The UoS LAVA group Approach to Generic Image Categoristion

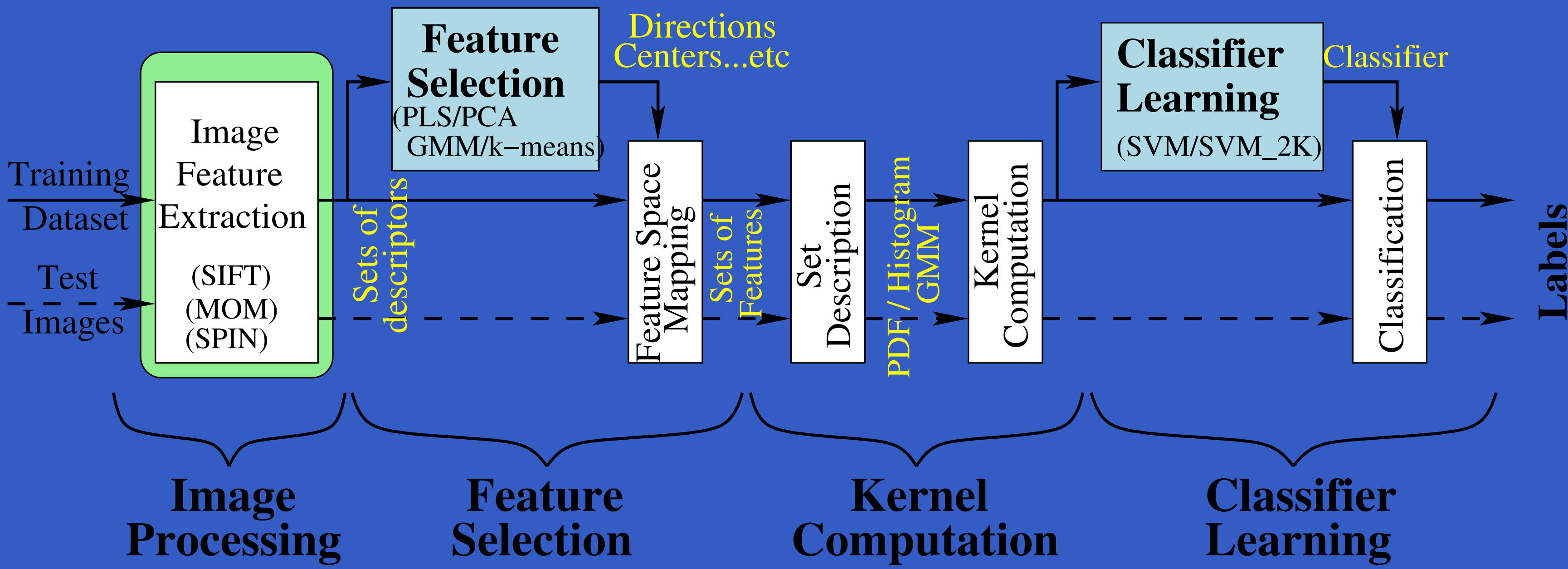
Jason Farquhar, Sandor Szedmak, Hongying Meng, John Shawe-Taylor

jdrf@ecs.soton.ac.uk

School of Electronics and Computer Science

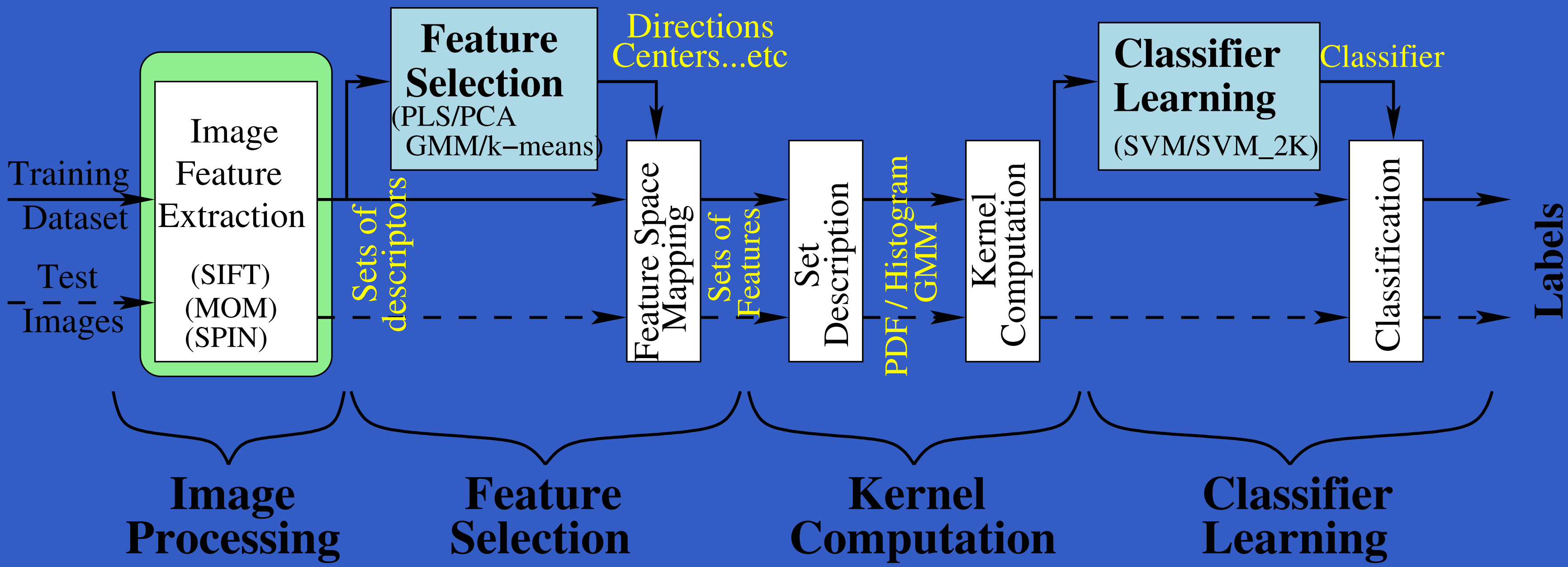
University of Southampton

The “set-of-patches” approach to image categorisation.



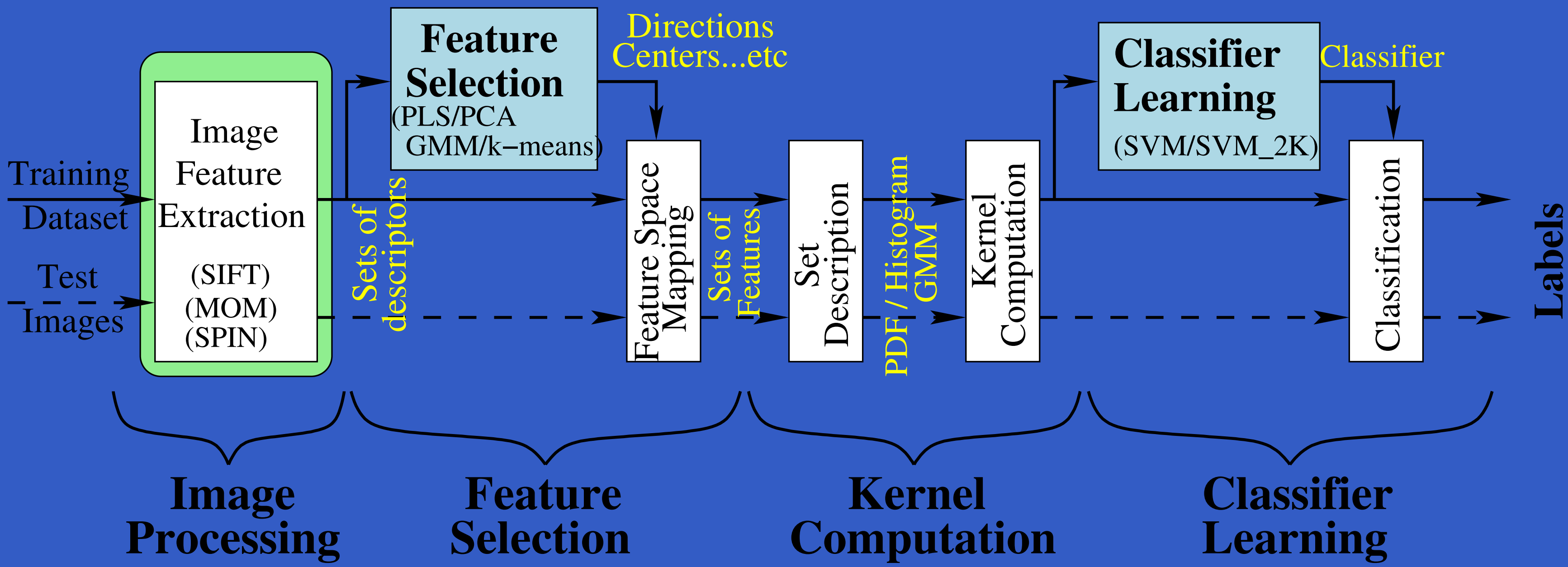
1. Image Processing – extract a set of interesting local patch descriptors from each image.

The "set-of-patches" approach to image categorisation.



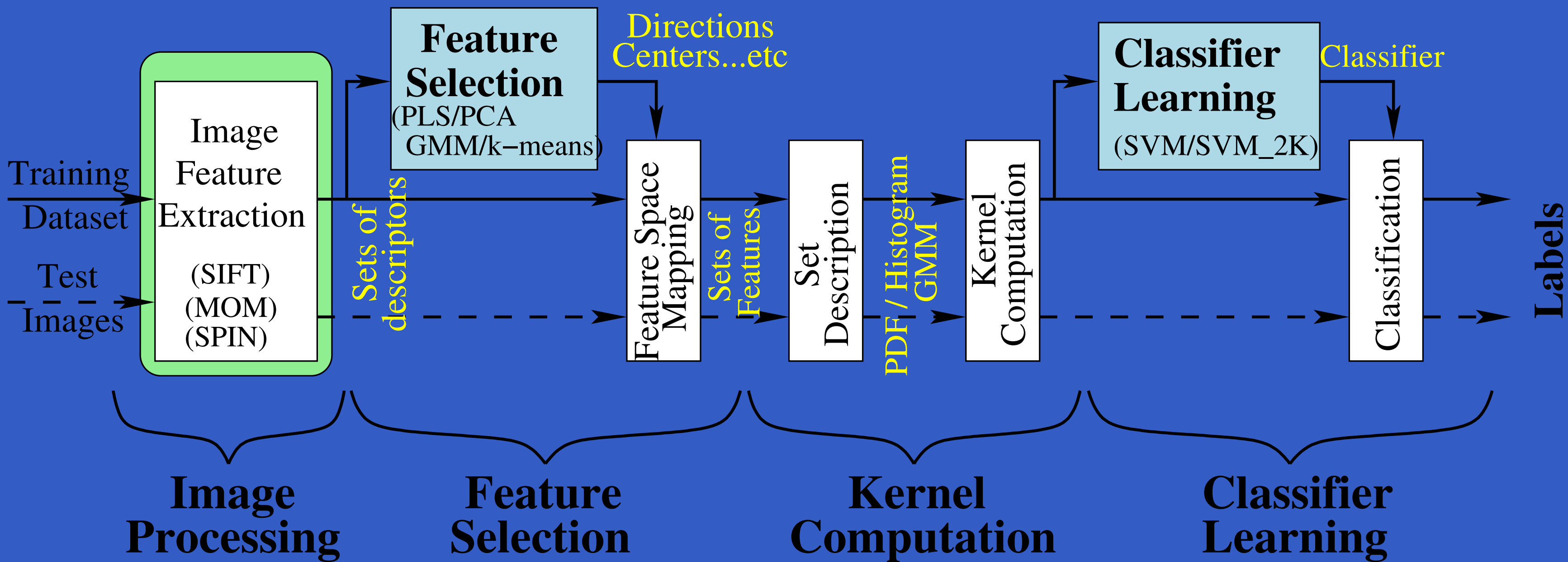
1. Image Processing
2. Feature Selection – identify features of the patch descriptions most useful for categorisation.

The “set-of-patches” approach to image categorisation.



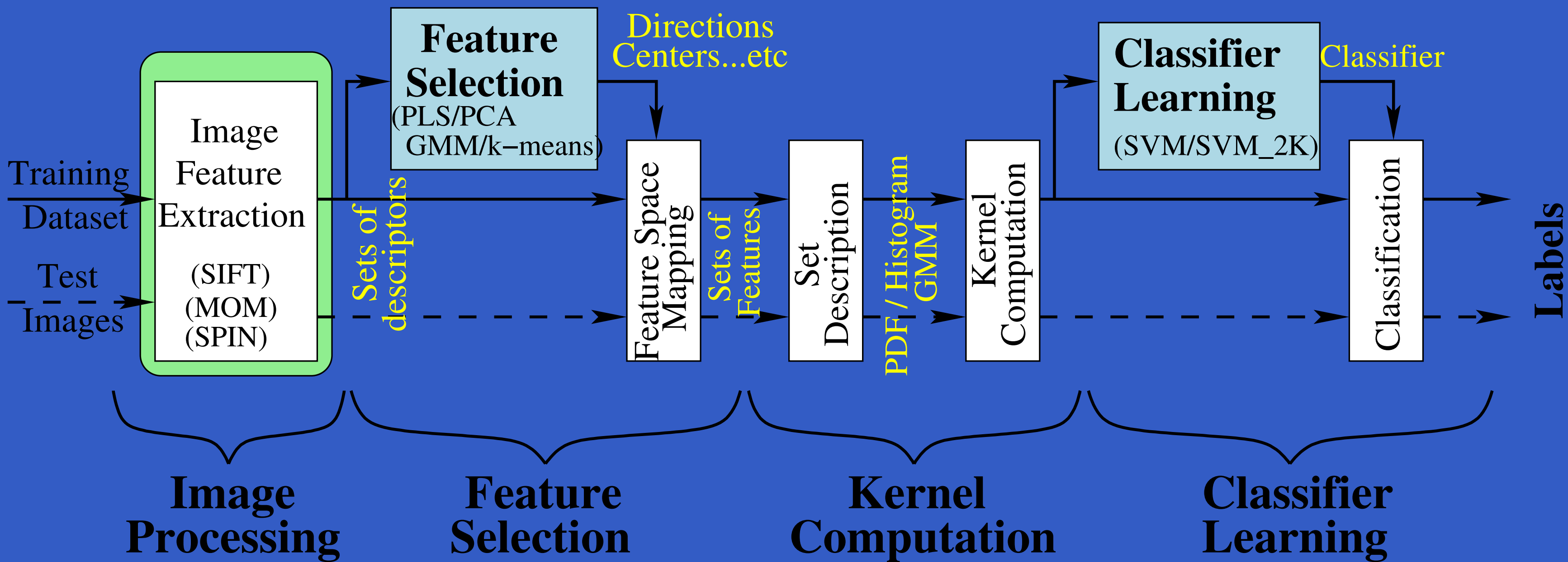
1. Image Processing
2. Feature Selection
3. Kernel Computation – compute a kernel between sets of features in each image.

The “set-of-patches” approach to image categorisation.



1. Image Processing
2. Feature Selection
3. Kernel Computation
4. Classifier Learning – learn a classifier from the computed kernels.

The “set-of-patches” approach to image categorisation.



1. Image Processing

2. Feature Selection
3. Kernel Computation
4. Classifier Learning

Image Processing

Similar to many of the other approaches.

1. Patch detection – identify interesting patches in the image.
   - Lapacian of Gaussians – detects circular “blob like” structures [Lindeberg (1998)].
   - Scale invariant Harris-Affine – detects elliptical patches containing corners or highly textured parts of the image. [Mikolajczyk & Schmid (2004)]

Image Processing

Similar to many of the other approaches.

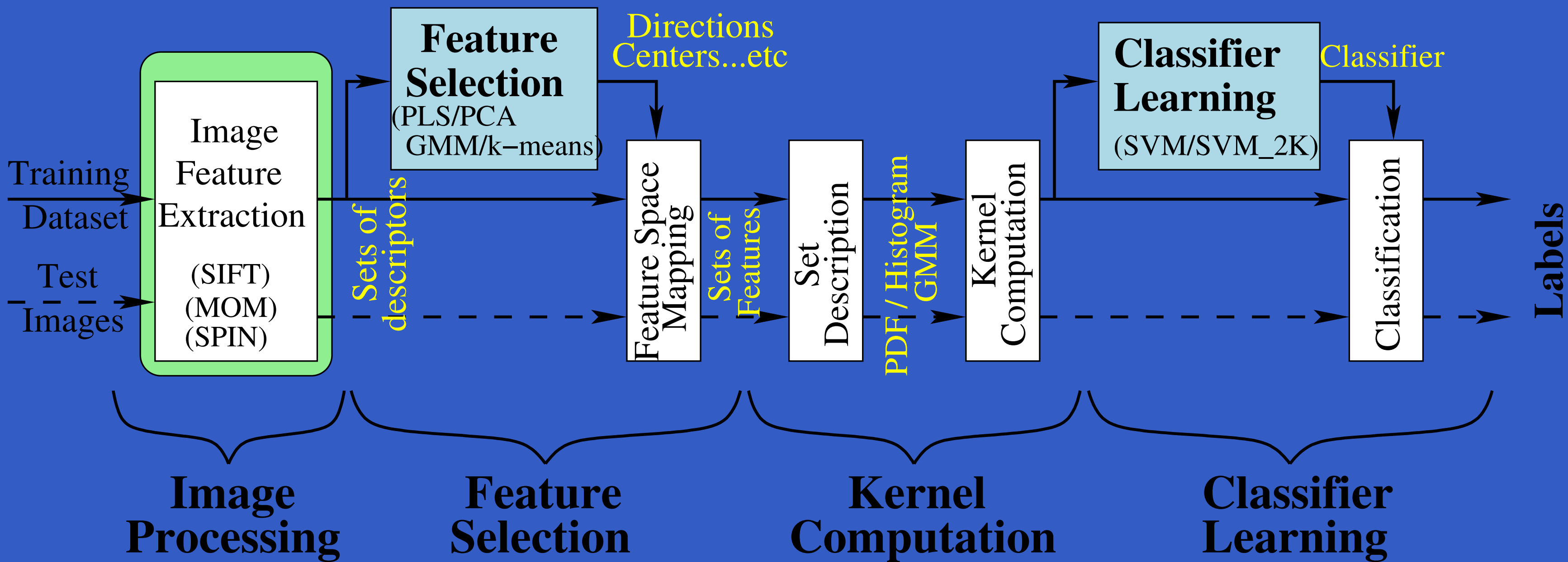
1. Patch detection – LoG or Harris-Affine.
2. Patch description – produce reproducible and robust descriptions of the patches.
   - SIFT – describes circular patch in terms of 8 smoothed directional image gradients at 16 positions within the patch. [D. Lowe 2004]

Image Processing

Similar to many of the other approaches.

1. Patch detection – LoG or Harris-Affine.
2. Patch description – SIFT
3. Output – a set of between 20 and 3000 128d SIFT patch descriptions per image

The “set-of-patches” approach to image categorisation.

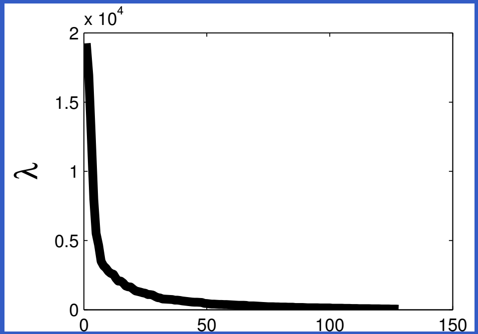


1. Image Processing
2. **Feature Selection**
3. Kernel Computation
4. Classifier Learning

Feature Selection

- SIFT descriptors high dimensional and contain a lot of redundant information.

  Eigenvalue decomposition of the data covariance shows that SIFTs dimensions are highly correlated



Feature Selection

- SIFT descriptors high dimensional and contain a lot of redundant information.
  Thus using full dimensional features,
  - May emphasise un-important noisy variations
  - Significantly increase learning time

Feature Selection

- SIFT descriptors high dimensional and contain a lot of redundant information.
  Thus using full dimensional features,
  - May emphasise un-important noisy variations
  - Significantly increase learning time
- Further, **most** of the detected image patches are in the background so aren't useful for object discrimination.

Feature Selection

- SIFT descriptors high dimensional and contain a lot of redundant information.
  Thus using full dimensional features,
  - May emphasise un-important noisy variations
  - Significantly increase learning time
- Further, **most** of the detected image patches are in the background so aren't useful for object discrimination.

Use feature selection techniques to identify most useful patch features for categorisation.

Feature Selection(2)

Many possible feature selection approaches, methods we have tried are,

1. **Clustering** – GMMs or k-Means used to identify points and regions in feature space which contain useful information.

   Then define new features, e.g. NN region membership or center distances.

2. **Sub-space mapping** – PCA or PLS used to identify feature space directions which contain useful information.

Feature Selection(2)

Many possible feature selection approaches, methods we have tried are,

1. **Clustering** – GMMs or k-Means used to identify points and regions in feature space which contain useful information.

   Then define new features, e.g. NN region membership or center distances.

2. **Sub-space mapping** – PCA or PLS used to identify feature space directions which contain useful information.

Initial experiments indicated that PLS gave best results so only examine this here.

Feature Selection(4)— Partial Least Squares (PLS)

PLS is similar to PCA *except*

- takes account of output labels $Y \in \mathbb{R}^{N \times L}$,
- by finding *pairs* of directions $\mathbf{u}, \mathbf{v}$ which maximise projected data/output cross-covariance,

$$\mathbf{u}, \mathbf{v} = \underset{\mathbf{u},\mathbf{v}}{\operatorname{argmax}} [\operatorname{cov} \{X\mathbf{u}, Y\mathbf{v}\}]^2$$

- This is equivalent to finding the first eigenvector of,

$$\lambda \mathbf{u} = X^T Y Y^T X \mathbf{u}$$

- Can repeat this process after *deflating* X to remove information already used to get $> L$ directions,

$$X_{i+1} = X_i (I - \mathbf{u}_i \mathbf{u}_i^T)$$

(N.B. need to undo the deflations to get final feature directions)

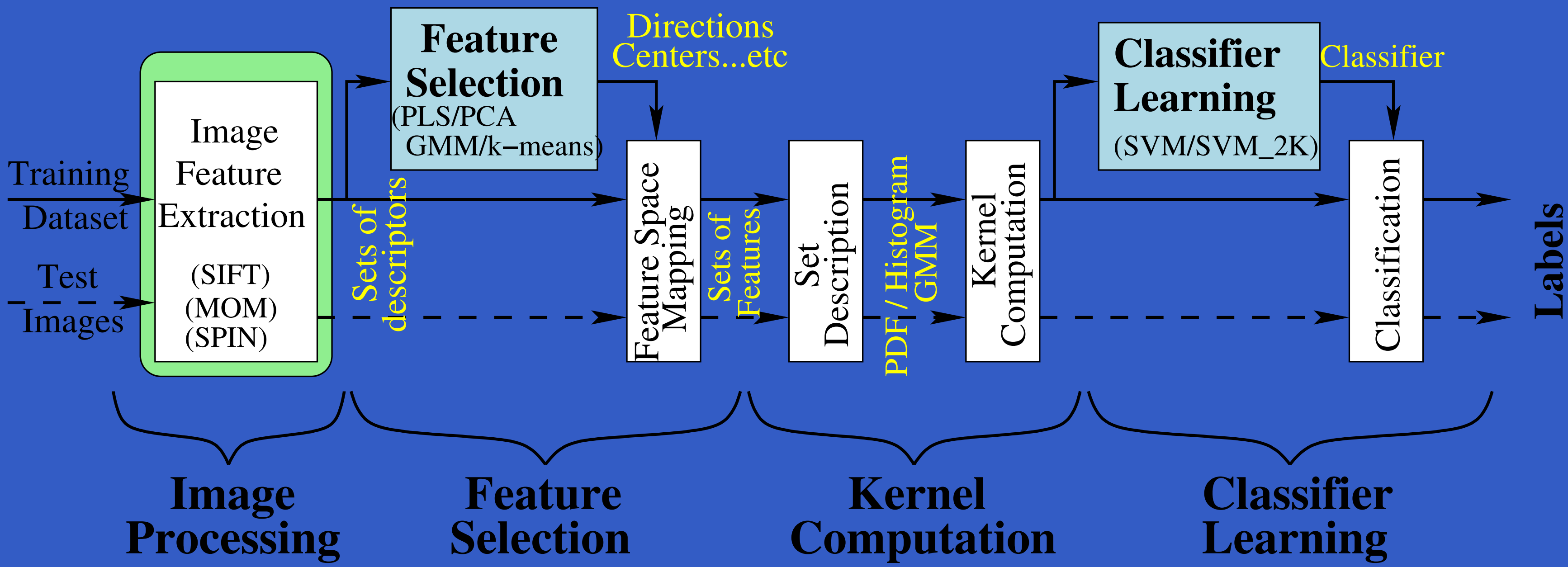
Feature Selection(5)

- Given feature directions $\mathbf{u}_1, \mathbf{u}_2, \ldots \mathbf{u}_{d_2}$ compute new features $\hat{X}$ by projecting X onto these directions,

$$\hat{X} = X * [\mathbf{u}_1 \mathbf{u}_2 \ldots \mathbf{u}_{d_2}]$$

- Output is set of features per image

The “set-of-patches” approach to image categorisation.



1. Image Processing
2. Feature Selection
3. **Kernel Computation**
4. Classifier Learning

Kernel Computation

- Input to this stage is set of feature vectors (one per detected patch) per-image
- Trying to categorise **images** not patches, so need a per-image kernel matrix for classifier learning

Kernel Computation

- Input to this stage is set of feature vectors (one per detected patch) per-image
- Trying to categorise **images** not patches, so need a per-image kernel matrix for classifier learning
- Define a kernel function over sets of feature vectors to compute this kernel matrix. Compute this in 2 stages,
  1. Map from sets of features to a single description
  2. Compute a kernel between the descriptions

Kernel Computation(2) – Set $\rightarrow$ description mapping

- Represent image i's set of features, $\hat{X}_i$, as a parameterised density distribution,

$$\hat{X}_i \longrightarrow \rho(\mathbf{x}|\theta_i)$$

  Many possible density models, e.g. histograms, GMMs

Kernel Computation(2) – Set $\rightarrow$ description mapping

- Represent image i's set of features, $\hat{X}_i$, as a parameterised density distribution,

$$\hat{X}_i \rightarrow \rho(\mathbf{x}|\theta_i)$$

  Many possible density models, e.g. histograms, GMMs

- For ease of kernel computation use a full-covariance Gaussian Probability Density Function (PDF),

$$\hat{X}_i \rightarrow \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_i, \Sigma_i)$$

  Use MAP to fit this model,
    - Provide regularisation for noise tolerance and capacity control,
    - Avoid singularities with low numbers of features.

Kernel Computation(3) – PDF Kernels

- Need a kernel between PDFs
- Many possible PDF similarity measures, e.g. K-L divergence, $\mathcal{X}^2$, Mutual-Information, Fisher-metric, etc.
- Proving these produce valid kernels is difficult

Kernel Computation(3) – PDF Kernels

- Need a kernel between PDFs
- Many possible PDF similarity measures, e.g. K-L divergence, $\mathcal{X}^2$, Mutual-Information, Fisher-metric, etc.
- Proving these produce valid kernels is difficult
- Use the Bhattacharyya affinity which is clearly a kernel,

$$K_{\mathsf{B}}(\mathbb{P}_1(\mathbf{x}), \mathbb{P}_2(\mathbf{x})) = \int_{-\infty}^{\infty} \sqrt{\mathbb{P}_1(\mathbf{x})}\sqrt{\mathbb{P}_2(\mathbf{x})}d\mathbf{x}$$

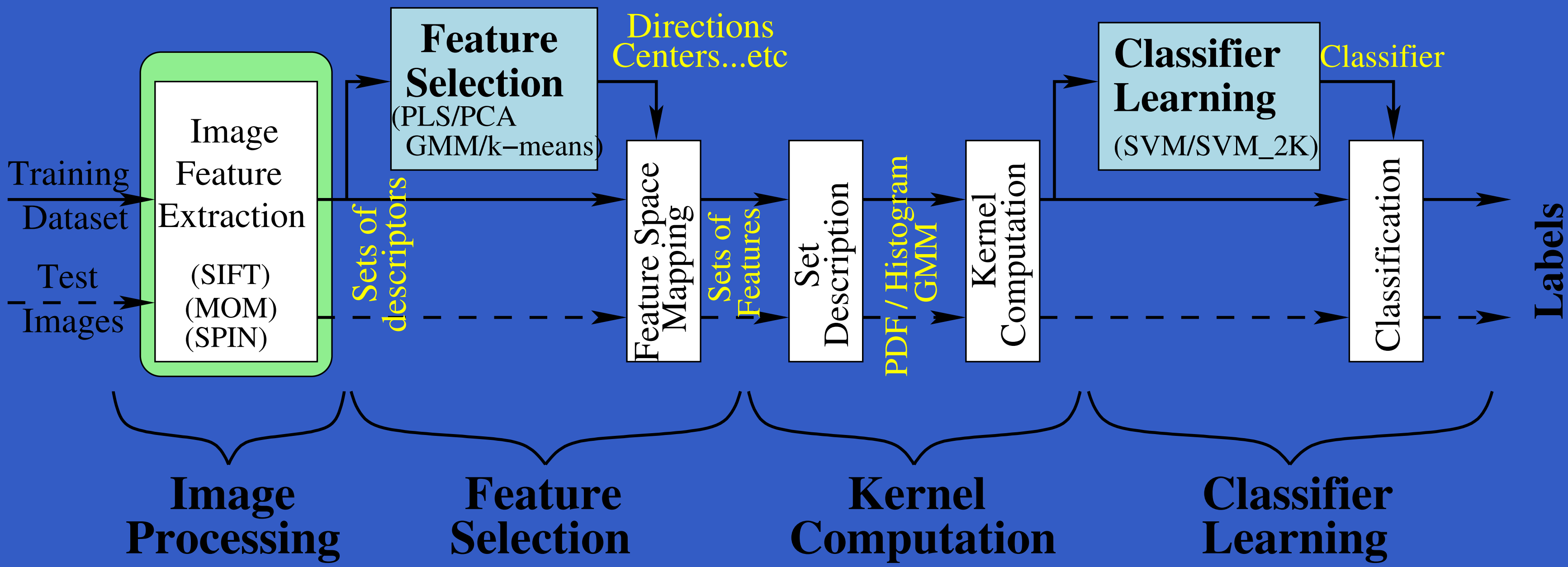
- K_{B} has an analytic solution for pairs of Gaussians

$$K_{\mathsf{B}}(\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \Sigma_1), \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \Sigma_2)) =$$

$$\frac{1}{2}^{\frac{d}{2}} \Sigma_+^{-\frac{1}{2}} \Sigma_1^{-\frac{1}{4}} \Sigma_2^{-\frac{1}{4}} \exp\left[-\frac{1}{4}(\mu_1^T \Sigma_1^{-1} \mu_1 + \mu_2^T \Sigma_2^{-1} \mu_2) - \mu_+^T \Sigma_+^{-1} \mu_+\right]$$

where, $\Sigma_+ = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1}$ and $\mu_+ = \Sigma_1^{-1}\mu_1 + \Sigma_2^{-1}\mu_2$

The “set-of-patches” approach to image categorisation.



1. Image Processing
2. Feature Selection
3. Kernel Computation
4. **Classifier Learning**

Classifier Learning

We have conducted experiments using either a conventional SVM or our extension, SVM_2K.

- SVM_2K is a two-kernel extension of the SVM,
- Uses a synthesis constraint, ψ, to force each sub-SVMs output, h_A, h_B, to be correlated,

$$\psi(h_A(\mathbf{x}_i^A), h_B(\mathbf{x}_i^B)) \leq \delta$$

Classifier Learning

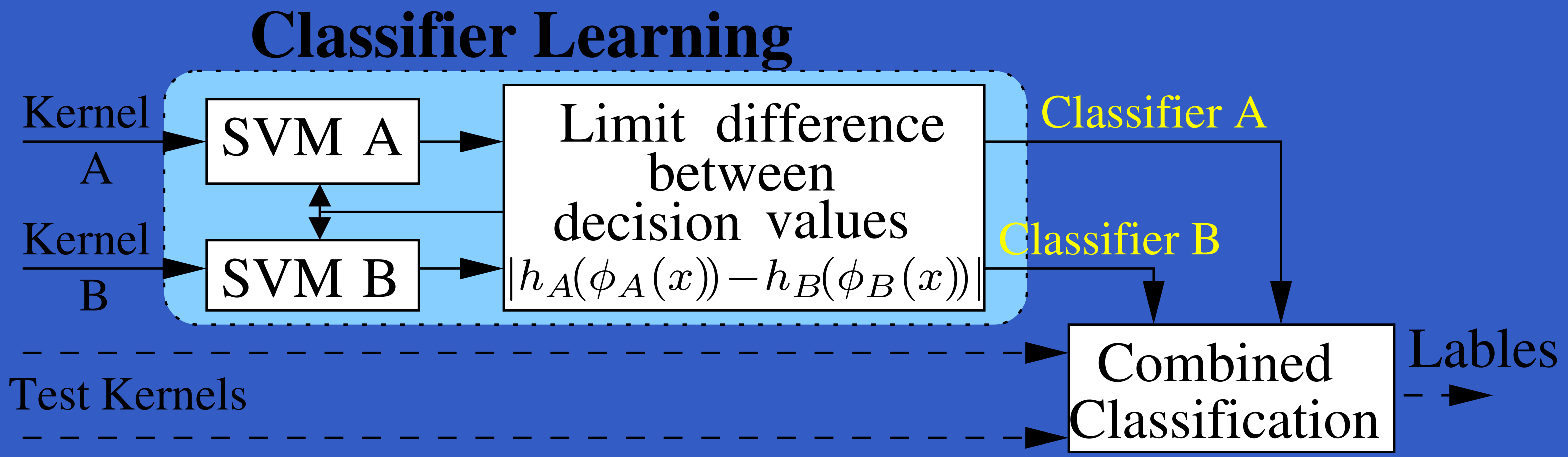
We have conducted experiments using either a conventional SVM or our extension, SVM_2K.

- SVM_2K is a two-kernel extension of the SVM,
- Uses a synthesis constraint, ψ, to force each sub-SVMs output, h_A, h_B, to be correlated,

$$\psi(h_A(\mathbf{x}_i^A), h_B(\mathbf{x}_i^B)) \leq \delta$$

- Idea is that different views of the same object should have correlated signal but (hopefully) uncorrelated noise.
- This is sometimes called co-training or multi-view learning.

Classifier Learning(2) — SVM_2K



The modified primal optimisation problem is,

$$\min \frac{1}{2}(||w_A||_2^2 + ||w_B||_2^2) + \mathbf{1}^T(C_A\xi^A + C_B\xi^B + D\eta)$$

subject to

Synthesis $\dashrightarrow$ $\psi(\langle w_A, \phi_A(x_i^A)\rangle + b_A, \langle w_B, \phi_B(x_i^B)\rangle + b_B) \leq \eta_i + \epsilon,$

subSVM A $\dashrightarrow$ $y_i(\langle w_A, \phi_A(x_i^A)\rangle + b_A) \geq 1 - \xi_i^A,$

subSVM B $\dashrightarrow$ $y_i(\langle w_B, \phi_B(x_i^B)\rangle + b_B) \geq 1 - \xi_i^B,$

$$\xi^A \geq 0,\ \xi^B \geq 0,\ \eta \geq 0,\ i = 1, \ldots, m.$$

Classifier Learning(3) — SVM_2K

- Using $\psi(x) = \text{abs}(x)$ resulting problem has special structure which allows efficient solution, using;
  1. **Augmented Lagrangian**: To eliminate equality dual constraints
  2. **Conditional Gradient**: To solve problem with fixed Lagrangian multipliers
  3. **Linear Programming**: To derive the next approximation of the optimum.
- This approach has linear complexity which the key to the algorithms efficiency!
- It is **over 1000** times faster for this problem than general purpose optimisers.

Results – VOC `test1` EER

Pt. Detector	Learner	M'bikes	Bikes	People	Cars
LoG	SVM	94.9	86.8	83.3	91.3
Harris-Affine	SVM	94.0	85.1	84.1	89.8
LoG + Har-Aff	SVM_2K	**97.2**	**89.5**	**88.1**	**91.3**

`train` and `val` used for training and parameter tuning:

- Feature Selection: 20 dimensional PLS
- Same directions used for all categories
- MAP prior = training set covariance and mean, weighted to represent 10 “virtual” feature vectors
- SVM penalties: determined by cross-validation for each kernel.

Results – VOC test1 EER

Pt. Detector	Learner	M'bikes	Bikes	People	Cars
LoG	SVM	94.9	86.8	83.3	91.3
Harris-Affine	SVM	94.0	85.1	84.1	89.8
LoG + Har-Aff	SVM_2K	**97.2**	**89.5**	**88.1**	**91.3**

- PLS feature selection plus PDF kernels gives good basic performance
- LoG seems to be slightly better than Har-Aff, probably because generates more patches ($\approx$1000 vs. $\approx$100)
- Combining features with SVM_2K further improves performance

Conclusions & Further work

No surprises here,

- Feature selection is critical to performance ...
- as is identifying a good kernel function

Also...

Conclusions & Further work

No surprises here,

- Feature selection is critical to performance ...
- as is identifying a good kernel function

Also...

- Forcing classifiers outputs to correlate can improve performance.

Conclusions & Further work

No surprises here,

- Feature selection is critical to performance ...
- as is identifying a good kernel function

Also...

- Forcing classifiers outputs to correlate can improve performance.

Future Work

- Alternative feature selection techniques
  - per-category feature selection
  - methods to suppress “non-object” patches
- Using more than 2 kernels in SVM_2K