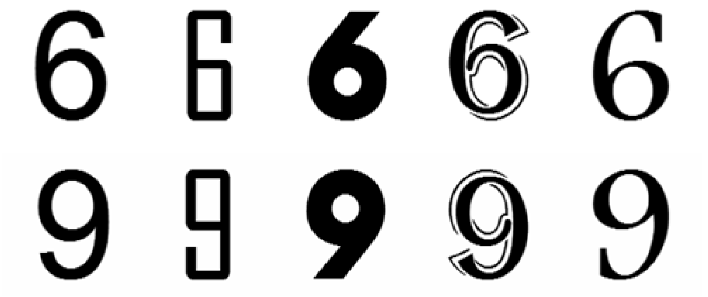
Learning the Discriminative Power-Invariance Trade-Off

Manik Varma
Microsoft Research India
manik@microsoft.com

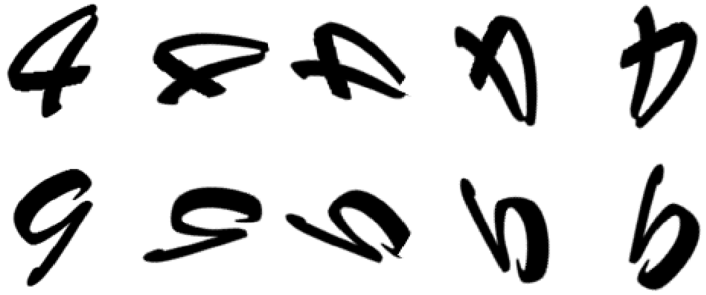
Debajyoti Ray
Gatsby Computational Neuroscience Unit
University College London
debray@gatsby.ucl.ac.uk

Slides adapted from Research.Microsoft Poster

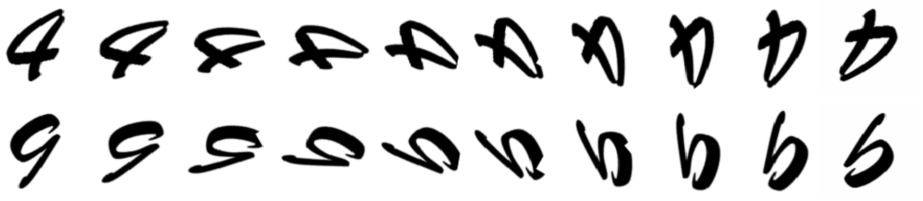
Task Specific Trade-Off



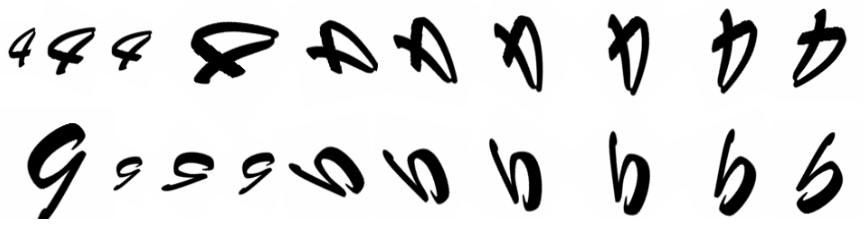
Don’t want rotation invariance



Do want rotation invariance



Don’t want size invariance

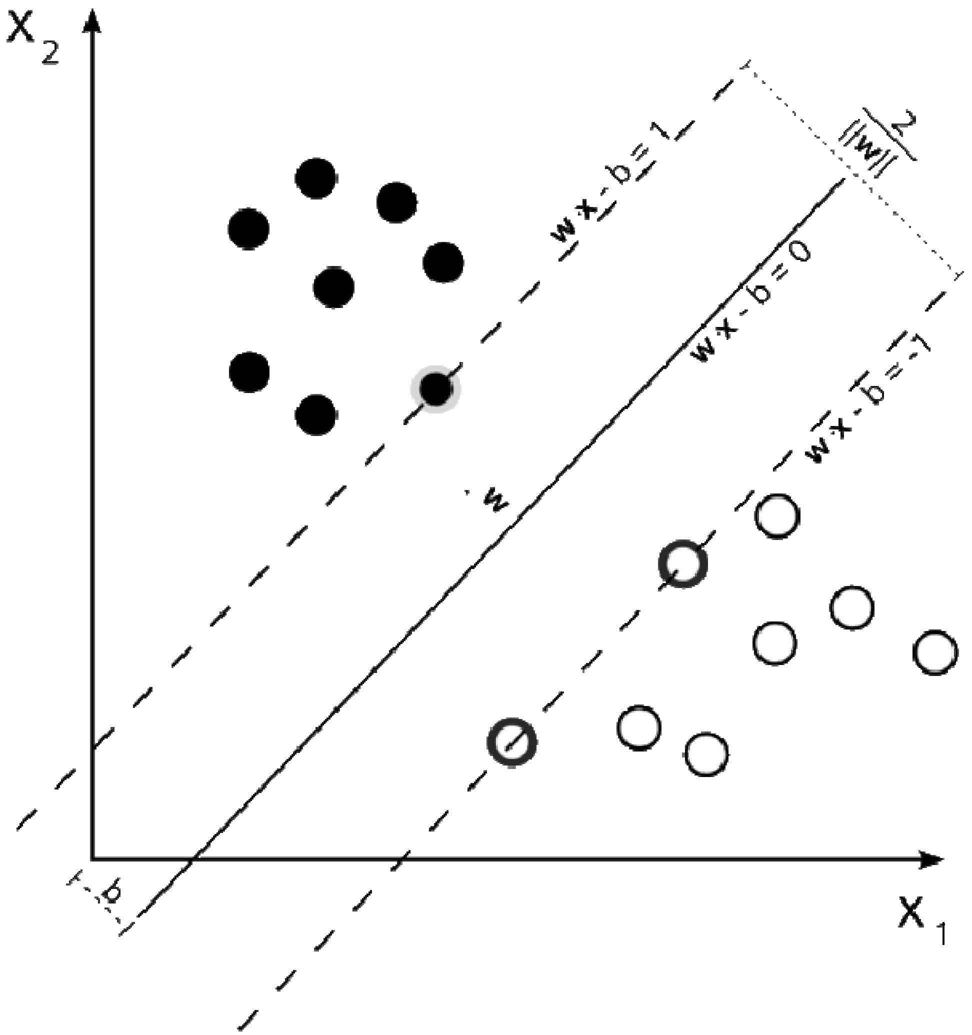


What is the right amount of invariance?

Solution: **Sparse Multiple Kernel Learning Classification Formulation:** We implement our proposed solution by learning the optimal domain specific kernel as a linear combination of base kernels, i.e.

$$K_{opt} = \sum d_k K_k$$

SVM and Kernels- a quick review (6 slides)



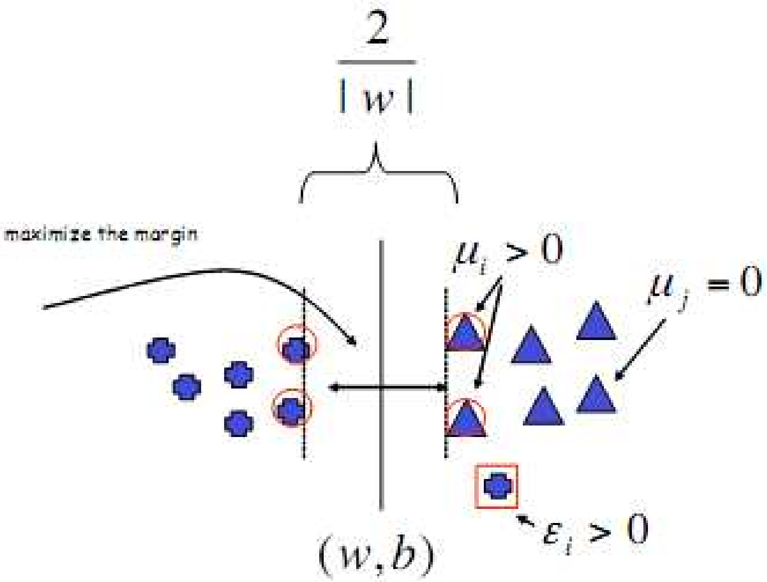
The QLP problem:

$$\min_{\mathrm{w,b},\varepsilon_i} \frac{1}{2} w \square w + v \sum_i \varepsilon_i$$

subject to

$$\mathrm{y}_i \left(\mathrm{w} \square \mathrm{x_i} \text{-b} \right) \geq 1\text{-}\varepsilon_i$$

$$\varepsilon_i \geq 0$$



The Lagrangian takes the following form:

$$L(\mathbf{w}, b, \epsilon_i, \mu) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + \nu \sum_{i=1}^{m} \epsilon_i - \sum_{i=1}^{m} \mu_i \left[y_i(\mathbf{w} \cdot \mathbf{x}_i - b) - 1 + \epsilon_i \right] - \sum_{i=1}^{m} \delta_i \epsilon_i$$

Where the criteria function is:

$$\theta(\mu) = \min_{\mathbf{w}, b, \epsilon} L(\mathbf{w}, b, \boldsymbol{\epsilon}, \boldsymbol{\mu}, \boldsymbol{\delta}).$$

Since the minimum is obtained at the vanishing partial derivatives:

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{w}} &= \mathbf{w} - \sum_i \mu_i y_i \mathbf{x}_i = 0 \\
\frac{\partial L}{\partial b} &= \sum_i \mu_i y_i = 0 \\
\frac{\partial L}{\partial \epsilon_i} &= \nu - \mu_i - \delta_i = 0
\end{aligned}$$

Substituting these results/constraints back into the Lagrangian we obtain the dual problem:

$$\max_{\mu_1,\ldots,\mu_m} \qquad \theta(\boldsymbol{\mu}) = \sum_{i=1}^{m} \mu_i - \frac{1}{2}\sum_{i,j} \mu_i\mu_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

$$subject\ \ to$$

$$0 \le \mu_i \le \nu \qquad i = 1, \ldots, m$$

$$\sum_{i=1}^{m} y_i \mu_i = 0$$

In compact form, define $M_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$

$$\theta(\boldsymbol{\mu}) = \boldsymbol{\mu}^\top \mathbf{1} - \tfrac{1}{2}\boldsymbol{\mu}^\top M \boldsymbol{\mu}$$

The Kernel Trick

- note that the dual problem doesn’t explicitly use x or any function other then the inner products.
- Instead use

$$k\left(x_i, x_j\right) = \phi\left(x_i\right) \cdot \phi\left(x_j\right)$$

were $\phi\left(x\right): R^n \rightarrow F$ where F is an inner-product space.

Classifying New Instances

- Solving The QLP of the dual form will yield the solution for the Lagrange multipliers µ1, ..., µm.
- we can express φ(w) as a function of the (mapped) examples:

$$\phi(w) = \sum \mu_i y_i \phi(x_i)$$

To classify a new point x:

$$\begin{aligned} f(\mathbf{x}) &= sign(\phi(\mathbf{w})^\top \phi(\mathbf{x}) - b) = sign(\sum_i \mu_i y_i \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}) - b) \\ &= sign(\sum_i \mu_i y_i k(\mathbf{x}_i, \mathbf{x}) - b). \end{aligned}$$

Back to our problem:
Learning the Discriminative Power-Invariance Trade-Off

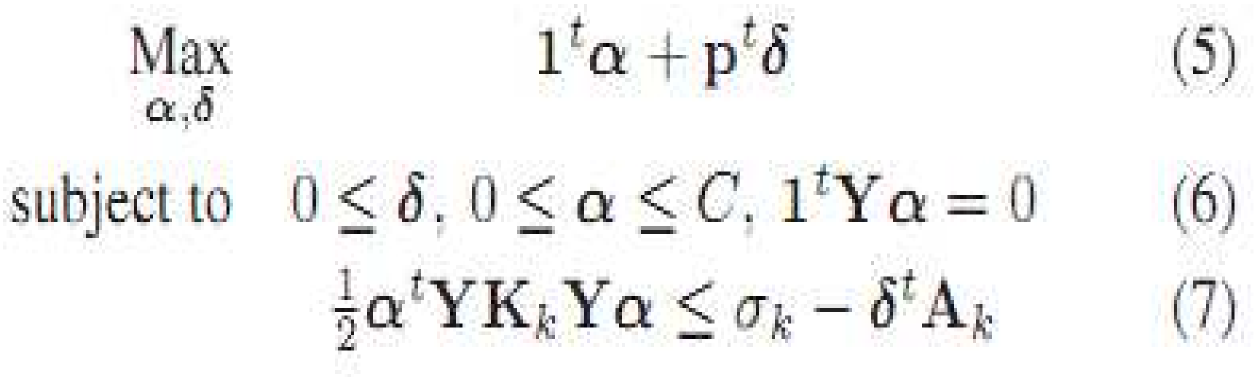
- Solution: “learning the optimal domain specific kernel as a linear combination of base kernels.
- “Kernalize” the base descriptors (many ways)

$$\mathbf{K}_{\text{opt}} = \sum_k d_k \mathbf{K}_k$$

$$\mathbf{K}_k(\mathbf{x},\mathbf{y}) = \exp(-\gamma_k f_k(\mathbf{x},\mathbf{y}))$$

$$\begin{array}{rcr}
\underset{\mathbf{w},\mathbf{d},\boldsymbol{\xi}}{\text{Min}} & \frac{1}{2}\mathbf{w}^t\mathbf{w} + C\mathbf{1}^t\boldsymbol{\xi} + \boldsymbol{\sigma}^t\mathbf{d} & (1) \\
\text{subject to} & y_i(\mathbf{w}^t\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i & (2) \\
 & \boldsymbol{\xi} \geq 0, \mathbf{d} \geq 0, \mathbf{A}\mathbf{d} \geq \mathbf{p} & (3) \\
\text{where} & \phi^t(\mathbf{x}_i)\phi(\mathbf{x}_j) = \sum_k d_k \phi_k^t(\mathbf{x}_i)\phi_k(\mathbf{x}_j) & (4)
\end{array}$$

The dual form



$$\max_{\alpha,\delta} \quad \mathbf{1}^t\boldsymbol{\alpha} + \mathbf{p}^t\boldsymbol{\delta} \tag{5}$$

$$\text{subject to} \quad 0 \le \boldsymbol{\delta},\ 0 \le \boldsymbol{\alpha} \le C,\ \mathbf{1}^t\mathbf{Y}\boldsymbol{\alpha} = 0 \tag{6}$$

$$\tfrac{1}{2}\boldsymbol{\alpha}^t\mathbf{Y}\mathbf{K}_k\mathbf{Y}\boldsymbol{\alpha} \le \sigma_k - \boldsymbol{\delta}^t\mathbf{A}_k \tag{7}$$

The dual is convex with a unique global optimum. It’s a standard SOCP problem and can be solved relatively efficiently using off the shelf packages such as SeDuMi.

- **Large Scale Reformulation:** We reformulate the primal so that we can use standard SVM solvers to tackle large scale problems involving hundreds of kernels.
- **Reformulation:** Minimise *T*(**d**) subject to *d* ≥ 0, **Ad ≥ p**
- **Where:**

$$\begin{aligned} T(\mathrm{d}) = \quad & \mathrm{Min}_{\mathrm{w},\xi} \quad \tfrac{1}{2}\mathrm{w}^t\mathrm{w} + C\mathbf{1}^t\boldsymbol{\xi} + \boldsymbol{\sigma}^t\mathrm{d} && (8) \\ & \text{subject to} \quad y_i(\mathrm{w}^t\phi(\mathrm{x}_i) + b) \geq 1 - \xi_i && (9) \\ & \qquad\qquad\quad \boldsymbol{\xi} \geq 0 && (10) \end{aligned}$$

The dual of T(d):

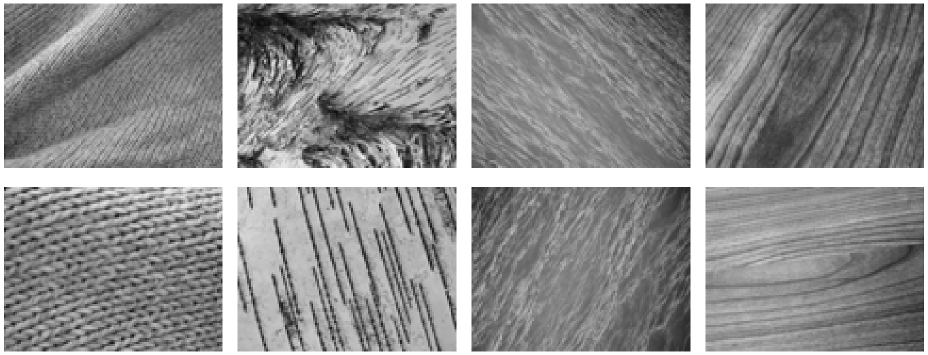
$$\begin{aligned} W(\mathrm{d}) = \underset{\alpha}{\mathrm{Max}} \quad & 1^t\alpha + \sigma^t\mathrm{d} - \tfrac{1}{2}\textstyle\sum_k d_k\alpha^t\mathrm{Y}\mathrm{K}_k\mathrm{Y}\alpha \; (11) \\ \text{subject to} \quad & 0 \leq \alpha \leq C,\; 1^t\mathrm{Y}\alpha = 0 \qquad (12) \end{aligned}$$

$$\frac{\partial T}{\partial d_k} = \frac{\partial W}{\partial d_k} = \sigma_k - \tfrac{1}{2}\alpha^{*t}\mathbf{Y}\mathbf{K}_k\mathbf{Y}\alpha^*$$

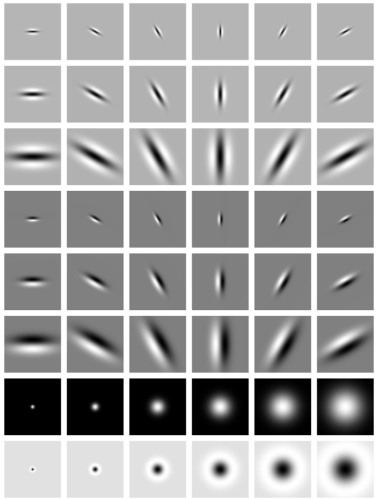
$$\Rightarrow \quad d_k^{n+1} = d_k^n - \epsilon^n(\sigma_k - \tfrac{1}{2}\alpha^{*t}\mathbf{Y}\mathbf{K}_k\mathbf{Y}\alpha^*)$$

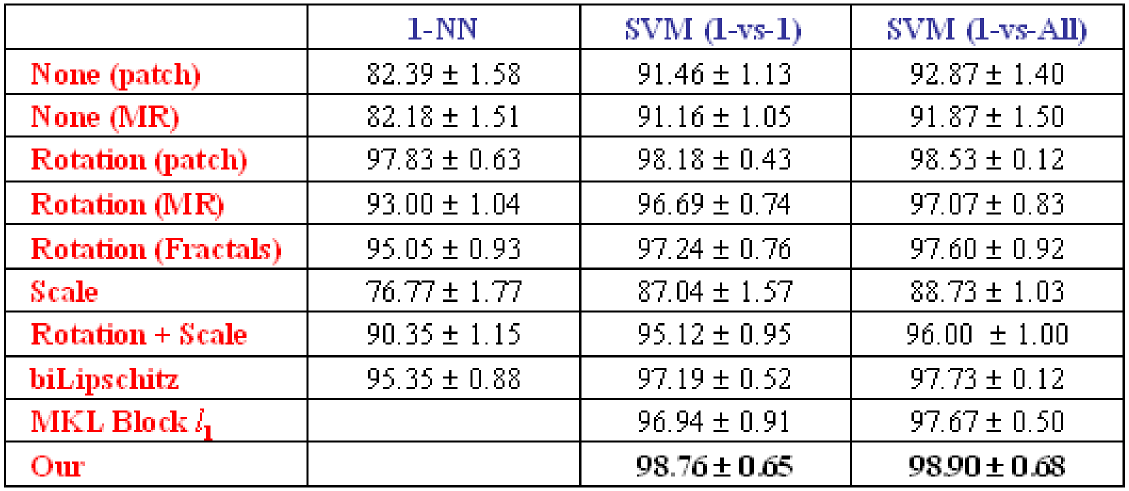
Results:

The UIUC Texture Database: 25 classes, 40 images per class.



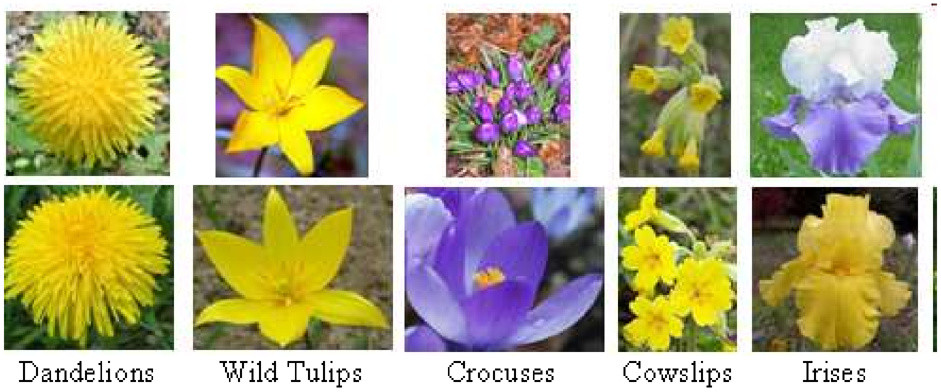
Class 23 Class 3 Class 7 Class 4



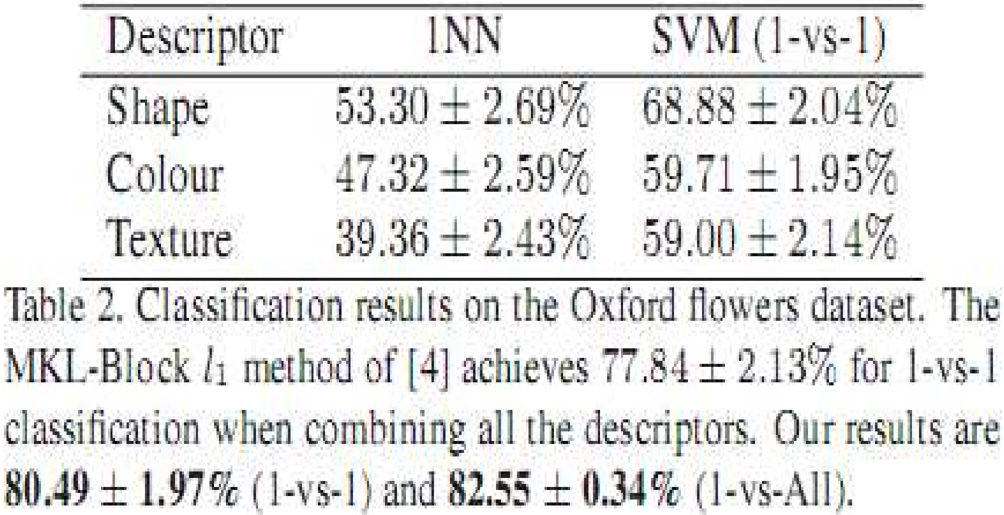


	1-NN	SVM (1-vs-1)	SVM (1-vs-All)
None (patch)	82.39 ± 1.58	91.46 ± 1.13	92.87 ± 1.40
None (MR)	82.18 ± 1.51	91.16 ± 1.05	91.87 ± 1.50
Rotation (patch)	97.83 ± 0.63	98.18 ± 0.43	98.53 ± 0.12
Rotation (MR)	93.00 ± 1.04	96.69 ± 0.74	97.07 ± 0.83
Rotation (Fractals)	95.05 ± 0.93	97.24 ± 0.76	97.60 ± 0.92
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**

The Oxford Flowers Database: 17 classes, 80 images per class.



	Shape	Colour	Texture
Dandelions vs Wild Tulips	3.94	0.00	0.00
Dandelions vs Crocuses	0.42	2.46	0.00
Cowslips vs Irises	1.48	2.00	1.36

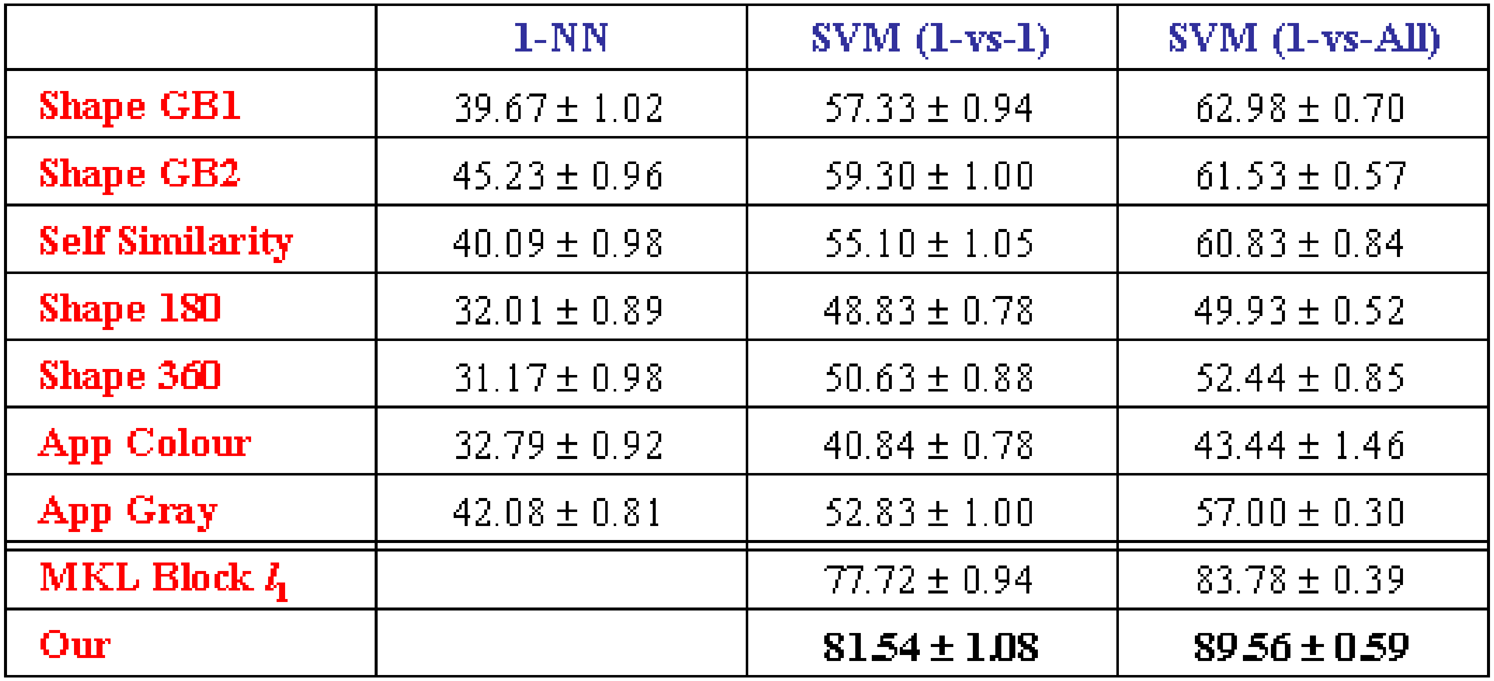


Descriptor	1NN	SVM (1-vs-1)
Shape	53.30 ± 2.69%	68.88 ± 2.04%
Colour	47.32 ± 2.59%	59.71 ± 1.95%
Texture	39.36 ± 2.43%	59.00 ± 2.14%

Table 2. Classification results on the Oxford flowers dataset. The MKL-Block l_1 method of [4] achieves 77.84 ± 2.13% for 1-vs-1 classification when combining all the descriptors. Our results are **80.49 ± 1.97%** (1-vs-1) and **82.55 ± 0.34%** (1-vs-All).

If we force texture weights to be greater than colour weights using **Ad** ≥ **p** we get 81.12 ± 2.09%.

Caltech 101 Object Categorization



	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**