Learning the Discriminative-Power Invariance Trade-Off for Image Classification

Manik Varma
Microsoft Research
India

Debajyoti Ray
Gatsby Unit
University College London

The Image Categorization Problem

Novel image to be classified

?

=

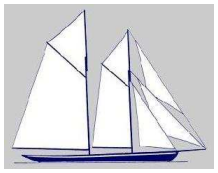
Chair

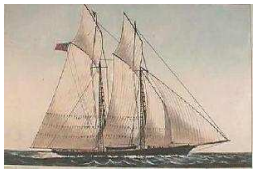




Schooner

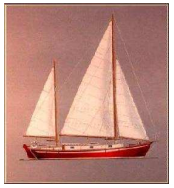
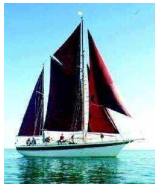






Ketch




Taj





Panda





Labelled images comprise training data

The Image Categorization Problem

The standard approach while tackling such problems

- 1st Stage – Feature Extraction
  - Extract a compact and relevant set of feature descriptors
  - Descriptors must be good at distinguishing objects from different categories
  - Descriptors must be invariant to changes in imaging conditions and intra-category variations
- 2nd Stage – Classification
  - Build accurate and fast classification algorithms

The Discriminative-Power Invariance Trade-Off

- Many hand-crafted feature descriptors have been proposed such as SIFT, Patches, Spin images, Filter banks, etc.
- Looking past the initial dissimilarities what really distinguishes one descriptor from another is the trade-off that is made between discriminative power and invariance.
    - Vanilla patches have very high discriminative power but little invariance.
    - The constant descriptor has complete invariance but no discriminative power.
- Each descriptor places itself somewhere in this spectrum according to its belief about the optimal trade-off

The Discriminative-Power Invariance Trade-Off

- However, there is no single optimal level of the trade-off for the general classification task
- Every specific classification problem can potentially require a different level of the trade-off.
- The trade-off depends on the variation in the classes, the amount of training data available and prior knowledge.
- As such, no single descriptor, which achieves a fixed level of the trade-off, can be suitable for all situations.

Example: Classifying 6 versus 9

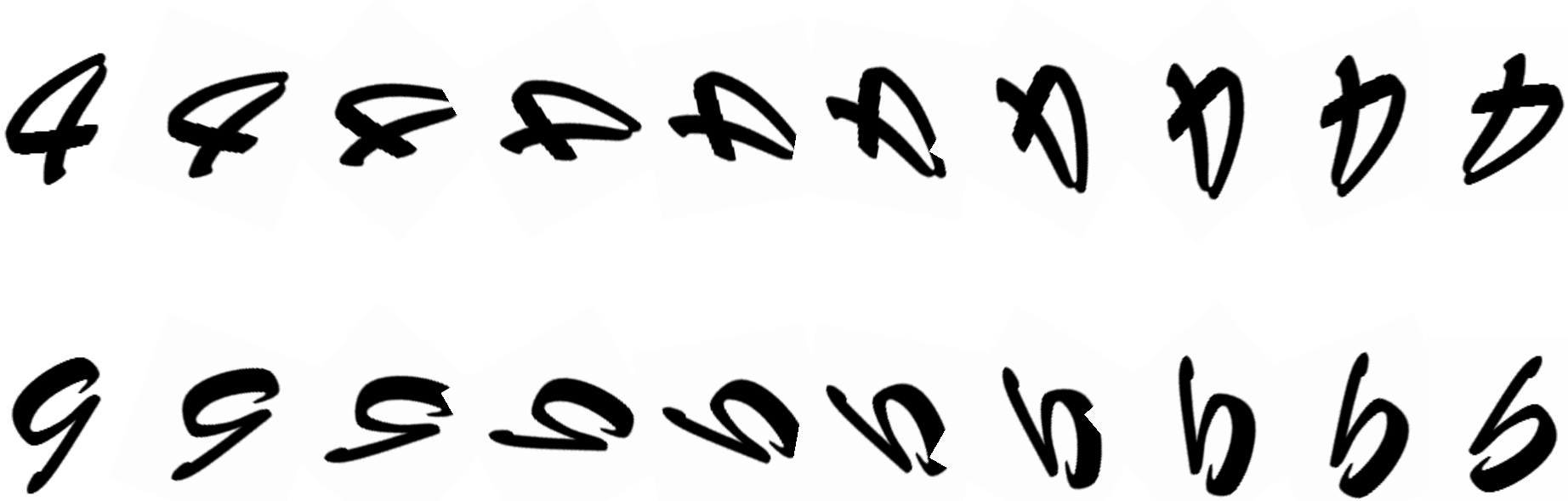
- Do not want 180° rotational invariance.

Example: Classifying 4 versus 9

- Do want 180° rotational invariance.

Example: Classifying 4 versus 9

• Don’t want invariance when a lot of training data is available as even nearest neighbour matching will do well.



• Even if an optimal descriptor could be hand-crafted for this task, it might no longer remain optimal as the training set size is varied.

Outline

- Our objective is to learn an optimal descriptor for a specified classification/matching task given training data and prior knowledge.
- This is achieved by learning the optimal trade-off between discriminating power and invariance for the given task.
- The problem is tackled in the SVM framework by learning the kernel matrix corresponding to the trade-off.
- We present comparative results on the Oxford flowers, UIUC textures and Caltech 101 databases.

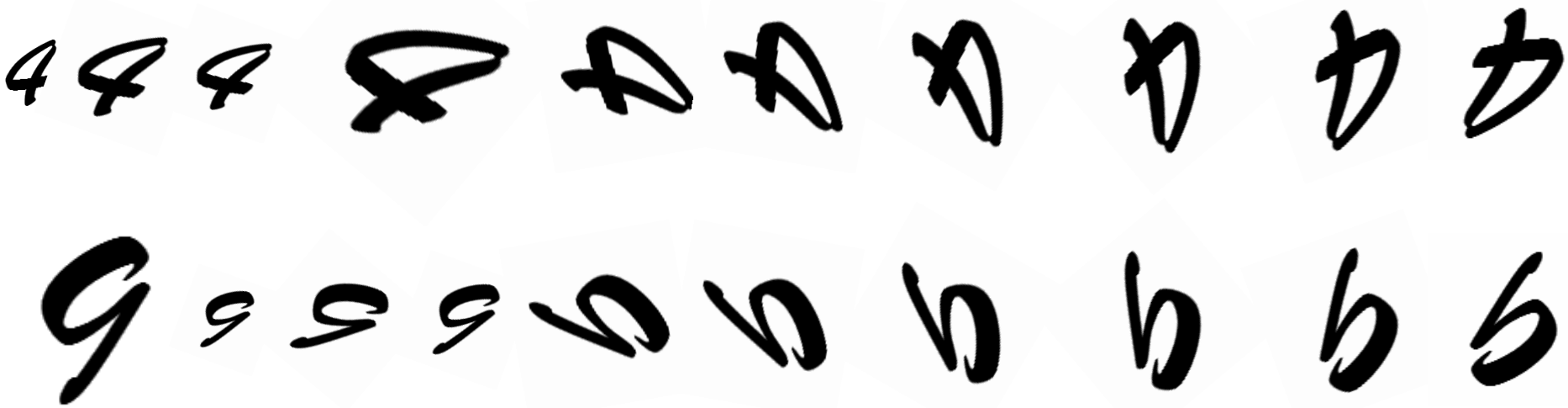
Learning the Trade-Off

- We can often get a rough sense of the level of invariance required by visual inspection alone.

- Rotation invariance should suffice.

Learning the Trade-Off

• However, pinpointing the exact trade-off level can be much harder.



• Just rotation or scale invariance will not suffice.

• Affine invariance will be too much as discriminative power will be lost and similar looking 4s and 9s confused.

Ideal Solution

- Every descriptor should have a continuously tuneable meta parameter controlling its level of invariance.
- Generate an infinite set of "base" descriptors spanning the entire range of the trade-off
- Select the single base descriptor which achieves the ideal level of the trade-off
- For the ideal descriptor, intra-class distances should be zero and inter-class distances should be infinity.
- This corresponds to a kernel matrix with structure

$$K_{\text{ideal}} = \left[\begin{array}{c|c} 1 & -1 \\ \hline -1 & 1 \end{array}\right]$$

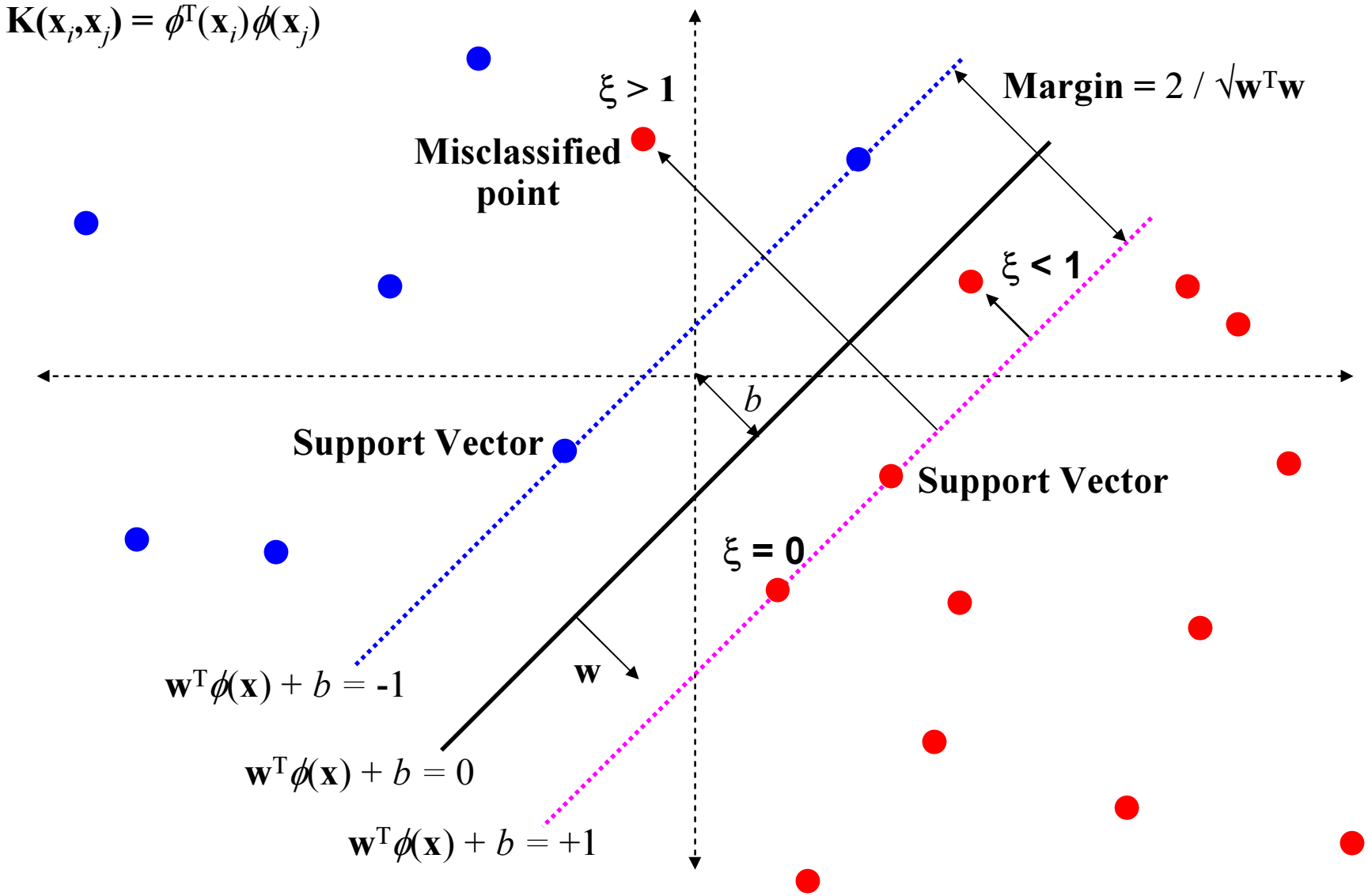
Proposed Solution

- Generate base descriptors by discretely sampling from the levels of invariance.
- Approximate the ideal case by forming a minimal combination of the base descriptor rather than selecting one of them.
- This is done in a framework geared specifically towards classification.
- To tackle the 4 vs 9 example
    - Combine only the rotation and scale invariant descriptors
    - The combined descriptor has neither invariance in full.
    - The distance between a digit and its rotated copy is therefore non-zero, but still remains tolerably small
    - Similarly, small scale changes will lead to small non-zero distances within class
    - However, distances will also be increased across classes and by a large enough margin to ensure good classification and generalisation.

Trading-Off Descriptors by Learning the Kernel

- We start with N_k base descriptors and associated N_k distance functions $f_1, f_2, \ldots, f_{Nk}$.
- Each base descriptor has a different level of invariance
  - Geometric: Rotation, scale, affine, none, etc.
  - Illumination: Affine, histogram equalisation, etc.
  - Cue: Shape, colour, texture.
- We kernelise each base descriptor to get a corresponding base kernel K_k.
- The optimal kernel is learnt as a linear combination of base kernels
  - $K_{opt}(x,y) = \Sigma\, d_k\, K_k(x,y)$
  - The weights d_k correspond to the trade-off between different levels of invariance.

The Standard SVM Formulation



The Standard SVM Formulation

- The primal formulation for l_1 C-SVM classification is
  - Minimise $\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\,\Sigma_i \xi_i$ with respect to $\mathbf{w}$, ξ, b.
  - Subject to
    - $y_i\,[\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i) + b] \geq 1 - \xi_i$
    - $\xi_i \geq 0$
  - .where
    - $(\mathbf{x}_i, y_i)$ is the i^{th} training point..
    - C is the misclassification penalty.
    - $\phi(\mathbf{x}_j)$ is an embedding such that $\mathrm{K}(\mathbf{x}_i,\mathbf{x}_j) = \phi^{\mathrm{T}}(\mathbf{x}_i)\phi(\mathbf{x}_j)$

Our Primal Formulation

- Our primal formulation is
  - Minimise $\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\,\Sigma_i \xi_i + \Sigma_k \sigma_k d_k$ with respect to $\mathbf{w}$, ξ, b, $\mathbf{d}$.
  - Subject to
    - $y_i\,[\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_i) + b] \geq 1 - \xi_i$
    - $\xi_i \geq 0$
    - $d_k \geq 0$ (for interpretability and computational efficiency)
    - $\mathbf{Ad} \geq \mathbf{p}$ (for incorporating any additional prior knowledge)
  - .where
    - $(\mathbf{x}_i, y_i)$ is the i^{th} training point..
    - C is the misclassification penalty.
    - $\mathrm{K}(\mathbf{x}_i,\mathbf{x}_j) = \phi^{\mathrm{T}}(\mathbf{x}_i)\phi(\mathbf{x}_j) = \Sigma_k\, d_k\, \phi_k{}^{\mathrm{T}}(\mathbf{x}_i)\phi_k(\mathbf{x}_j) = \Sigma_k\, d_k\, \mathrm{K}_k(\mathbf{x}_i,\mathbf{x}_j)$

Remarks About the Primal

- Just as in the standard SVM case, the primal variables can't be solved for explicitly.
  - ϕ is typically unspecified.
  - **w** might be very high dimensional
- We would like to discover a set of minimal invariances. This is done via l_1 regularisation of **d** so that most of the weights are learnt to be zero.
- σ encodes our prior knowledge about the relative importance of the different invariances.
- Similarly $\mathbf{Ad} \geq \mathbf{p}$ can be used to enforce prior constraints.

Our Dual Formulation

- The equivalent dual formulation becomes
  - Maximise $\mathbf{1}^T\alpha + \mathbf{p}^T\delta$ with respect to α, δ
  - Subject to
    - $0 \leq \alpha_i \leq C$
    - $\mathbf{1}^T\mathbf{Y}\alpha = 0$
    - $\frac{1}{2}\alpha^T\mathbf{Y}\mathbf{K}_k\mathbf{Y}\alpha \leq \sigma_k - \delta^T\mathbf{A}_k$
  - where
    - α are the Lagrange multipliers corresponding to the support vectors
    - $\mathbf{Y}$ is a diagonal matrix such that $Y_{ii} = y_i$
    - $\mathbf{A}_k$ is the k^{th} column of $\mathbf{A}$,

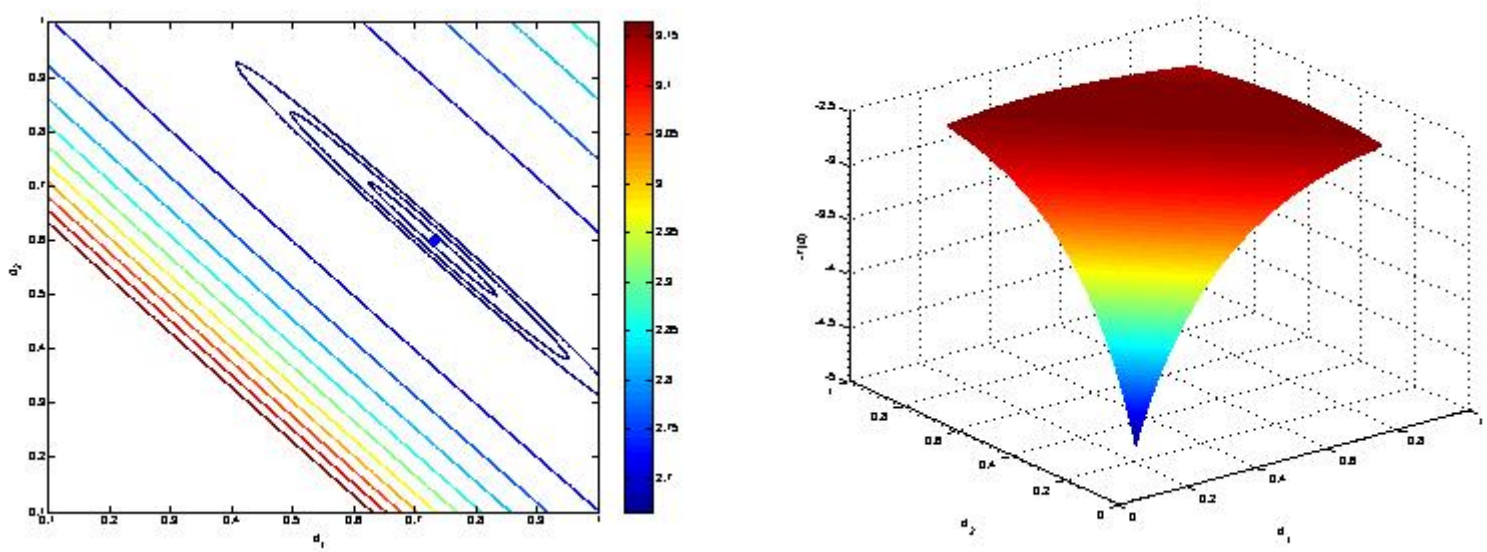
Our Dual Formulation

- The equivalent dual formulation becomes
  - Maximise $\mathbf{1}^T\alpha + \mathbf{p}^T\delta$ with respect to α, δ
  - Subject to
    - $0 \leq \alpha_i \leq C$
    - $\mathbf{1}^T\mathbf{Y}\alpha = 0$
    - $\frac{1}{2}\alpha\mathbf{Y}\mathbf{K}_k\mathbf{Y}\alpha \leq \sigma_k - \delta^T\mathbf{A}_k$
- The dual is convex with a unique global optimum. It falls into the category of Second Order Cone Programs (SOCP).
- SOCPs can be solved relatively efficiently using off-the-shelf numerical optimisation packages such as SeDuMi.

Large Scale Reformulation

- We reformulate the optimisation so as to tackle large scale problems involving hundreds of kernels.
- We adopt an iterative two stage approach [Chapelle *et al.* ML 02, Rakotomamonjy *et al.* ICML 07]. In stage 1, we optimise over the kernel weights while in stage 2 we fix the weights and optimise over the SVM parameters.
- The primal is reformulated as
  - Minimise $T(\mathbf{d})$ subject to $d_k \geq 0$, $\mathbf{Ad} \geq \mathbf{p}$
  - where $T(\mathbf{d}) = \text{Min } \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C \Sigma_i \xi_i + \Sigma_k \sigma_k d_k$ w.r.t. $\mathbf{w}$, ξ, b.
  - Subject to
    - $y_i [\mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}_i) + b] \geq 1 - \xi_i$
    - $\xi_i \geq 0$

Large Scale Reformulation



Contour plot of $T(\mathbf{d})$ and surface plot of $-T(\mathbf{d})$ for Gerenuk vs Panda from the Caltech 101 database

- The strategy is to minimise T by gradient descent via the iteration

$$d_k^{n+1} = d_k^n - \varepsilon^n \, \partial T / \partial d_k$$

- The step size ε^n is chosen as a variant of the Armijo rule to guarantee convergence.

Large Scale Reformulation

- To compute $\partial T / \partial d_k$ we turn to the dual of T
  - $W(\mathbf{d}) = \text{Max } \mathbf{1}^{\mathrm{T}}\alpha + \boldsymbol{\sigma}^{\mathbf{T}}\mathbf{d} - \frac{1}{2} \Sigma_k d_k \alpha^{\mathbf{T}}\mathbf{Y}\mathbf{K}_{\mathrm{k}}\mathbf{Y}\alpha$ with respect to α
  - Subject to $0 \leq \alpha_i \leq C$, $\mathbf{1}^{\mathrm{T}}\mathbf{Y}\alpha = 0$
- By the principle of strong duality $T(\mathbf{d}) = W(\mathbf{d})$.
- Let α^* maximise W. Then, W is differentiable with respect to $\mathbf{d}$ if α^* is unique [Bonnans and Shapiro 2000].
- If all the kernel matrices are positive definite then W is strictly concave and α^* is therefore unique.
- Furthermore, $\partial W / \partial d$ is independent of α^* [Chapelle *et al*. ML 02].
- Thus, $\partial T / \partial d_k = \partial W / \partial d_k = \sigma_k - \frac{1}{2}\alpha^{*\mathrm{T}}\mathbf{Y}\mathbf{K}_{\mathrm{k}}\mathbf{Y}\alpha^*$
- Since $\mathbf{d}$ is fixed, W is the standard SVM dual and any large scale SVM solver of choice can be used to solve for α^*.

Final Algorithm

1. Initialise $\mathbf{d}^0$ randomly
2. Repeat until convergence
   a) Form $K(x,y) = \Sigma_k \, d^n_k \, K_k(x,y)$
   b) Use a large scale SVM solver of choice to solve the standard SVM problem with kernel K and obtain α^*.
   c) Update $d_k^{n+1} = d_k^n - \varepsilon^n(\sigma_k - \frac{1}{2}\alpha^{*T}\mathbf{Y}\mathbf{K}_k\mathbf{Y}\alpha^*)$
   d) Project $\mathbf{d}^{n+1}$ back onto the feasible set if it does not satisfy the constraints $\mathbf{d}^{n+1} \geq 0$ and $\mathbf{A}\mathbf{d}^{n+1} \geq \mathbf{p}$

Multi-Class Classification

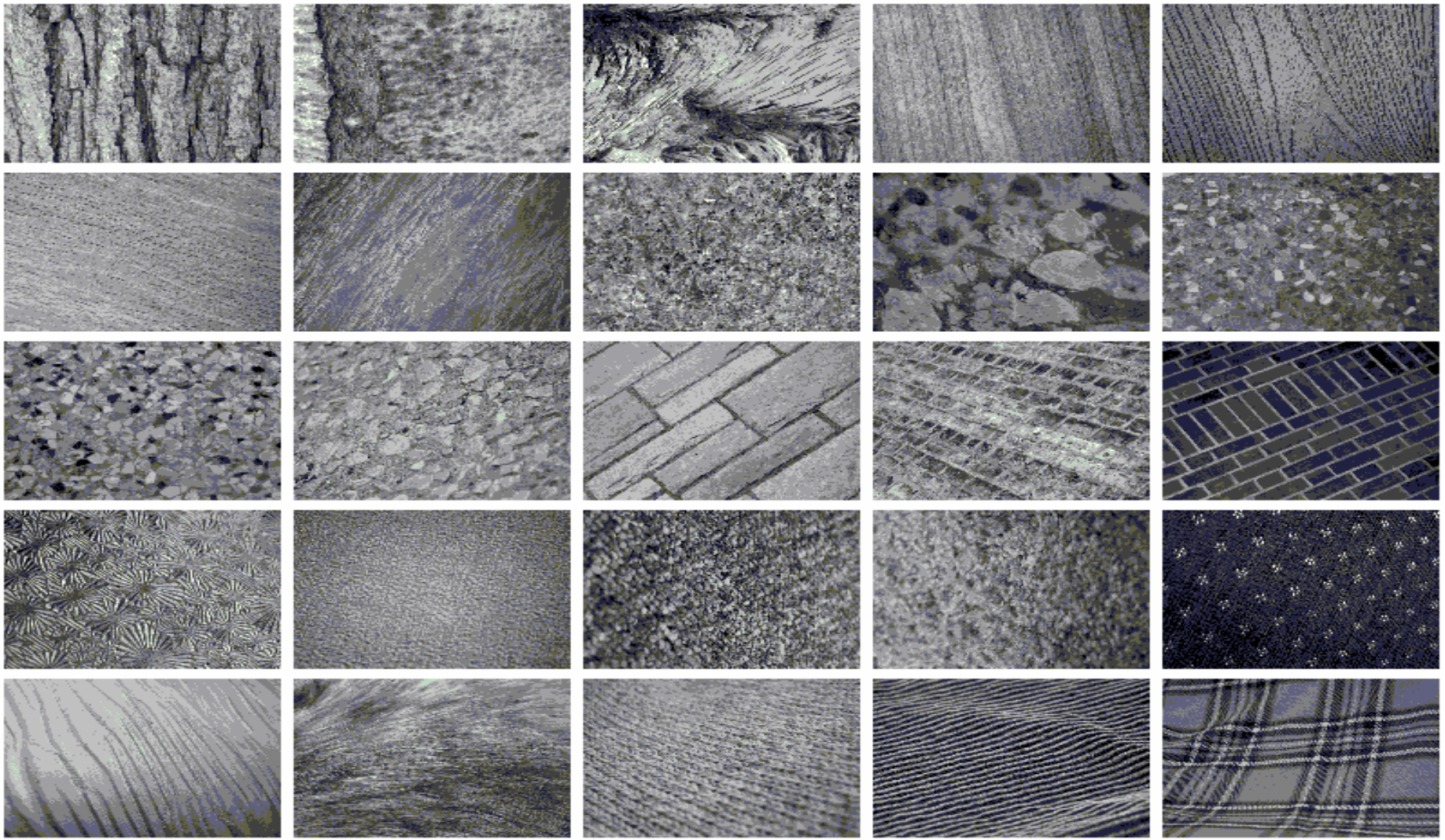
- The kernel learning approach is equally applicable to 1-vs-1 or 1-vs-All classification and we try both methods.
- We perform 1-vs-1 classification via majority vote. Thus for an N_c class problem
  - We learn $N_c * (N_c - 1) / 2$ binary classifiers covering every class pair.
  - A novel image is classified by taking the majority vote over classes of all the learnt classifiers.
- A DAGSVM would have identical training but the classification would be much faster (but the ordering of the DAG is important).
- We perform 1-vs-All classification according to the maximum distance from the margin.

Prior Knowledge and Hyper Parameters

- We would like to test how general our formulation is so parameters are fixed for all experiments (no fine tuning is done).
- No prior knowledge is assumed
  - σ_k = constant throughout (no descriptor is favoured *a priori*).
  - The constraints **Ad ≥ p** are not used (unless specified).
- The χ^2 kernel is used throughout
  - $K_k(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\gamma_k \chi 2(\mathbf{x}_i,\mathbf{x}_j))$
  - γ_k = 1 / mean of all χ^2 distances in current training set.
- The misclassification penalty is fixed to C = 1000 throughout
- Potentially both C and γ_k can be learnt in the proposed framework.

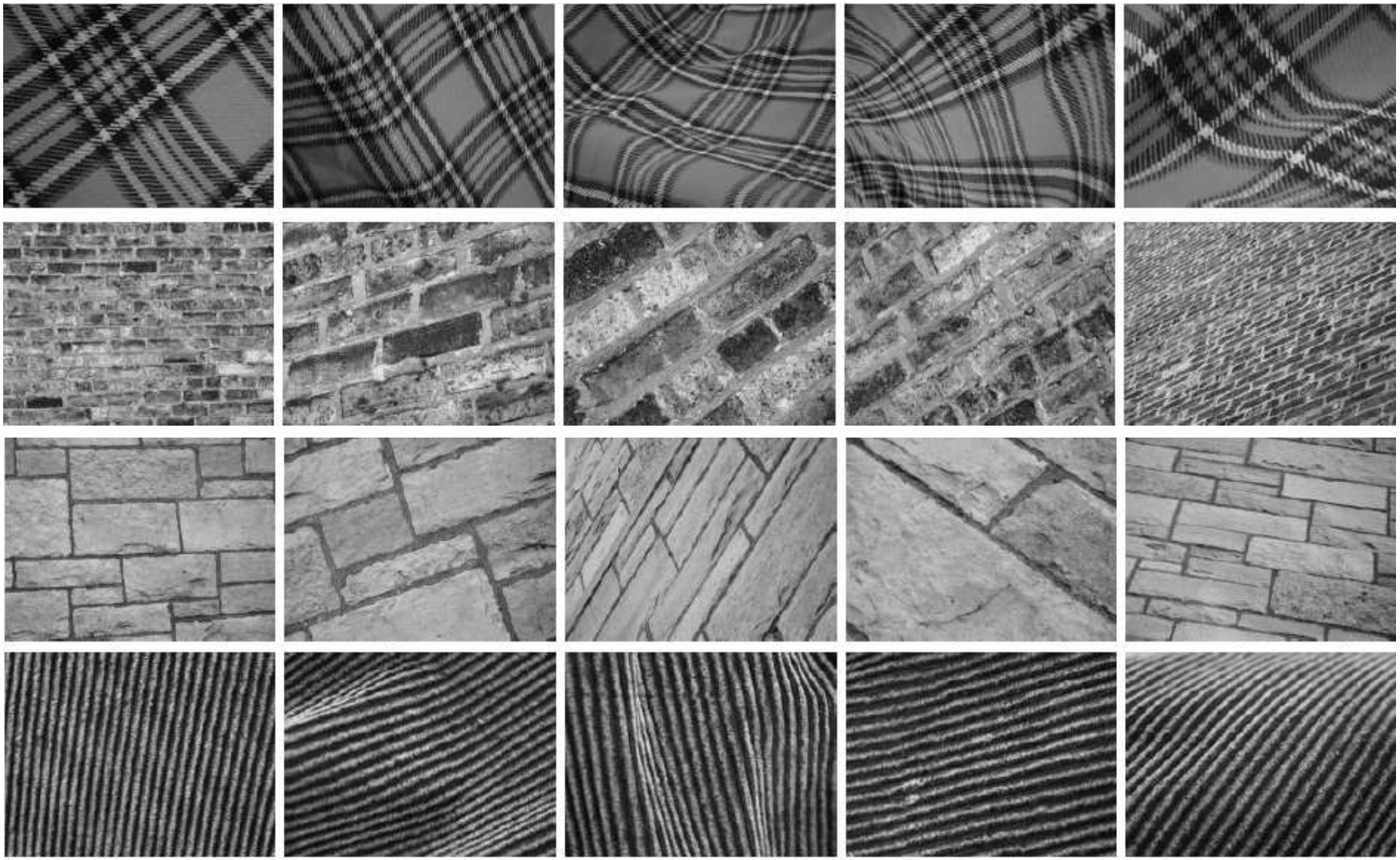
The UIUC Texture Database

- Texture database collected by Lazebnik *et al*. [CVPR 2003, PAMI 2005].



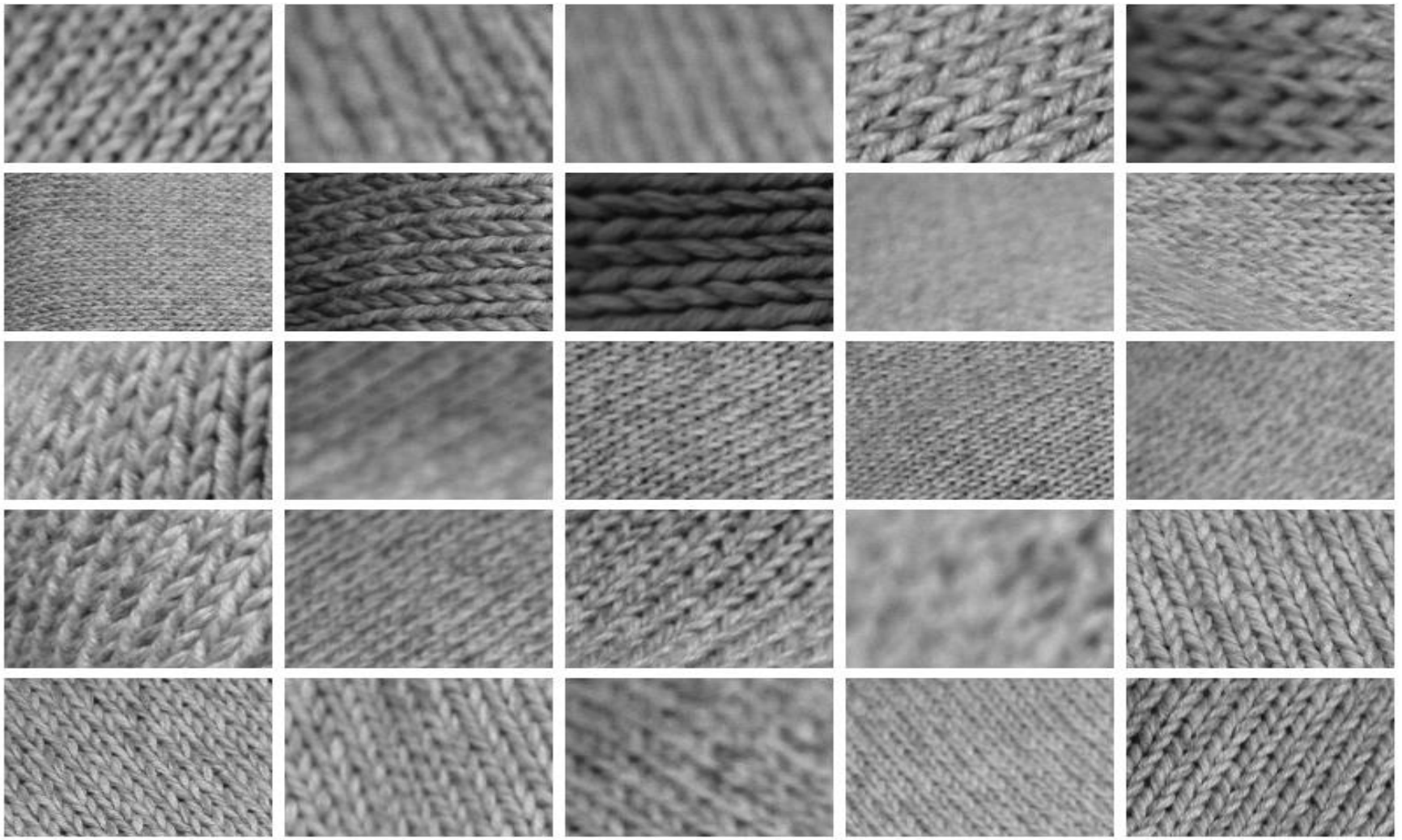
The UIUC Texture Database

- The images vary a lot due to viewpoint changes and non-rigid deformations.



The UIUC Texture Database

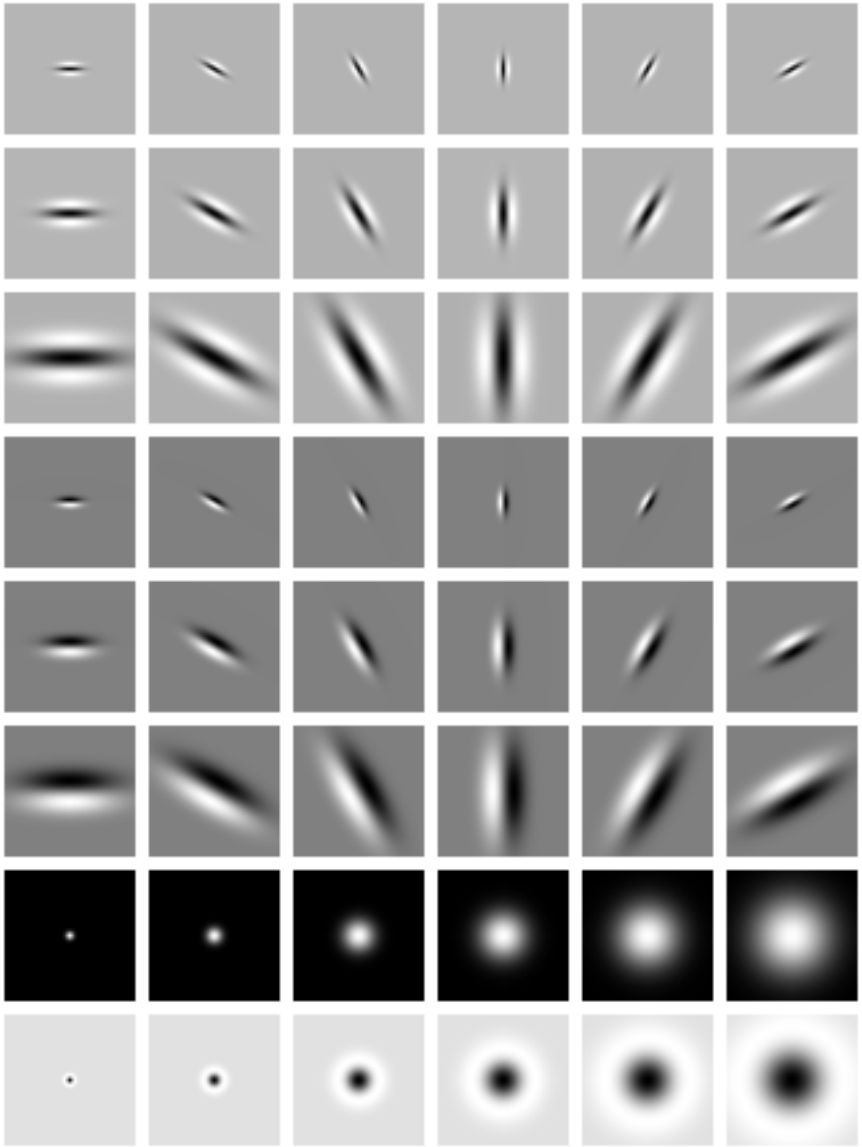
- The images vary a lot due to viewpoint changes and non-rigid deformations.



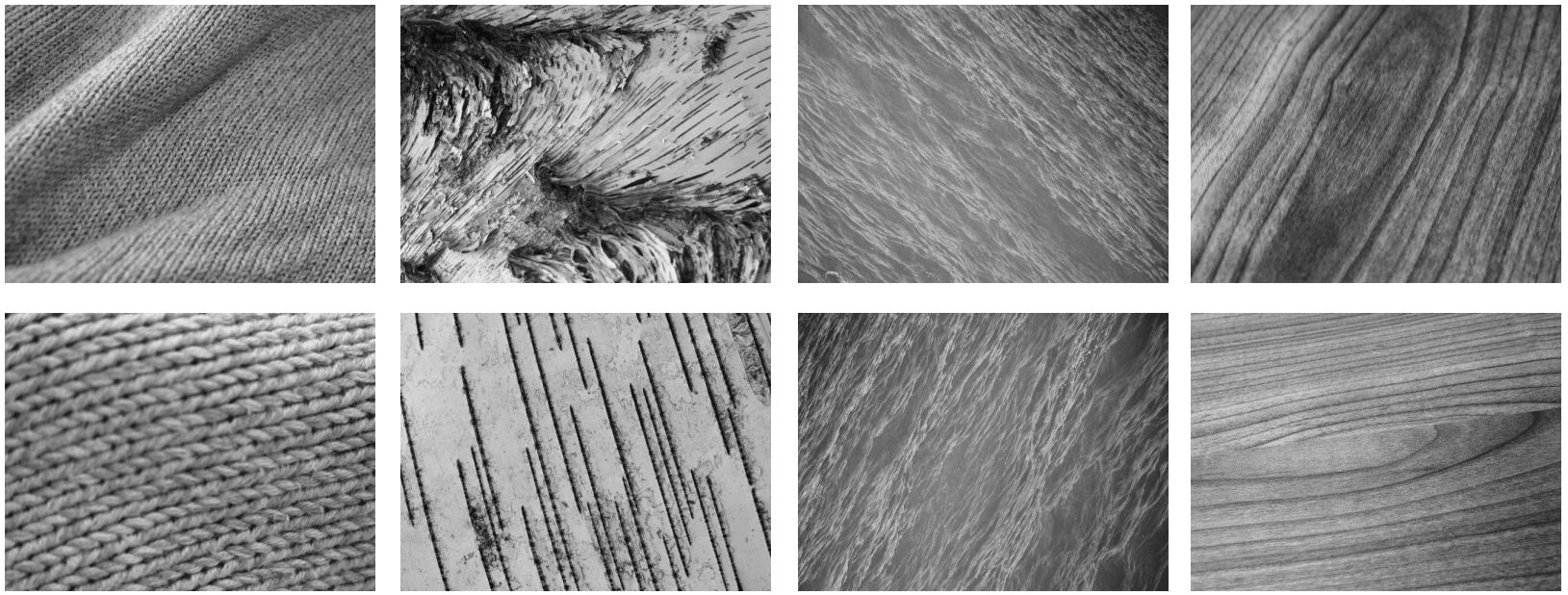
Experimental Setup – UIUC Textures

- Experimental setup kept identical to that of Zhang *et al*. [IJCV 2006]
  - 25 classes, 40 images / class = 1000 images in all.
  - 20 images / class used for training and the other 20 for testing.
  - Results reported over 20 random splits.
- We try multiple texture features with different levels on invariance
  - Patch: No invariance
  - MR filters: Invariant to patches in the null space of the filter bank
  - Rotationally invariant patches, fractal intercepts and maximum responses over oriented filters result in different types of rotationally invariant features
  - Maximum response can be taken over just scale to give scale invariant features.
  - Maximum response over both orientation and scale leads to rotation and scale invariant features (but not affine invariant).
  - Fractal slopes lead to invariance to biLipschitz transforms.

The Maximum Response Filters (ECCV 2002)



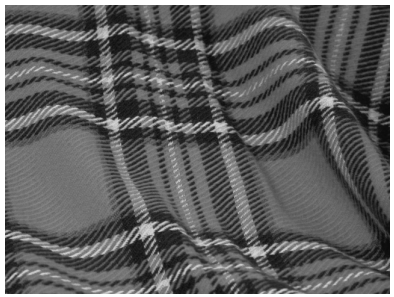
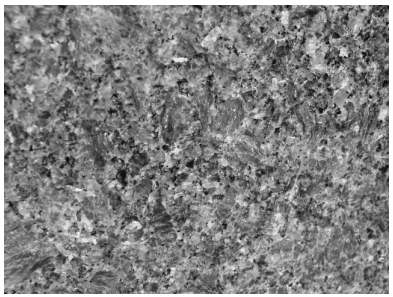
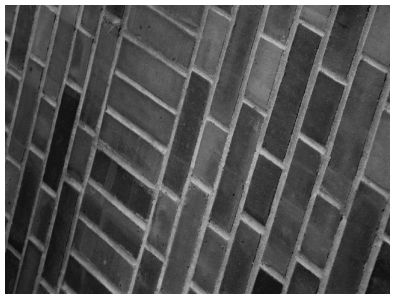
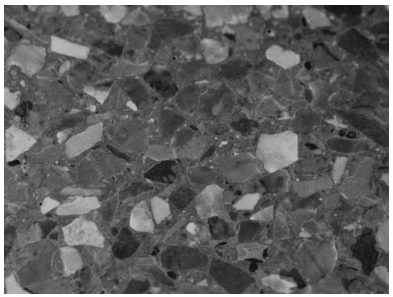
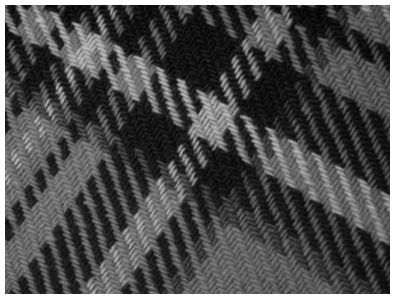
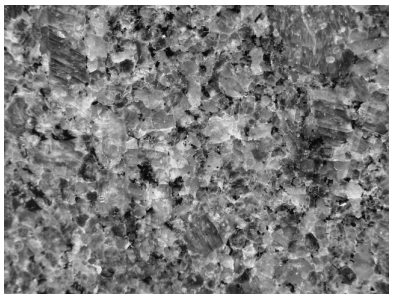
Experimental Results – UIUC Textures



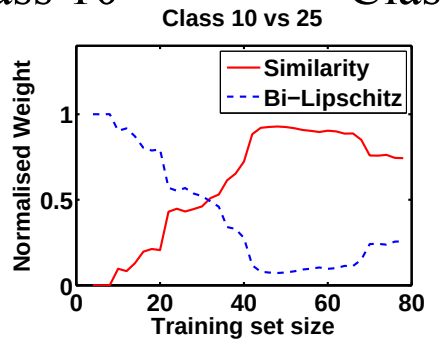
Class 23 Class 3 Class 7 Class 4

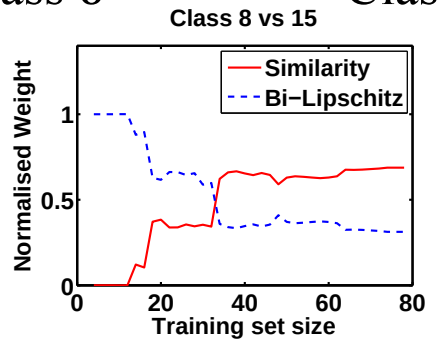
- Class 23 vs Class 3: biLipschitz = 1.97, Rotation = 0.00, all the rest are zero..
- Class 23 vs Class 7: biLipschitz = 0.23, Rotation = 1.46, all the rest are zero..
- Class 4 vs Class 7: biLipschitz = 0.00, Rotation = 1.79, all the rest are zero.

Experimental Results – UIUC Textures


Class 10 Class 25 Class 8 Class 15





• Higher levels of invariance are automatically learnt for smaller training set sizes while lower levels of invariance are learnt for larger set sizes.

Experimental Results – UIUC Textures

	1-NN	SVM (1-vs-1)	SVM (1-vs-All)
None (patch)	82.39 ± 1.58	91.46 ± 1.13	92.87 ± 1.40
None (MR)	82.18 ± 1.51	91.16 ± 1.05	91.87 ± 1.50
Rotation (patch)	97.83 ± 0.63	98.18 ± 0.43	98.53 ± 0.12
Rotation (MR)	93.00 ± 1.04	96.69 ± 0.74	97.07 ± 0.83
Rotation (Fractals)	95.05 ± 0.93	97.24 ± 0.76	97.60 ± 0.92
Scale	76.77 ± 1.77	87.04 ± 1.57	88.73 ± 1.03
Rotation + Scale	90.35 ± 1.15	95.12 ± 0.95	96.00 ± 1.00
biLipschitz	95.35 ± 0.88	97.19 ± 0.52	97.73 ± 0.12
MKL Block l_1		96.94 ± 0.91	97.67 ± 0.50
Our		**98.76 ± 0.65**	**98.90 ± 0.68**

• Our results compares to the state of the art 98.70% ± 0.4 by Zhang *et al*. [IJCV 2006] using an equally weighted combination of SIFT, SPIN and RIFT descriptors.

• However, if the method of Zhang *et al*. is applied on the given features, our method is always superior with a mean of 2.00 ± 0.76%.

The Oxford Flower Database

• Flower database collected by Nilsback and Zisserman [CVPR 2006] includes images from various websites and personal contributions.

The Oxford Flower Database

• The database contains flowers with many colours, shapes and textures but no single feature is sufficient for classification.

Need colour invariance but shape and texture can be used to discriminate between classes

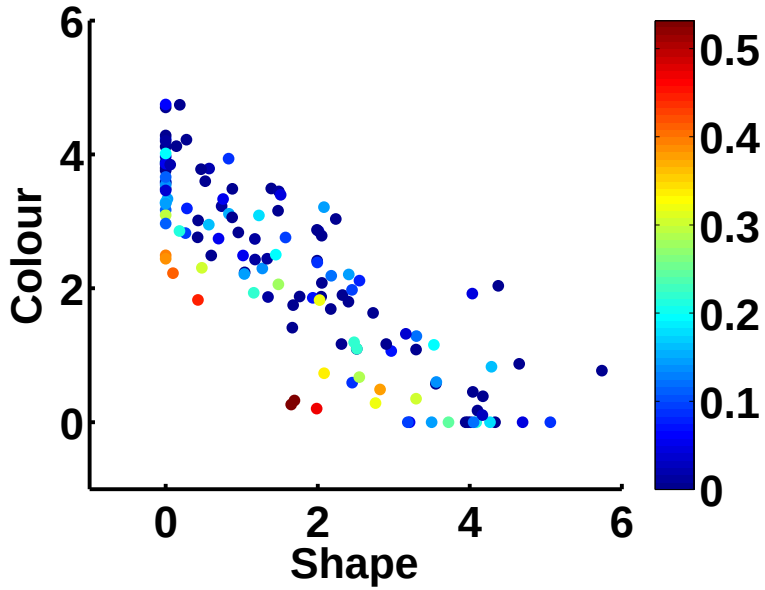
Need shape invariance but colour and texture can be used to discriminate between classes

Experimental Setup – Oxford Flowers

- Experimental setup kept identical to that of Nilsback and Zisserman [CVPR 2006]
  - 17 classes, 80 images / class = 1360 images in all.
  - From each class, 40 images are used for training, 20 for validation and 20 for testing (we do not use the validation set).
  - Results reported over 20 random splits including 3 splits from the original paper.
- We start with χ^2 distance matrices between features provided directly by the authors
  - Shape: Histograms of densely sampled SIFT features
  - Colour: Histograms of HSV features
  - Texture: Histograms of MR8 filter responses
- Cue combination can be seen as a special case in our formulation
  - HSV features might have high colour discriminative power but might be invariant to texture and shape changes in the object.
  - .Similarly for colour and shape descriptors

Experimental Results – Oxford Flowers

	1-NN	SVM (1-vs-1)	SVM (1-vs-All)
Shape	53.30 ± 2.69	68.88 ± 2.04	70.20 ± 1.33
Colour	47.32 ± 2.59	59.71 ± 1.95	60.00 ± 1.93
Texture	39.36 ± 2.43	59.00 ± 2.14	62.94 ± 2.30
MKL Block l_1		77.84 ± 2.14	81.47 ± 1.06
Our		**80.49 ± 1.97**	**82.55 ± 0.34**



• Naïve brute force 1-vs-1 search for the optimum weights leads to severe over fitting. We therefore fix the weights for all the classes when performing brute force search and get 80.62 ± 1.65%. Our method achieves comparable results without using a validation set.

• Brute force 1-vs-All search is too computationally expensive to be viable.

• Colour dominates most frequently (60.29%), shape (38.24%) and texture (1.47%).

• Forcing texture weights to be higher than colour yields 81.12 ± 2.09%.

• Our results are better by over 11% as compared to using any single descriptor.

Experimental Results – Oxford Flowers










Dandelions Wild Tulips Crocuses Cowslips Irises

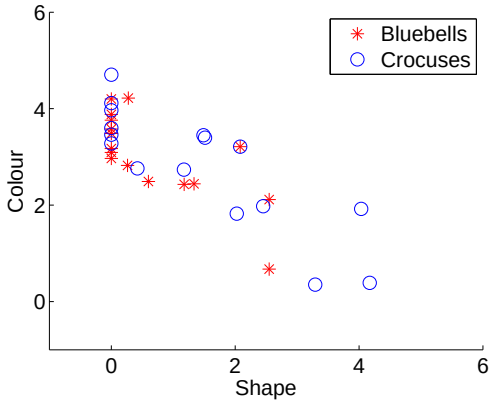
- Dandelions vs Wild Tulips: Shape = 3.94, Colour = 0.00, Texture = 0.00.
- Dandelions vs Crocuses: Shape = 0.42, Colour = 2.46, Texture = 0.00.
- Cowslips vs Irises: Shape = 1.48, Colour = 2.00, Texture = 1.36

Experimental Results – Oxford Flowers

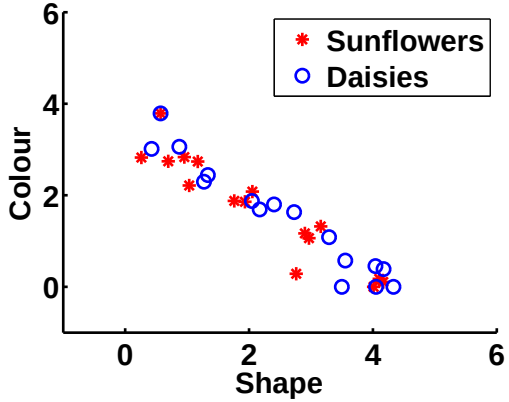
Bluebells

Sunflowers

Crocuses



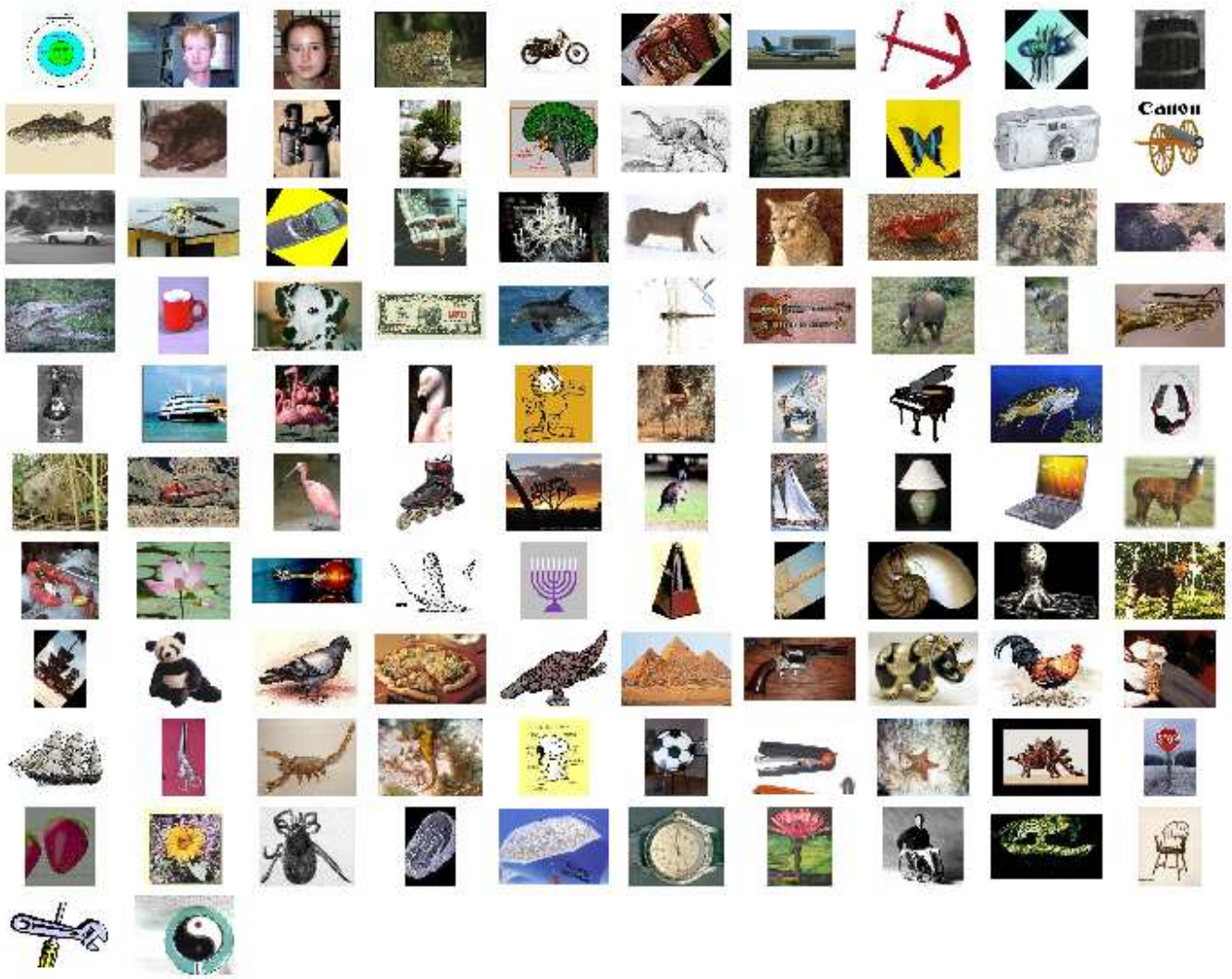
Daisies



• By and large, Bluebells share the same invariances as Crocuses and this information can be used for transfer learning. The same is true for Sunflowers and Daisies.

The Caltech 101 Database

- Object category database collected by Fei-Fei *et al*. [PAMI 2006].



Experimental Setup – Caltech 101

- Experimental setup kept identical to that of Zhang *et al*. [CVPR 2006]
  - 102 classes, 30 images / class = 3060 images in all.
  - 15 images / class used for training and the other 15 for testing.
  - Results reported over 20 random splits (including the 2 used by Zhang *et al*.).
- We learn the trade-off between shape and appearance cues:
  - AppGray: Spatial pyramid kernel over histograms of SIFT codebook (SIFT descriptors are densely sampled at multiple scales)..
  - AppColour: Same as AppGray but computed over HSV colour channels.
  - Shape 180: Spatial pyramid kernel of histogram of oriented gradients.
  - Shape 360:. Gradients are taken in the range [0, 360] rather than [0, 180].
  - Shape via Geometric Blur1: Symmetrised average minimum distance between Geometric Blur features in the two images.
  - Shape via Geometric Blur 2: Incorporates additional geometric distortion term.

Experimental Results – Caltech 101

	1-NN	SVM (1-vs-1)	SVM (1-vs-All)
Shape GB1	39.67 ± 1.02	57.33 ± 0.94	62.98 ± 0.70
Shape GB2	45.23 ± 0.96	59.30 ± 1.00	61.53 ± 0.57
Self Similarity	40.09 ± 0.98	55.10 ± 1.05	60.83 ± 0.84
Shape 180	32.01 ± 0.89	48.83 ± 0.78	49.93 ± 0.52
Shape 360	31.17 ± 0.98	50.63 ± 0.88	52.44 ± 0.85
App Colour	32.79 ± 0.92	40.84 ± 0.78	43.44 ± 1.46
App Gray	42.08 ± 0.81	52.83 ± 1.00	57.00 ± 0.30
MKL Block l_1		77.72 ± 0.94	83.78 ± 0.39
Our		**81.54 ± 1.08**	**89.56 ± 0.59**

Comparisons to the state-of-the-art

- Zhang *et al.* [CVPR 06]: 59.08 ± 0.38% by combining shape and texture cues. Frome *et al.* add colour and get 60.3 ± 0.70% [NIPS 2006] and 63.2% [ICCV 2007].
- Lin *et al.* [CVPR 07]: 59.80% by combining 8 features using Kernel Target Alignment
- Bosch *et al.* [CIVR 07]: 71.4 ± 0.8% via brute force search over 4 kernels (excluding background). For the same kernels we get 79.85 ± 0.04% (including the background class).

Experimental Results – Caltech 101

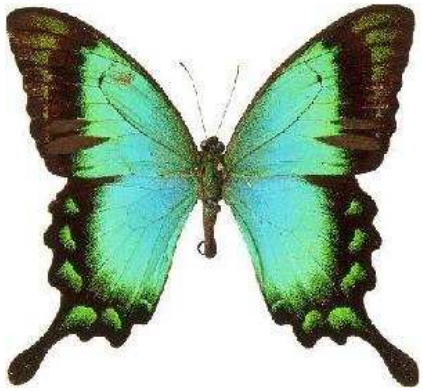
Pizza

Soccer Ball

Watch

Car Side

Octopus



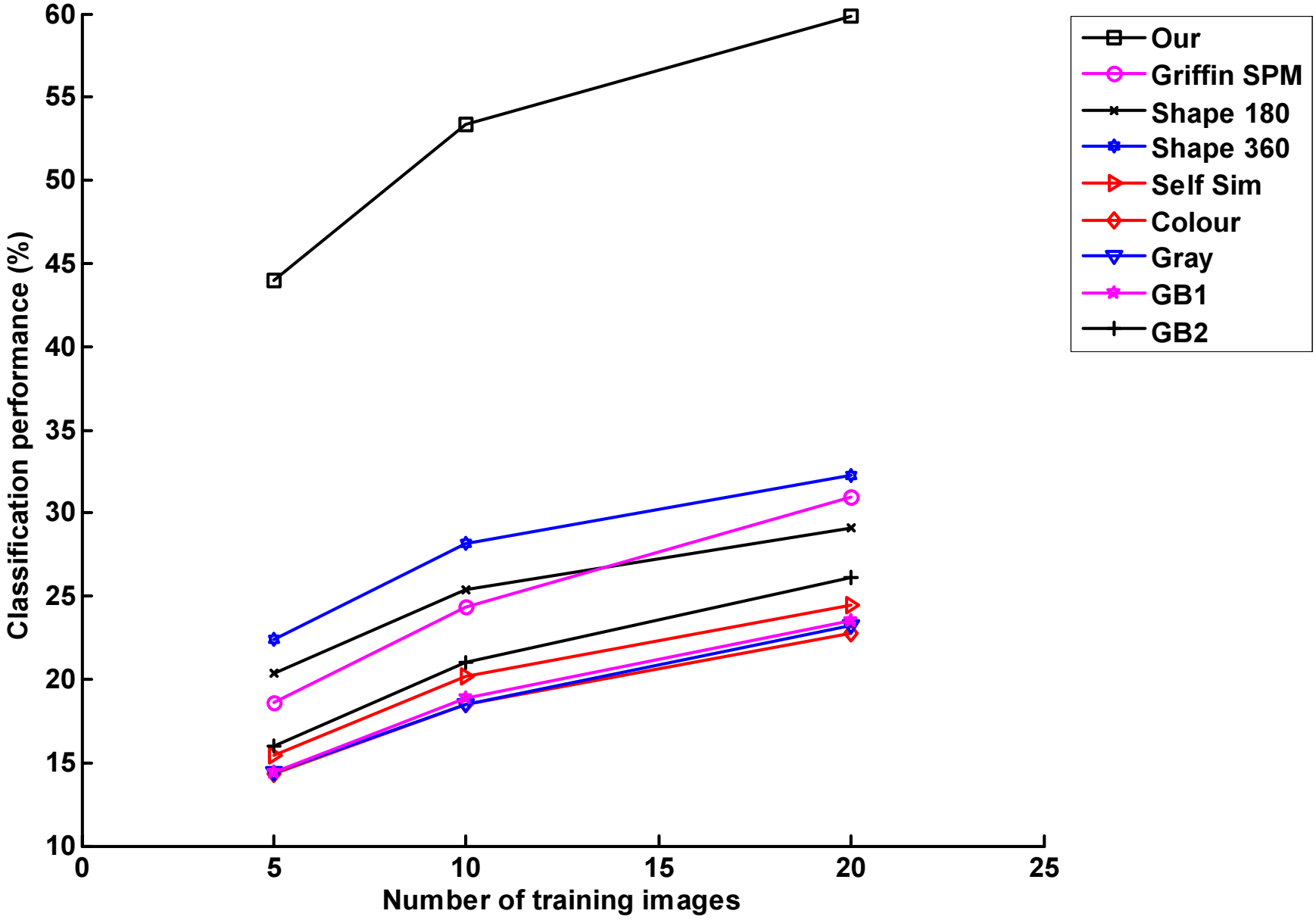
Butterfly

Guitar

No shape information is used for these classification tasks

No appearance information is used for these classification tasks

Experimental Results – Caltech 256



Experimental Results – Caltech 256

No. of Kernels	Kernels	5 Training	10 Training	20 Training
4	**AppGray + AppColour + Shape180 + Shape360**	41.28	49.08	56.92
5	**4 + Self Similarity**	43.39	52.42	59.98
6	**5 + Shape GB1**	44.31	52.34	60.55
7	**6 + Shape GB2**	43.63	53.09	60.31

Change in classification performance as the number of kernels is varied.

Conclusions

- The proposed framework appears to be generally applicable.
- No hand tuning of parameters was required.
- It can be used to combine heterogeneous information. This is particularly relevant when human intuition of invariances might not be very good – such as when combining audio, video and text.
- It can work with poor/highly specialised descriptors. Most of the times the weight will be set to 0 unless the descriptor can actually improve classification performance. This is not the case for equally weighted descriptors where the overall classification performance might fall when poor descriptors are included.
- Our results appear to be similar to a brute force search on a validation set.
    - Brute force search becomes impractical when more than 4 kernels are present
    - It is possible to over fit to the validation set.
    - Uses precious training data (not feasible when training on 5 images/class)

Acknowledgements

- For providing base kernel matrices and helpful discussions
  - P. Anandan
  - Anna Bosch
  - Rahul Garg
  - Varun Gulshan
  - Jagadeesh Kudur
  - Jitendra Malik
  - Maria Elena Nilsback
  - Patrice Simard
  - Kentaro Toyama
  - Hao Zhang
  - Andrew Zisserman