selectivity to spatial frequency, orientation and scale (*e.g.* oriented edge filters, Gabor filters,

[3] We ignore photometric normalisation for brevity, but owing to its importance in the efficacy of the descriptor [8], we show how to reintroduce it later.

etc.). More recently however, weakly selective architectures such as random filters in convolutional networks have shown good performance in object classification tasks [15]. This brings into question the purpose of filters and the intrinsic properties of invariant representations in visual recognition.

From a practical perspective, most V1-inspired features use an over-complete representation based on filtered versions of images, incurring a storage cost linear in the number of filters. Whilst this seems reasonable at a glance, consider a simple example of storing 200000 50×50 images in double precision. In the case of raw pixels this amounts to only 3.72 GB of storage, a manageable figure on current desktop hardware. Filtering these images with 40 Gabor filters (5 scales and 8 orientations), commonly used in facial expression recognition [9], storage blows out to an untenable 149 GB. Strategies have been proposed to curb storage complexity, however they are largely based on heuristic subsampling or data dependent matrix factorisations that do not generalise well to new problems or datasets.

The role of V1-inspired features has been studied largely in isolation to the learning architecture used for classification. One criticism of ignoring the learning strategy when studying features is that structure unimportant to the classifier may be preserved, resulting in (i) additional computational burden, and (ii) ambiguity in representation. A fundamental motivation of our work is to consider these entities as intimately coupled. Analysing them as a whole yields insights that would not otherwise be apparent.

Contributions: We make three specific contributions in this paper:

- We show that a particular class of V1-inspired features can be rewritten as a linear function of the Kronecker expansion of an image (§2). This linear transform can be viewed as a data-independent matrix which induces a weighted margin in max-margin learning (§4).
- We postulate that a lower dimensional matrix should be able to approximate the same prior but at a substantially lower computational cost (§3), and empirically show that this is indeed the case (§5). This reduces both the storage complexity of the data and the time taken to train the resulting SVM, in theory enabling training on significantly larger datasets with little loss in classification performance.
- We demonstrate that reinterpreting the role of V1-inspired features as a weighted margin reveals some valuable insights into: (i) the uniqueness of the filters commonly used in these architectures, and (ii) the capacity of a linear SVM using V1-inspired features tending towards a quadratic kernel SVM (§4).

Empirical results are detailed in §5 across a number of visual classification tasks. Matlab code is available at `hiltonbristow.com/software`.

Prior Art: Ashraf *et al.* [1] originally explored the link between feature extraction and a weighted margin for visual classification tasks. By restricting their scope to linear features, they view filtering as a weighted margin on the data in the Fourier domain. We instead explore an inherently nonlinear embedding,

more akin to current models of early biological vision. Due to the high dimensionality of the resulting problem (not encountered by Ashraf *et al.* by virtue of the convolution theorem), we seek to explicitly represent the feature maps in a lower dimensional space.

Vedaldi and Zisserman [17] and Bo *et al.* [3] both propose methods for explicitly representing kernels so lower dimensional approximations can be found, independent of data. The appeal of both approaches is the speedup in training and evaluation time that can be enjoyed by learning a linear rather than kernel SVM. Vedaldi considers the case of approximately representing the implicit feature associated with additive kernels (*i.e.* kernels useful for matching histograms) whilst Bo considers the case of incorporating preprocessed oriented edge energies, along with spatial position and colour directly into the kernel function. Our method, by contrast, relates the raw image pixel intensities directly with the feature pipeline. By having this direct relationship we can gain fundamental insights into the importance of particular architectures and redundancies in V1-inspired features to actual classification performance within a linear SVM.

2 Problem Formulation

Feature representations such as HOG and SIFT, and other more exotic architectures such as convolutional networks, crudely approximate the function of V1 complex cells. They commonly involve (1) edge orientation detection, (2) nonlinear rectification to increase edge bandwidth and discard edge step direction, (3) contrast normalisation to remove photometric variation, and (4) downsampling/pooling to histogram the resulting edge directions. We show now that a specific form of this pipeline can be expressed through a mixture of Kronecker products and convolution operations.

V1 Form: Given a vectorised input image of intensities $\mathbf{x} \in \Re^D$, coarse edge orientation detection can be cast as a 2D convolution operation with a bank of oriented filters, $\{\mathbf{g}_f\}_{f=1}^F$. A simple pointwise quadratic function, cast here as the Hadamard product ($\odot$) between two filtered versions of an image fulfils both the rectification and nonlinearity steps. Spatial pooling is achieved through 2D convolution ($*$) of the rectified response with a boxcar filter $\mathbf{b}$. The feature map $\Phi(\mathbf{x})$ can thus be expressed as,

$$\Phi(\mathbf{x}) = [\Phi_1(\mathbf{x}), \Phi_2(\mathbf{x}), \ldots, \Phi_F(\mathbf{x})]^T \tag{1}$$

where,

$$\Phi_f(\mathbf{x}) = \mathbf{b} * [(\mathbf{g}_f * \mathbf{x}) \odot (\mathbf{g}_f * \mathbf{x})] \,. \tag{2}$$

This particular architecture has taken the name "convolutional square pooling" and has shown good performance across a range of tasks [2]. Many variations on this feature pipeline have been advocated in literature previously, such as the use of max rather than average pooling, a sigmoid nonlinear function after rectification, and the estimation of orientation energies using an arctangent function on the horizontal and vertical edge energies. Motivations for the specific

form used in this paper are that it: (i) is similar in philosophy to these other variants, (ii) still offers excellent performance when applied to a variety of classification tasks, and (iii) has greater flexibility in manipulation, thus leading to our re-interpretation as a weighted margin within a linear SVM.

Kronecker Form: Manipulation of the form in Equation 2 is limiting due to the Hadamard product ($\odot$). By defining a relation between the Hadamard and Kronecker product ($\otimes$) however, we can exploit some properties of the latter.

Theorem 21 *The Hadamard product between any two equal size vectors* $\mathbf{x}_i \in \Re^D$ *and* $\mathbf{x}_j \in \Re^D$ *can be written as,*

$$\mathbf{x}_i \odot \mathbf{x}_j = \mathbf{M}(\mathbf{x}_i \otimes \mathbf{x}_j) \tag{3}$$

such that $\mathbf{M} \in \Re^{D \times D^2}$. We can explictly define $\mathbf{M}$ as,

$$\mathbf{M} = \begin{bmatrix} \mathbf{e}_1^T \otimes \mathbf{e}_1^T \\ \vdots \\ \mathbf{e}_D^T \otimes \mathbf{e}_D^T \end{bmatrix} \tag{4}$$

given that $\mathbf{e}_i \in \Re^D$ is a vector of zeros with 1 at the i-th element.

Replacing 2D convolution operations (*e.g.* $\mathbf{g} * \mathbf{x}$) with convolution matrices (*e.g.* $\mathbf{G}\mathbf{x}$) and applying Theorem 21 to Equation 2, the response to a single filter can be written as,

$$\begin{aligned} \Phi_f(\mathbf{x}) &= \mathbf{B}\mathbf{M}[(\mathbf{G}_f\mathbf{x}) \otimes (\mathbf{G}_f\mathbf{x})] \\ &= \mathbf{B}\mathbf{M}(\mathbf{G}_f \otimes \mathbf{G}_f)(\mathbf{x} \otimes \mathbf{x}) \; . \end{aligned} \tag{5}$$

The full response to a bank of filters can be written as,

$$\Phi(\mathbf{x}) = \mathbf{L}(\mathbf{x} \otimes \mathbf{x}) \tag{6}$$

where,

$$\mathbf{L} = \begin{bmatrix} \mathbf{B}\mathbf{M}(\mathbf{G}_1 \otimes \mathbf{G}_1) \\ \vdots \\ \mathbf{B}\mathbf{M}(\mathbf{G}_F \otimes \mathbf{G}_F) \end{bmatrix} \; . \tag{7}$$

For two V1-inspired feature maps $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$, the kernel is defined as the inner product of the maps,

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle \; . \tag{8}$$

Since the feature maps have a closed form expression, the kernel can be written explicitly as,

$$\Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) = (\mathbf{x}_i \otimes \mathbf{x}_i)^T \mathbf{L}^T \mathbf{L} (\mathbf{x}_j \otimes \mathbf{x}_j) \tag{9}$$

$$= (\mathbf{x}_i \otimes \mathbf{x}_i)^T \mathbf{S} (\mathbf{x}_j \otimes \mathbf{x}_j) \tag{10}$$

where $\mathbf{L} \in \Re^{DF \times D^2}$ implies that the rank of $\mathbf{S}$ is at most DF. Thus after some manipulation, the form of V1-inspired features can be rearranged with the filter and data terms isolated. This suggests that the feature map is only dependent on the *joint* response from the filters and blur kernels, and that the weighting matrix $\mathbf{S}$ can be completely precomputed in the absence of data.

3 Computational Efficiency

Whilst $\mathbf{S}$ is rank deficient, its high dimensionality (*i.e.* $D^2 \times D^2$) makes it infeasible to work with directly. In practice, we wish to find a matrix of rank $K \ll DF$ that makes a good approximation to $\mathbf{S}$ whilst never explicitly computing $\mathbf{S}$ or its eigenvectors.

Indirectly Computing the Eigenvectors: From the thin singular value decomposition (SVD) of $\mathbf{L}$,

$$\mathbf{L} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T \tag{11}$$

the right singular vectors $\mathbf{V} \in \Re^{DF \times D^2}$ correspond to the eigenvectors of $\mathbf{L}^T\mathbf{L} = \mathbf{S} \in \Re^{D^2 \times D^2}$, and the left singular vectors $\mathbf{U} \in \Re^{DF \times DF}$ to the eigenvectors of $\mathbf{L}\mathbf{L}^T$ which we denote $\mathbf{S}^* \in \Re^{DF \times DF}$. The eigenvectors $\mathbf{V}$ of $\mathbf{S}$ can be found efficiently by first computing the eigenvectors $\mathbf{U}$ of $\mathbf{S}^*$, then from Equation 11,

$$\mathbf{V}^T = (\boldsymbol{\Sigma}^T\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma})^{-1}(\mathbf{U}\boldsymbol{\Sigma})^T\mathbf{L} \tag{12}$$

$$= \boldsymbol{\Sigma}^{-1}\mathbf{U}^T\mathbf{L}\,. \tag{13}$$

Letting $\hat{\mathbf{U}}, \hat{\boldsymbol{\Sigma}}, \hat{\mathbf{V}}$ be components of the SVD of $\mathbf{L}$ with the K largest magnitude singular values preserved, and $\hat{\Phi}(.)$ the corresponding low dimensional feature map, then

$$\mathbf{S} \approx \hat{\mathbf{V}}\hat{\boldsymbol{\Sigma}}^2\hat{\mathbf{V}}^T\,. \tag{14}$$

The distribution of singular values in $\mathbf{S}^*$ suggests how well a rank reduction will preserve the information in $\mathbf{S}$. Figure 1 shows the eigenspectra of typical $\mathbf{S}$ matrices constructed from a number of filter representations. The spectra hint at the significant redundancies that can be exploited to reduce storage and computational costs associated with computing the low rank feature map. The $\sim \frac{1}{f}$ slope of the spectra correlates well with statistics observed in natural images.

Applying the Eigenvectors: An explicit representation of $\mathbf{S}$ is unnecessary since the goal is to find an efficient closed-form expression for the feature maps. Thus,

$$\Phi(\mathbf{x}_i)^T\Phi(\mathbf{x}_j) \approx (\mathbf{x}_i \otimes \mathbf{x}_i)^T\hat{\mathbf{V}}\hat{\boldsymbol{\Sigma}}^2\hat{\mathbf{V}}^T(\mathbf{x}_j \otimes \mathbf{x}_j) \tag{15}$$

and since the kernel is separable, a single feature map in isolation becomes,

$$\hat{\Phi}(\mathbf{x}_i) = \hat{\boldsymbol{\Sigma}}\hat{\mathbf{V}}^T(\mathbf{x}_i \otimes \mathbf{x}_i)\,. \tag{16}$$

Substituting Equation 13 into Equation 16 gives,

$$\hat{\Phi}(\mathbf{x}_i) = \hat{\mathbf{U}}^T\mathbf{L}(\mathbf{x}_i \otimes \mathbf{x}_i)\,. \tag{17}$$

Whilst $\mathbf{L}$ is sparse for compact support filters, storage in memory quickly becomes prohibitive with increasing image size. For our earlier example of a 50×50 pixel input and 40 filters with 20×20 pixel support, storing the full $\mathbf{L}$ matrix will

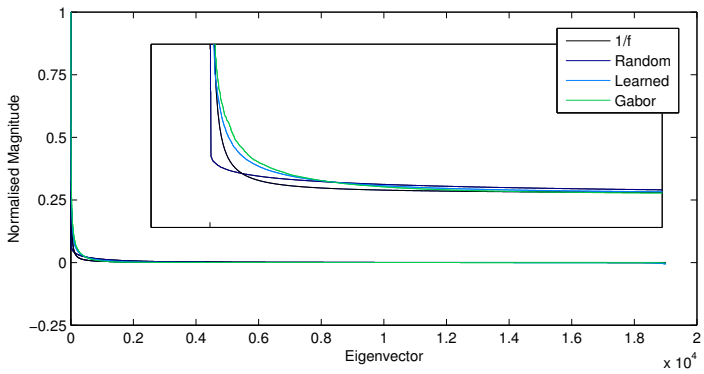


Fig. 1. The eigenspectrum of $\mathbf{S}$ for a number of filter representations. The $\sim \frac{1}{f}$ distribution suggests a low rank approximation could be found, which preserves most of the variance in $\mathbf{S}$. The magnified region shows in greater detail the energy distribution of the largest eigenvalues. The learned filters (using the method of [10]) have the most compact energy spectrum, followed by the Gabor filters, with the random filters having the broadest spectrum.

require on the order of 657 GB. We know however, that the joint portion $\mathbf{L}(\mathbf{x} \otimes \mathbf{x})$ can be efficiently computed using the original method of convolutions via,

$$\mathbf{L}(\mathbf{x} \otimes \mathbf{x}) = \begin{bmatrix} \mathbf{b} * [(\mathbf{g}_1 * \mathbf{x}) \odot (\mathbf{g}_1 * \mathbf{x})] \\ \vdots \\ \mathbf{b} * [(\mathbf{g}_F * \mathbf{x}) \odot (\mathbf{g}_F * \mathbf{x})] \end{bmatrix} . \tag{18}$$

By taking this approach, only $\hat{\boldsymbol{\Sigma}}$ and $\hat{\mathbf{U}}$ ever need be explicitly computed. For the example above, storing $\hat{\mathbf{U}}$ of rank $K = D$ will consume only 1.86 GB of memory.

Computing the feature map of Equation 1 incurs a cost of $O(DF \log D)$ operations and storage $O(DF)$. Computing the proposed feature map of Equation 17 incurs an added $O(KDF)$ operations but storage is only $O(K)$ where $K \ll DF$. Our feature map therefore realises a tradeoff between computational complexity and storage complexity, and results in a representation that is manageable for large amounts of high dimensional data, and as shown following, tractable in time when learning an SVM.

4 V1-Inspired Features & SVMs

Support vector machines have seen extensive use in visual classification tasks, and have proved particularly successful in tasks involving V1-inspired features [4]. Linear SVMs have a number of inherent advantages over kernel SVMs: (i) faster learning times, (ii) the ability to learn from larger datasets, (iii) low computation cost during evaluation as the summation over support weights and vectors can

be pre-computed, and most importantly, (iv) for some applications identical if not superior performance to nonlinear kernels (*e.g.*, RBF, polynomial, tanh) [5].

Given a set of training features and labels $\{\Phi(\mathbf{x}), y_i\}_{i=1}^{l}$, $\Phi(\mathbf{x}) \in \Re^{DF}$, $y_i \in \{+1, -1\}$ a linear SVM attempts to find the solution to the following optimisation problem,

$$\begin{aligned} \min_{\mathbf{w}, \xi_i \geq 0} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i\mathbf{w}^T\Phi(\mathbf{x}_i) \geqslant 1 - \xi_i, \quad i = 1 \ldots l \end{aligned} \tag{19}$$

where C is a penalty parameter and ξ_i are the slack variables introduced to offset the effects of outliers in the final solution.[4]

It is well understood in SVM literature that the $\mathbf{w}^T\mathbf{w}$ term in Equation 19 is inversely proportional to the margin of the solution. Maximising this margin is central to the generalisation properties of SVMs. The type of margin being maximised in this feature space is based on an unweighted (i.e. Euclidean) distance. Inspired by [1], however, we can demonstrate that an equivalent form of Equation 19 can be obtained by solving,

$$\begin{aligned} \min_{\mathbf{v}, \xi_i \geq 0} \quad & \frac{1}{2}\mathbf{v}^T\mathbf{S}^{-1}\mathbf{v} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to} \quad & y_i\mathbf{v}^T(\mathbf{x}_i \otimes \mathbf{x}_i) \geqslant 1 - \xi_i, \quad i = 1 \ldots l \end{aligned} \tag{20}$$

where the role of features has been completely subsumed into the weighted margin term $\mathbf{v}^T\mathbf{S}^{-1}\mathbf{v}$. The solutions to Equation 19 ($\mathbf{w} \in \Re^{DF}$) and 20 ($\mathbf{v} \in \Re^{D^2}$) are related by $\mathbf{w} = \mathbf{L}\mathbf{v}$ where $\mathbf{L} \in \Re^{DF \times D^2}$ is previously defined in Equation 10. A key realisation here is that the role of the features is completely described as a margin manipulation – the weighting term is only applied to the margin term and not the data term.

Capacity of the Classifier: This result links well with previous work of Shivaswamy and Jebara [16] concerning what "type" of margin should be maximised during the estimation of a max margin classifier such as an SVM. In this work Shivaswamy and Jebara discussed the importance of selecting the "correct" kind of margin when learning an SVM and how maximising a margin based on Euclidean distance might not always be the best choice in terms of classifier generalisation. In fact when one sets $\mathbf{S} = \mathbf{I}$ then the solution to the objective in Equation 20 reverts to a classical kernel SVM since,

$$(\mathbf{x}_i \otimes \mathbf{x}_i)^T\mathbf{I}(\mathbf{x}_j \otimes \mathbf{x}_j) = (\mathbf{x}_i^T\mathbf{x}_j)^2 \tag{21}$$

where the kernel employed is a homogeneous second-order polynomial. Understanding how V1-inspired features improve the capacity of a linear SVM will

[4] The bias b is accounted for in $\mathbf{w} \leftarrow [\mathbf{w}^T, b]$ by $\Phi(\mathbf{x}) \leftarrow [\Phi(\mathbf{x})^T, 1]^T$ but is omitted here for clarity.

become important in our experiments section (§5). It is more reasonable to compare the classification performance of V1-inspired features to a quadratic kernel than to (a linear kernel on) raw pixels, since the latter has substantially less capacity.

Complexity in SVM Training and Prediction: When training an SVM classifier, we modify Equation 19 to instead use our low dimensional feature map $\hat{\Phi}(\mathbf{x})$, which yields an optimisation over a lower (K) dimensional $\hat{\mathbf{w}}$,

$$\begin{aligned} \min_{\hat{\mathbf{w}},\xi_i \geq 0} \ & \frac{1}{2}\hat{\mathbf{w}}^T\hat{\mathbf{w}} + C\sum_{i=1}^{l}\xi_i \\ \text{subject to } \ & y_i\hat{\mathbf{w}}^T\hat{\Phi}(\mathbf{x}_i) \geqslant 1-\xi_i, \quad i = 1\dots l\ . \end{aligned} \tag{22}$$

Further to the space-time tradeoffs of §3, our method also realises a preprocessing-learning tradeoff, which has benefits when training large datasets and enumerating over different training schemes. We do not, however, have to make that same tradeoff during prediction. By taking advantage of the form of $\hat{\Phi}(\mathbf{x})$ from Equation 13, we can promote $\hat{\mathbf{w}}$ from a K dimensional space to a DF dimensional space through $\mathbf{w} = \mathbf{U}\hat{\mathbf{w}}$, such that for a vectorised test image $\mathbf{x}_i$,

$$\mathbf{w}^T\Phi(\mathbf{x}_i) \equiv \hat{\mathbf{w}}^T\hat{\Phi}(\mathbf{x}_i) \tag{23}$$

where $\Phi(\mathbf{x})$ is the original feature map of Equation 1.

Uniqueness of Filters: The structured form of the $\mathbf{S}$ matrix gives us an insight into the role of filters in the margin manipulation, specifically the uniqueness of the filter responses and their joint contribution to the invariant representation. The matrix $\mathbf{S} = \mathbf{L}\mathbf{L}^T$ can be represented as a concatenation of $F \times F$ submatrices,

$$\begin{aligned} \mathbf{L}_i\mathbf{L}_j^T &= \mathbf{B}\mathbf{M}(\mathbf{G}_i \otimes \mathbf{G}_i)(\mathbf{G}_j \otimes \mathbf{G}_j)^T\mathbf{M}^T\mathbf{B}^T \\ &= \mathbf{B}\mathbf{M}(\mathbf{G}_i\mathbf{G}_j^T) \otimes (\mathbf{G}_i\mathbf{G}_j^T)\mathbf{M}^T\mathbf{B}^T\ . \end{aligned} \tag{24}$$

From this form one can see that the role of the individual filters in this form is not unique since $\mathbf{G}_i\mathbf{A}\mathbf{A}^{-1}\mathbf{G}_j^T = \mathbf{G}_i\mathbf{G}_j^T$ where $\mathbf{A}$ is any arbitrary full rank transform matrix. Further, it is possible to show that the interaction of these filters $\mathbf{G}_i\mathbf{G}_j^T$ is unique up to a sign ambiguity.[5] Finally, it is possible to see where spatial invariance stems from in the weighting matrix $\mathbf{S}$ since for $i = j$ local phase is lost, and when $i \neq j$ only relative phase is preserved.

5 Experiments

Here we evaluate our methodology on MNIST, Caltech 101 and Cohn Kanade+ datasets to illustrate the applicability of our method to a range of computer

[5] Since $\mathbf{x} \otimes \mathbf{x} = \text{vec}(\mathbf{x}\mathbf{x}^T)$ where we know through the SVD that one can recover $\mathbf{x}$ up to a sign. Here we assume $\mathbf{x} = \text{vec}(\mathbf{G}_i\mathbf{G}_j^T)$ from Equation 24.

vision domains. Since we have already motivated the scalability of our method to large datasets through our initial thought experiment, we instead focus on showing our rank reduced features remain competitive on established benchmarks. We mimic the experimental setup of other authors who have used similar V1-inspired features.

For each of the experiments we consider 5 cases: (i) rank DF $\mathbf{S}$ matrix, (ii) rank D $\mathbf{S}$ matrix, (iii) random filters rather than frequency and orientation selective filters, (iv) $\mathbf{S} \rightarrow \mathbf{I}$ corresponding to a quadratic kernel on the pixels, and (v) pixels. Where we say $\text{rank}(\mathbf{S}) = D$, we take D to be the dimensionality of the vectorised input image. In the case of frequency and orientation selective filters, we use a bank of log Gabor filters. In the case of random filters, we use the same number of filters as the Gabor case, and ensure that each filter has zero mean and unit norm. For each convolution, we only keep the central area that is the same size as the input image.

Reintroducing Photometric Normalisation: Jarrett *et al.* [8] show that rectification and photometric normalisation are the single most important factors in improving the performance of a recognition system. We too note this to be the case, especially in images exhibiting large photometric variation, as observed in natural images (*e.g.* Caltech 101).

Given a pointwise processing stage $\Psi(.)$ that maps $\mathfrak{R}^{DF} \rightarrow \mathfrak{R}^{DF}$ Equation 17 can be extended to

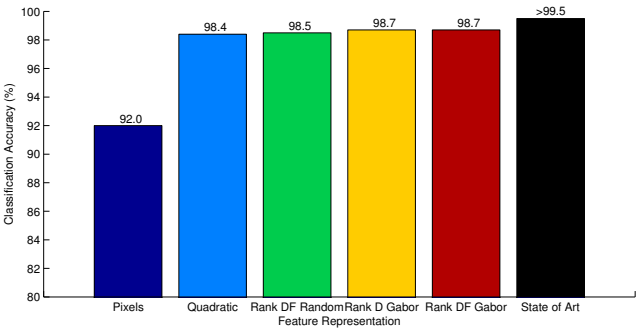
$$\Phi(\mathbf{x}) = \mathbf{U}^T \Psi\big(\mathbf{L}(\mathbf{x} \otimes \mathbf{x})\big) \; . \tag{25}$$

This allows us to include mid-processing such as photometric normalisation without loss of generality.

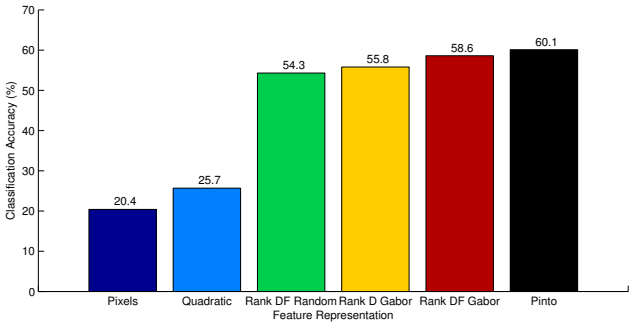
MNIST: MNIST is a handwritten character recognition dataset containing 60000 training examples and 10000 test examples of the characters $0-9$. Each character is roughly centred in a 28×28 window and quantised to 8-bit grayscale. Although MNIST is an ageing dataset, LeCun's convolutional network architecture – which for a single layer closely follows our parametric form – has shown particularly impressive performance at the task [8].

We use 48 Gabor filters at 12 orientations and 4 frequencies each of size 28×28, and a boxcar filter of size 3×3. We remove the photometric normalisation step from our model, but preprocess each image by power normalisation. Due to the large number of training data and the resulting descriptor dimensionality of 37632, we opt to train the resulting linear SVM in the prime. Average classification performance is shown in Figure 2(a).

Caltech101: Caltech 101 is a "natural" object recognition dataset containing 101 object classes, each with $40-800$ instances. The objects are roughly centred and in similar poses, though vary in appearance. Pinto has pointed to a number of flaws in the dataset and argues that it lacks true real-world variability, and supports his claims by achieving good performance with a simple biologically motivated feature representation [14]. We mimic his setup and achieve similar performance whilst illustrating some advantages of our method.



(a) MNIST



(b) Caltech 101

Fig. 2. Average classification performance across all classes of the (a) MNIST, and (b) Caltech 101 dataset for different feature descriptor representations. (Pixels) raw pixels, (Quadratic) quadratic kernel on raw pixels, (Rank DF Random) the full $\mathbf{S}$ matrix constructed from random filters, (Rank D Gabor) a low rank approximation to the full $\mathbf{S}$ matrix constructed from Gabor filters, (Rank DF Gabor) the full $\mathbf{S}$ matrix constructed from Gabor filters, (State of Art) State of the Art benchmark for MNIST taken from a survey of 60 algorithms, (Pinto) the reference method of Pinto [14].

We use 92 Gabor filters at 16 orientations and 6 scales each of size 43×43, and a boxcar filter of size 17×17. We preprocess the images by resizing and cropping each to fit a 150×150 pixel box. We modify our model to include a downsampling matrix which subsamples each filter response by a factor of 5 (to a 30×30 image). Average classification performance is shown in Figure 2(b).

PCA on Responses: To deal with the "curse of dimensionality", many papers have been devoted to finding low dimensional approximations to descriptors using PCA, LDA or nonlinear dimensionality reduction methods [11,18,6]. These methods have two inherent problems: the reduction is data dependent and needs to be recomputed for each new set of data, and the reduction must occur in the original dimensionality and may not be feasible in time or space.

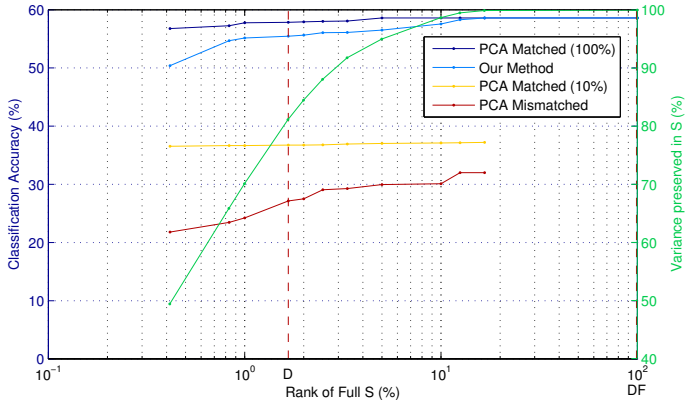


Fig. 3. A comparison of dimensionality reduction techniques on the Caltech 101 dataset. Performance is measured as average classification accuracy across classes as a function of descriptor dimensionality. (PCA Matched 100%) PCA loadings calculated from the entire training set. (Our Method) Dimensionality reduction using a low rank approximation to $\mathbf{S}$. (PCA Matched 10%) PCA loadings calculated from 10% of the training set, with equal class representation. (PCA Mismatched) PCA loadings calculated from Cohn Kanade+ dataset. The green curve shows the variance of $\mathbf{S}$ preserved as a function of the rank. A descriptor of rank D not only models 80% of the variance in the original DF representation, but achieves similar classification performance. PCA consistently performs $\sim 4\%$ better, but only in well-matched conditions.

Equation 17 suggests that the matrix $\mathbf{U}$ acts to transform the feature onto a low rank orthonormal basis which preserves the highest modes of variance - in essence PCA. The advantages of this approach are twofold: the reduction can be precomputed in the absence of data and the reduction is based on the filter components that are likely to be discriminative rather than the observed modes of deformation specific to each training set. Figure 3 shows the classification performance of our method as a function of feature length, using the Caltech 101 setup with Gabor filters. A number of PCA schemas are shown for comparison. PCA Matched (10%) and PCA Mismatched show how PCA fails to generalise when the data used to calculate the loadings either does not span the full extent of geometric variability in the training and testing sets, or is from a different domain entirely. Our method suffers neither of these drawbacks, yet approaches the performance of PCA with loadings calculated from the full training set (PCA Matched (100%)).

Cohn Kanade+: Cohn Kanade+ is an expression recognition dataset consisting of 68-point landmark, broad expression and FACS labels across 123 subjects and 593 sequences. Each sequence varies in length and captures the neutral expression in the first frame and the peak formation of facial expression in the last. We follow the experimental setup of Lucey *et al.* [13], however we consider only the broad expressions and discard the AU labels.

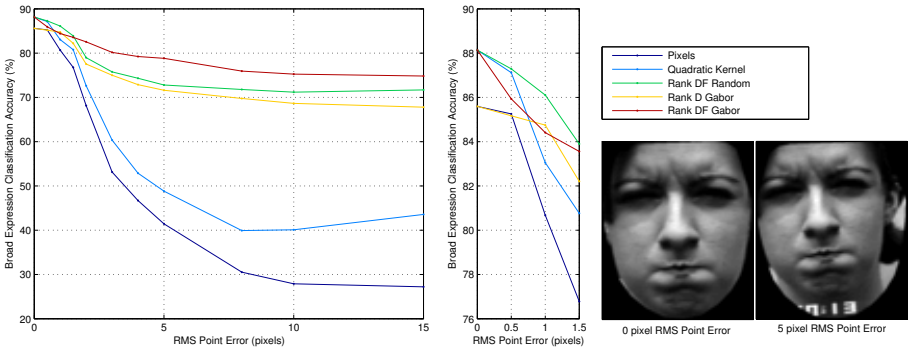


Fig. 4. Classification performance on Cohn Kanade+ broad expressions as a function of increasing registration error. Feature representations have better robustness to registration error. The central magnified panel shows that with perfect registration, the rank DF representations converge to a quadratic kernel and the rank D representations converge to (a linear kernel on) raw pixels. A quadratic kernel represents the inherent capacity of our V1-like feature parameterisation in a linear SVM learning scheme.

We register each face to a canonical geometric template then measure classification accuracy across all expressions with increasing registration error. Results are shown in Figure 4.

6 Discussion

The application of V1-inspired features can be reinterpreted as a weighted margin on the on the Kronecker basis expansion of an image. This insight becomes clearer in Equation 20 when viewed in the context of training a linear SVM. The prior on the margin is a global spatial weighting on the responses to oriented edge filters, which appear to encode some phase invariance along with relationships between frequency and orientation bands. The Cohn Kanade+ dataset was used to explore the weighted margin insight under known ground-truth geometric distortions. The results of Figure 4 reveal a pervasive insight. Image features give better robustness to registration error than raw pixel representations. With perfect registration however, the performance of rank DF representations converge to a quadratic kernel on the raw pixels, whilst the performance of the rank D representations converge to (a linear kernel on) raw pixels.

This suggests that in the absence of geometric noise, the filter prior over the data has no influence. The process of *gaining invariance* importantly does not improve performance outright; but rather only in the face of geometric mismatch. With perfect registration the class separation is sufficiently large that a prior on the margin has no effect on the discriminability of the decision hyperplane. It is only with increasing registration error and increasing nonlinearity of the true

decision boundary that the prior helps guide the separating hyperplane to a good solution.

Casting the prior in a lower dimensional space whilst retaining good performance shows the function of features is more about generalisation than increasing classifier capacity. This prior encodes information important in gaining invariance to geometric variability with substantially lower capacity than a quadratic kernel. Nonetheless, the results of experiments on MNIST (Figure 2(a)) and Caltech 101 (Figure 2(b)) show that for a 60-fold decrease in dimensionality, the rank D approximation to $\mathbf{S}$ suffers only a $0-3\%$ decrease in performance.

Insights aside, the crux of the rank reduction lies in its relationship to and advantage over regular PCA. Because PCA is data dependent, it relies on an explicit representation of the entire training set, and a strong affinity between the observed geometric variability in the training and testing sets. As the amount and availability of crowd-sourced data increases, so too does the need for data-independent dimensionality reduction schemes. The $\mathbf{S}$ matrix is data agnostic and a rank reduction on this matrix is equivalent to an optimisation over the most important frequency components and their spatial support. Further, we show in Figure 3 that this approach is comparable with PCA. In essence, our choice of filters conveys our intuition about what spatial and frequency content is semantically important in images. A rank reduction on $\mathbf{S}$ acts to preserve the most important parts of this prior.

In light of this, we end by making two comments: (i) the choice of filters is still important as they constitute an assumed prior over the image statistics, however (ii) rather than expressing the prior indirectly through filters (which are not unique), we should consider treating the application of V1-inspired features as a machine learning task with a prior on natural image statistics and directly optimise for a weighted margin (which *is* unique) using conventional and well-established machine learning techniques.

7 Conclusions and Future Work

This paper has presented a new form for V1-inspired features, with filter terms decoupled from data terms. By integrating the learning strategy into the feature design, we reveal that the filter matrix prior acts to weight the margin in an SVM with an implicit quadratic kernel capacity. We speculated from the redundant eigenspectrum of this matrix that a low dimensional approximation to this matrix should be able to approximate the same margin manipulation but at greatly reduced computational cost, and showed this to be the case across a range of visual classification tasks. This approach enables a data independent dimensionality reduction scheme, appropriate for large-scale learning.

This work has freed the V1-inspired feature from its canonical parametrisation into a form more readily accepted by existing machine learning techniques. This unleashes a wealth of new questions about the optimality of V1-inspired features, the structure of $\mathbf{S}$ and the "best" weighting matrix.

Acknowledgements: Dr Simon Lucey is the recipient of an Australian Research Council Future Fellowship (project number FT0991969).

References

1. A. B. Ashraf, S. Lucey, and T. Chen. Reinterpreting the application of Gabor filters as a manipulation of the margin in linear support vector machines. *Pattern Analysis and Machine Learning*, 32(7):1335–41, July 2010. 2, 7
2. J. Bergstra, G. Desjardins, P. Lamblin, and Y. Bengio. Quadratic polynomials learn better image features. Technical report, Universite de Montreal, 2009. 3
3. L. Bo, X. Ren, and D. Fox. Kernel Descriptors for Visual Recognition. *Advances in Neural Information Processing Systems*, pages 1–9, 2010. 3
4. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition*, pages 886–893, 2005. 1, 6
5. R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008. 7
6. I. Fodor. A survey of dimension reduction techniques. Technical report, Lawrence Livermore National Laboratory, 2002. 10
7. D. Hubel and T. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106, 1962. 1
8. K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? *International Conference on Computer Vision*, pages 2146–2153, Sept. 2009. 1, 9
9. M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311, 1993. 2
10. H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. *Advances in Neural Information Processing Systems*, 19:801, 2007. 6
11. C. Liu and H. Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image Processing*, 11(4):467–76, Jan. 2002. 10
12. D. Lowe. Object recognition from local scale-invariant features. *International Conference on Computer Vision*, pages 1150–1157 vol.2, 1999. 1
13. P. Lucey, S. Lucey, and J. Cohn. Registration invariant representations for expression detection. *International Conference on Digital Image Computing: Techniques and Applications*, (i):255–261, 2010. 11
14. N. Pinto, D. D. Cox, and J. J. DiCarlo. Why is real-world visual object recognition hard? *PLoS computational biology*, 4(1):e27, Jan. 2008. 9, 10
15. A. Saxe, P. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Ng. On random weights and unsupervised feature learning. *Advances in Neural Information Processing Systems*, pages 1–9, 2010. 2
16. P. Shivaswamy and T. Jebara. Relative margin machines. *Advances in Neural Information Processing Systems*, 21:1–8, 2008. 7
17. A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *Computer Vision and Pattern Recognition*, (iii):3539–3546, 2010. 3
18. M. Yang and L. Zhang. Gabor feature based sparse representation for face recognition with Gabor occlusion dictionary. *European Conference on Computer Vision*, pages 448–461, 2010. 10