Non-retrieval: blocking pornographic images

Alison Bosson[1], Gavin C. Cawley[2], Yi Chan[2], and Richard Harvey[2]

[1] Clearswift Corporation, 1310 Waterside, Arlington Business Park, Theale,
Berkshire, RG7 4SA, UK. alison.bosson@clearswift.com

[2] School of Information Systems, University of East Anglia, Norwich, NR4 7TJ, UK.
{gcc,yc,rwh}@sys.uea.ac.uk

Abstract. We extend earlier work on detecting pornographic images. Our focus is on the classification stage and we give new results for a variety of classical and modern classifiers. We find the artificial neural network offers a statistically significant improvement. In all cases the error rate is too high unless deployed sensitively so we show how such a system may be built into a commercial environment.

1 Introduction

Dealing with pornography in the workplace is a serious challenge for many large organisations but employing a block-all-images email policy no longer provides a viable solution. Email is more media-based than ever before, and it is common for business mail to contain images such as logos, publicity shots etc. In a commercial environment, an image analysis is required to automatically classify embedded or attached images as acceptable or inappropriate. The problem therefore is the *non*-retrieval of certain types of image.

Although the identification of human skin is commonplace in vision systems, the detection of pictures containing nudity and pornography is a fairly specialised area (some relevant systems include [1–3] and [4, 5]). These systems contain a skin filter which is usually based on colour sometimes with texture as a secondary feature. Skin filters are now fairly standard so we give only a brief explanation Section 2. Here we wish to focus on the classification and deployment of such systems which we describe in Section 3 and subsequent sections.

2 Image Processing

For skin filters based on colour we note that the choice of colour feature usually leads to some discussion of the correct colour space (see [4] for discussions of alternative colour spaces). In practice we [4, 5], and others [2], find that provided there is enough training data and a histogram-based representation of the colour distribution is used then the choice of colour space is not critical. We compute the likelihood ratio $L(\boldsymbol{c}|\text{skin}) = \Pr\{\boldsymbol{c}|\text{skin}\}/\Pr\{\boldsymbol{c}|\text{not skin}\}$ for a quantized colour space. Figure 1 (second from left) shows the likelihood of pixel colours for an example image using a likelihood histogram with 25^3 bins in RGB space.

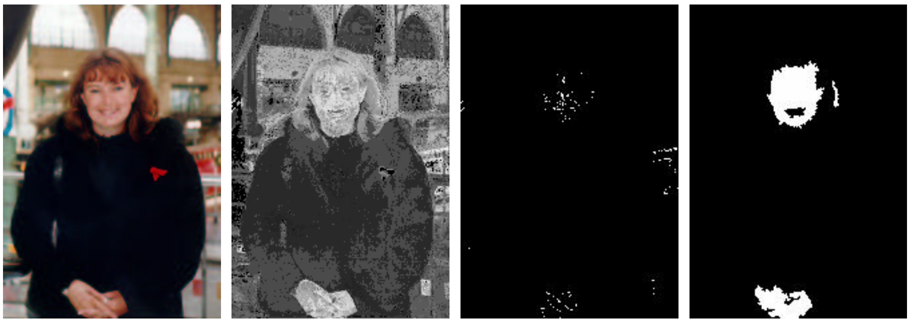

Fig. 1. Original image (left) and associated log-likelihood image (second from left) displayed so that the lowest non-zero likelihood ($\log L = -7.84$) is black and the maximum likelihood, ($\log L = 4.99$) is white; seed points for region growing algorithm (third from left) and final mask (right).

Likelihood images such as the one shown on the right of Figure 1 may be used to produce segments that represent regions of skin by thresholding the likelihood image at the odds set by the ratio of the priors. Care is needed to avoid two common problems: firstly that an image may contain isolated pixels that have the same colour as skin but are associated with the background (examples of such pixels can be seen on the bottom right of the second image in Figure 1) and secondly the likelihood distribution for a particular image is not guaranteed to contain the mode of the training set likelihood distribution which can cause low likelihood values. In the image in Figure 1 for example, part of the skin segment associated with the woman's face appears to have a lower likelihood that those of the bus in the background. However a legitimate assumption is that skin regions are of reasonable area compared to the total image area and contain a locally maximum likelihood value. We therefore use a region-growing algorithm that uses as its seed points likelihood local maxima above a certain threshold. The regions are then grown out to a lower likelihood threshold. A typical sequence of operations is shown in Figure 1.

This likelihood segmentation approach has been tested using a database consisting of 1000 training images and 1000 test images manually segmented to provide the ground truth. The manually generated skin segments are polygonal and include interior regions such as eyes, mouths, hair and shadows that may not be skin coloured. For each putative colour space we compute ROC curves for varying upper and lower thresholds [5]. Along the curves thresholds vary in the interval $(0.1, 0.9)$ of the peak likelihood for that image. Doing this confirms the conclusions in [4] that the HSV colour space gives the best performance. A typical operating point for the HSV system is:

$$\boldsymbol{P} = \begin{bmatrix} p(\bar{s}|\bar{s}) & p(\bar{s}|s) \\ p(s|\bar{s}) & p(s|s) \end{bmatrix} = \begin{bmatrix} 0.82 & 0.18 \\ 0.17 & 0.83 \end{bmatrix} \qquad (1)$$

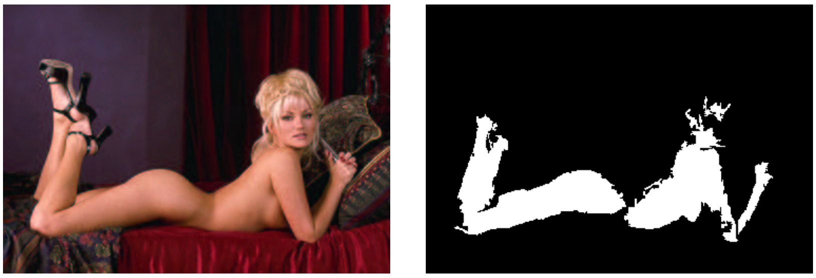

Fig. 2. Example segmentation from [4] and [5]

where, for example, $p(\bar{s}|s)$ denotes the probability that a pixel from a skin region is classified as one from a non-skin region. It is useful to compare these results with [2] in which the authors also conclude that a histogram-based approach is superior to parametric representations of colour distributions. The ROC curves and operating point in [2] are similar to the ones reported here but the definition of skin in [2] is narrower because here shadows, mouths and some hair are contained in the skin masks. The labelled skin set in this paper also includes pornography unlike the public database in [2].

Figure 2 shows an example segmentation for an image drawn from the test set. High resolution images such as this one usually give qualitatively better results than the low resolution images but, provided the test images contain skin colours found in the training set the automatic segmentations are close to those obtained manually. Having identified areas of skin it is necessary to extract higher level features on which to distinguish the classes of image. For this task a larger data set is needed.

These data consist of 11,005 images collected from email and web traffic in a commercial environment. The manually segmented images are a subset of the this set. The data are hand-classified into five categories: 1994 pornographic images (nude pictures that show genitalia or sexual acts); 1973 images of nudity; 1626 images of people (showing people in all poses not covered in other categories showing people); 1803 images of portraiture (which is restricted to head and shoulders portraits of a type prevalent on the web); 1767 graphics images (containing computer generated web graphics, buttons and so on) and 1842 miscellaneous images that could not be classified into one of the previous classes. There is considerable overlap between classes which are subjective. Additionally we define two meta-classes consisting of the unacceptable images (nude plus pornography) and the acceptable (all other images). The proportions of images were chosen to be broadly representative of a range of commercial environments but we know there is considerable variation in these priors between sites. This issue in discussed further later.

There are suggestions for high-level features based on grouping of skin segments [1] that might distinguish these classes but here we have a requirement to process the images speedily so, along with [2] and [3], are interested to try simpler features. For each blob in the image we have computed: area; centroid; the length of the major axis of an ellipse with the same second-order moments as the blob; the minor axis length; eccentricity and orientation of the same ellipse; the area of a convex hull fitted to the blob; the diameter of a circle with the same area as the blob; the solidity (the proportion of the convex hull area accounted for by the blob); the extent (the proportion of the area of a rectangular bounding box accounted for by the blob); the number of colours in the image (graphics are often associated with few colours) and the area of any faces located in the image (we use a commercial face finder to detect and localise faces). These features are ranked using the mutual information of the class given the single feature. Doing this gives the subset of five features that we use: the fractional area of the largest skin blob; the number of skin segments; the fractional area of the largest skin segment; the number of colours in the image and the fractional area of skin that is accounted for by a face.

3 Pattern Recognition

The image processing and feature extraction steps, described in the previous section, produce a vector of features for each image, that we hope will serve to distinguish pornographic from non-pornographic images. The task is then to find the decision rule that optimally separates acceptable from unacceptable images, given a set of labelled examples, $\mathcal{D} = \{(\boldsymbol{x}_i, t_i)\}_{i=1}^{n}\,,\ \boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^d,\ t_i \in \{0, +1\}$, where $\boldsymbol{x}_i$ represents the feature vector for pattern i, and t_i indicates whether pattern i is considered dubious ($t_i = 1$) or acceptable ($t_i = 0$). In the remainder of this section, we briefly describe the four statistical pattern recognition methods compared in this paper.

The output of a generalised linear model [6] is given by $y = g\left(\boldsymbol{w} \cdot \boldsymbol{x} + b\right)$, where, in this case, the *link* function, $g(a)$, is the logistic function, $g(a) = 1/(1 + e^{-a})$. The link function constrains the output of the linear model to lie within the range [0, 1], such that it can be regarded as an estimate of conditional probability, $y_i \approx P(t_i \mid \boldsymbol{x}_i)$. Assuming the target patterns, t_i, are an independent identically distributed (i.i.d) sample drawn from a Bernoulli distribution conditioned on the corresponding input vectors, $\boldsymbol{x}_i$, the negative log-likelihood of the data, known as the cross-entropy, is given by

$$E_{\mathcal{D}} = -\sum_{i=1}^{n} \left\{ t_i \log y_i + (1 - t_i) \log(1 - y_i) \right\}. \tag{2}$$

The vector of optimal model parameters $(\boldsymbol{w}, b)$ is given by the minimum of (2), which may be found via the iterative reweighted least squares algorithm. For multi-class problems, a 1-of-c coding scheme is normally adopted in which the model has c output units, one for each class, and the target for the k^{th} output unit, for a pattern belonging to class $\mathcal{C}_l$, is $t_k = \delta_{kl}$, where δ_{kl} is the Kronecker delta function. The cross-entropy then becomes $E_{\mathcal{D}} = -\sum_{i=1}^{n} \sum_{k=1}^{c} t_i^k \log y_i^k$.

The *softmax* link function, $y_k = \exp(a_k)/\sum_{k'}\exp(a_{k'}))$, is then used to constrain the outputs of the model to lie within the range [0, 1] and to sum to one.

The k-nearest neighbour classifier [7] assigns a test pattern $\boldsymbol{x}$ to the class most strongly represented by the k most similar patterns contained in the training set, according to some distance metric, D, in this case the Euclidean distance, $D_{\mathrm{Euclid}}(\boldsymbol{x}, \boldsymbol{x}') = \|(\boldsymbol{x} - \boldsymbol{x}')\|_2$. The fraction of nearest neighbours belonging to class $\mathcal{C}_a$, provides a simple estimate of *a-posteriori* probability, i.e. $P(\mathcal{C}_a \mid \boldsymbol{x}) \approx k_a/k$. As k tends to infinity this estimate is equal to the true *a-posteriori* probability. The distance metric and k can be chosen so as to minimise the leave-one-out error rate (for two-class problems, k is normally odd in order to prevent ties).

A multi-layer perceptron classifier (see e.g. Bishop [8]), consists of a network of simple neurons (each having a structure similar to a generalised linear model) arranged in layers with strictly feed-forward connections. The parameters of this model, $\boldsymbol{w}$, are determined by minimising a functional, $M = E_{\mathcal{D}} + \alpha E_{\mathcal{W}}$, consisting of a data misfit term, $E_{\mathcal{D}}$, in this case the cross-entropy (2), and a regularisation term, $E_{\mathcal{W}}$, penalising overly complex models. In this study we adopt the regularisation term $E_{\mathcal{W}} = \sum_{i=1}^{W} |w_i|$ (which corresponds to a Laplacian prior over model parameters), where W is the number of parameters. This regularisation term provides both formal regularisation and structural stabilisation as redundant weights are set exactly to zero and can be pruned from the network [9]. The regularisation parameter α, which controls the bias-variance trade-off (e.g. [8]), is integrated out analytically as described by Williams [9].

The support vector machine (e.g. [10]) constructs a maximal margin linear classifier in a high dimensional feature space, $\mathcal{F}(\boldsymbol{\Phi} : \mathcal{X} \to \mathcal{F})$, defined by a positive definite kernel function, $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}')$, giving the inner product $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\Phi}(\boldsymbol{x}) \cdot \boldsymbol{\Phi}(\boldsymbol{x}')$. For this study, we use the anisotropic Gaussian radial basis function (RBF) kernel $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left\{-(\boldsymbol{x} - \boldsymbol{x}')^T \mathrm{diag}(\boldsymbol{\gamma})(\boldsymbol{x} - \boldsymbol{x}')\right\}$, where $\boldsymbol{\gamma}$ is a vector of scaling factors for each attribute. The output of a support vector machine is given by the expansion $f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i t_i \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}) - b$. The optimal coefficients, $\boldsymbol{\alpha}$, of this expansion are given by the maximiser of $W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} t_i t_j \alpha_i \alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, subject to $0 \leq \alpha_i \leq C,\ i = 1, \ldots, n$, and $\sum_{i=1}^{n} \alpha_i t_i = 0$. C is a regularisation parameter controlling a compromise between maximising the margin and minimising the number of training set errors. The bias parameter, b, is chosen in order to satisfy the second Karush-Kuhn-Tucker (KKT) condition, $0 < \alpha_i < C \Rightarrow t_i f(\boldsymbol{x}_i) = 1$. Fortunately many of the coefficients will assume non-zero values, so the kernel expansion will generally be sparse. Estimates of *a-posteriori* probabilities can be obtained via logistic regression on $f(\boldsymbol{x})$ [11]. The regularisation parameter, C, and kernel parameters, such as $\boldsymbol{\gamma}$, are selected so as to minimise an upper-bound on the leave-one-out error [12].

We adopt a 10-fold cross-validation strategy to obtain an almost unbiased estimate of generalisation performance [13]. Table 1 shows the composite confusion matrices for the four classifiers compared, compiled over the test partitions

Table 1. Confusion matrices for generalised linear model (a), k-nearest neighbour (b), multilayer perceptron (c) and support vector machine (d) classification of acceptable and unacceptable images.

(a)

		Observed	
		T	**F**
Predicted	**T**	2787	880
	F	1180	6158

(b)

		Observed	
		T	**F**
Predicted	**T**	3355	814
	F	612	6224

(c)

		Observed	
		T	**F**
Predicted	**T**	3327	764
	F	640	6274

(d)

		Observed	
		T	**F**
Predicted	**T**	3219	705
	F	748	6333

resulting from 10-fold cross-validation. The optimal value of k, for the k-nearest neighbour classifier, was selected in each cross-validation trial to minimise the leave-one-out cross-validation error over the training partition. The mean value of k was 47.8 (std. error 3.71). For the MLP classifier, a single layer of hidden units was used, initially consisting of 32 neurons, giving 225 free parameters. The Bayesian regularisation and pruning algorithm reduced this to a mean of 9.4 units (std. error 0.476) and 43.7 parameters (std. error 1.57) over 10 cross-validation trials. The mean number of support vectors used in the SVM classifier is 4555.2 (std. error of 8.38).

Table 2 summarises the mean classification accuracy of each classifier over the test partitions resulting from 10-fold cross-validation. The k-NN, MLP and SVM classifiers are all superior to the GLM approach, justifying the use of non-linear methods. The relative performance of classifiers systems can be assessed via tests of statistical significance. McNemar's test [14] is used to determine whether the difference in the accuracies of a pair of classifiers is statistically significant. In conducting the necessary set of 6 tests the probability of falsely rejecting the null hypothesis (that there is no significant difference) in at least one test at the 0.05

Table 2. Mean test-partition accuracy by classification method and also area under ROC curves by classification method.

Method	Mean accuracy	Std. err.	Mean area	Std. err.
GLM	0.813	0.004	0.889	0.004
***k*-NN**	0.870	0.004	0.931	0.003
MLP	0.872	0.004	0.937	0.002
SVM	0.868	0.005	0.915	0.004

level of statistical significance is $1-(1-0.05)^6 \approx 0.265$ (assuming that the results of the tests are independent). As we are more concerned in this study with type I error than type II error (accepting a null hypothesis that is false), we should use the *Bonferroni* adjustment [15]; to obtain a statistical significance at the 0.05 level across all 6 tests, $\alpha = 1 - \sqrt[n]{(1-0.05)} \approx 0.0085$. Table 3 summarises the results of McNemar's test of statistical significance. The non-linear methods (k-NN, MLP and SVM) are found to be significantly better than linear methods and the MLP and k-NN significantly superior to the SVM, but the difference in performance between the k-NN and MLP is not statistically significant.

Table 3. Statistical significance of classifier system performance. The upper triangle gives the superior classifier in a pair-wise comparison, statistically superior victors are shown underlined, the lower triangle gives the corresponding level of statistical significance. For example, the entry in the fourth column of the third row indicates that the MLP is superior to the SVM, the third column of the fourth row indicates that the difference in performance is statistically significant according to McNemar's test.

Classifier	**GLM**	**k-NN**	**MLP**	**SVM**
GLM	-	$\underline{k\text{-NN}}$	$\underline{\text{MLP}}$	$\underline{\text{SVM}}$
k-NN	< 0.001	-	MLP	$\underline{k\text{-NN}}$
MLP	< 0.001	0.252	-	$\underline{\text{MLP}}$
SVM	< 0.001	< 0.01	< 0.01	-

Plotting the true-positive rate of a classifier, which is defined as the proportion of positive patterns correctly classified as positive, versus the false-positive rate, which the proportion of negative patterns incorrectly classified as positive, gives the receiver operating characteristic (ROC). The ROC curve then provides a graphical assessment of the performance of a classifier under different misclassification costs, by showing the increasing rate of false-positive errors that must be tolerated in order to improve the true-positive rate. The best classification rules appear toward the upper-left hand corner of the ROC plot. Figure 3 shows the receiver operating characteristic for the four classifiers evaluated in this study. If nothing is known about the true operational *a-priori* probabilities or equivalently misclassification costs, the area under the ROC curve provides a reasonable performance statistic for comparing classifier systems [16]. Table 2 gives the mean area under the ROC curve of each classifier over the test partitions resulting from 10-fold cross-validation. Fitting a convex hull to individual ROC curves gives an area of 0.943, indicating that a combination of classifiers is preferred in uncertain environments.

Multi-layer perceptron networks were also used to solve the six-class pattern recognition task. A further classification stage into the meta-classes gives similar results to those reported with the two-class classifier with that advantage of being able to more sensitively adjust for new class priors. Table 4 shows a composite confusion matrix compiled over the test partitions resulting from 10-

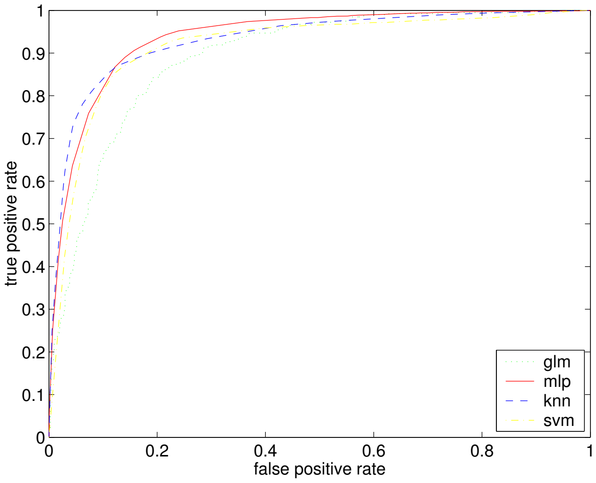

Fig. 3. ROC curves for GLM, k-NN, MLP and SVM classifiers.

fold cross-validation. The MLP classifier achieves a mean test-partition accuracy of 0.520940 with a standard error of 0.005724. The six-class multi-layer perceptron network can also be used to implement the 2-class detector, designating an image as unacceptable if the sum of the *a-posteriori* probabilities for classes "pornography" and "nude" exceeds 0.5. Table 5 shows a composite confusion matrix compiled over the test partitions resulting from 10-fold cross-validation. The MLP classifier achieves a mean test-partition accuracy of 0.872331 with a standard error of 0.003481. As expected, this is almost identical to the accuracy achieved by the two-class multi-layer perceptron classifier.

Table 4. Confusion matrix for multi-layer perceptron classification of images into six categories, under 10-fold cross-validation. The true class runs horizontally and predicted class runs vertically.

	porn	**nude**	**people**	**portrait**	**misc**	**graphics**
porn	1357	765	50	112	124	14
nude	457	809	64	251	175	28
people	5	23	487	62	248	398
portrait	65	190	194	1126	230	116
misc	101	162	332	187	920	177
graphics	9	24	499	65	145	1034

Table 5. Confusion matrix for multi-layer perceptron classification of pornographic images, under 10-fold cross-validation

		Observed	
		T	**F**
Predicted	**T**	3327	765
	F	640	6273

4 Discussion

The image classifier described in this paper is integrated into a mail-based security product MAILsweeper™[3] which is a *content security* solution that sits at an SMTP gateway, assessing email traffic entering and leaving a company and protecting the organisation from mail-borne threats such as viruses, breaches of confidentiality, offensive email content, legal liability and copyright infringement etc. MAILsweeper disassembles emails into their components, for example, zipped email attachments will be unzipped. These are then analysed according to user-defined policies which may be company-wide, department-wide or unique to an individual employee.

The outcome for a particular mail message is determined by its classification. Mails that are *clean* are allowed to pass to the intended recipient but for mails that, for example, contain large attachments, unknown file-types, offensive or confidential material, delivery may be delayed until a user-defined time; the item may be copied; returned to the sender; quarantined or deleted. Notifications and alerts to administrators/senders/recipients may accompany these final message classifications.

The image analyser add-on for MAILsweeper is called PORNsweeper™. As emails are disassembled into their components, any images are passed to PORNsweeper for classification. It first tries to match the incoming image to any of the images in its *exception list.* These are common images, stored as an MD5 hash, that may be pre-classified by an administrator as pornographic or safe. Any incoming image not in the exceptions list is passed to the analyser. If an image is classified as safe the email will be delivered as usual. If, however it is found to be unacceptable, the MAILsweeper system will quarantine the image for the administrators inspection. Any false positives that are blocked may be released from quarantine and may be added to the clean exceptions list to prevent future incorrect classifications. From an administrative perspective, PORNsweeper may be used to constantly monitor all images in emails entering and/or leaving an organisation or it may be used in short bursts, providing a snapshot of email activity.

This paper has provided evidence of a successful skin segmentation algorithm and suggested how this might form part of an automated pornography detector.

[3] All trademarks are the property of their respective owners

The results of the classification experiments show that a non-linear classifier is essential. The choice of classifier depends on implementation issues such as speed and memory usage. The MLP performs well on both these counts and also has the best classification performance. The performance of the SVM is disappointing given the strong theoretical justification of this approach. A possible explanation might be that the model selection criterion unduly favours hyperparameters specifying highly regularised classifiers.

References

1. Fleck, M.M., Forsyth, D.A., Bregler, C.: Finding naked people. In: European Conference on Computer Vision. Volume II., Springer-Verlag (1996) 593–602
2. Jones, M.J., Rehg, J.M.: Statistical color models with application to skin detection. Technical Report CRL 98/11, Compaq Cambridge Research Laboratory (1998)
3. Wang, J., Wiederhold, G., Firschein, O.: System for screening objectionable images using Daubechies' wavelets and color histograms. In Stenmetz, R., Wolf, L.C., eds.: Proc. IDMS'97. Volume 1309., Springer-Verlag LNCS (1997) 20–30
4. Chan, Y., Harvey, R., Smith, D.: Building systems to block pornography. In Eakins, J., Harper, D., eds.: Challenge of Image Retrieval, BCS Electronic Workshops in Computing series (1999) 34–40
5. Chan, Y., Harvey, R., Bagham, J.: Using colour features to block dubious images,. In: Proc. Eusipco 2000. (2000)
6. McCullagh, P., Nelder, J.A.: Generalized Linear Models. 2nd edn. Volume 37 of Monographs on Statistics and Applied Probability. Chapman & Hall (1989)
7. Dasarathy, B.V., ed.: Nearest neighbour (NN) norms: NN pattern classification techniques. IEEE Computer Society, Washingtion, DC (1991)
8. Bishop, C.M.: Neural networks for pattern recognition. OUP (1995)
9. Williams, P.M.: Bayesian regularisation and pruning using a Laplace prior. Neural Computation **7** (1995) 117–143
10. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines (and other kernel-based learning methods). CUP (2000)
11. Platt, J.C.: Probabilities for SV machines. In Smola, A.J., Bartlett, P.J., Schölkopf, B., Schuurmans, D., eds.: Advances in Large Margin Classifiers. MIT Press, Cambridge, Massachusetts (2000) 61–73
12. Joachims, T.: Estimating the generalization performance of a SVM efficiently. Technical Report LS-8 No. 25, Univerität Dortmund, Fachbereich Informatik (1999)
13. Stone, M.: Cross-validatory choice and assessment of statistical predictions. Journal of the Royal Statistical Society **B 36** (1974) 111–147
14. Gillick, L., Cox, S.: Some statistical issues in the comparison of speech recognition algorithms. In: Proceedings, ICASSP. Volume 1. (1989) 532–535
15. Zalzberg, S.L.: On comparing classifiers: pitfalls to avoid and a recommended approach. Data Mining and Knowledge Discovery **1** (1997) 317–327
16. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition **30** (1997) 1145–1159