An Investigation into Applying Support Vector Machines to Pixel Classification in Image Processing

Douglas Clarke, David Albrecht, and Peter Tischer

Monash University,
School of Computer Science and Software Engineering
{dougc, dwa, pet}@csse.monash.edu.au

Abstract. Support Vector Machines (SVMs) have been used successfully for many classification tasks. In this paper, we investigate applying SVMs to classification in the context of image processing. We chose to look at classifying whether pixels have been corrupted by impulsive noise, as this is one of the simpler classification tasks in image processing. We found that the straightforward application of SVMs to this problem led to a number of difficulties, such as long training times, performance that was sensitive to the balance of classes in the training data, and poor classification performance overall. We suggest remedies for some of these problems, including the use of image filters to suppress variation in the training data. This led us to develop a two-stage classification process which used SVMs in the second stage. This two-stage process was able to achieve substantially better results than those resulting from the straightforward application of SVMs.

1 Introduction

In this paper we investigate the application of Support Vector Machines (SVMs) to the image processing domain, in particular the application of SVMs to the classification of features within images. We consider a common problem of image processing, the removal of impulsive noise corruption from images, and develop Support Vector Machine classifiers to detect pixels in an image that are corrupted by impulsive noise.

Impulsive noise is a form of corruption in which the values of a random number of pixels in an image are lost and are replaced by values which are both random and independent of the pixels that are replaced. Impulsive noise corruption can be modelled by salt-and-pepper noise, in which corrupt pixels take on a value at the minimum or maximum pixel intensity (corresponding to a value of either 0 or 255 for 8-bit images) or by a more general model in which corrupt pixels take on a random value distributed uniformly over the range of possible pixel values. This paper deals with noise coming from the second model.

It is desirable to remove impulsive noise corruption from images, both to improve the visual appearance of the image, and to remove corrupt pixel values

G.I. Webb and Xinghuo Yu (Eds.): AI 2004, LNAI 3339, pp. 140–151, 2004.

from further image processing. Many noise filters have been proposed to remove noise corruption from images by replacing pixels corrupted by impulsive noise with an estimate of the pixels' original values in an attempt to reconstruct the original noise-free image. We decompose the problem of removing impulsive noise into two tasks: identifying the noisy pixels in an image, and determining a suitable value with which to replace each noisy pixel.

In this paper, we concentrate on the detection of pixels corrupted by impulsive noise. We treat the detection of noise as a classification problem, and develop Support Vector Machines for categorising the pixels of an image into two groups: pixels corrupted by noise and pixels that represent image structure. A Support Vector Machine classifier is trained on a set of pixels labelled as either "noise" or "uncorrupt" and, following training, its performance is evaluated on a set of unlabelled pixels.

Pixels corrupted by impulsive noise are random-valued and carry no information about the value of the original pixel, and so pixels identified as noise by the SVM can be reconstructed only through interpolation using their non-corrupt neighbours. For some tasks, such as statistical analysis of an image, it is sufficient to identify the corrupt pixels of an image so that they can be omitted from further processing, since pixel replacement through interpolation adds no additional information for analysis.

This paper begins with a brief introduction to Support Vector Machine classifiers. The proposed noise detector is presented in terms of the problems that had to be overcome in the development of an SVM-based approach. In section 2.1 we describe the generation of the training dataset. Next in section 2.2 we improve the distinction between noisy and uncorrupt pixels. In section 2.3 we suppress variation within the class of uncorrupt pixels to improve the accuracy of pixel classification. Finally, in section 2.4 we present a composite approach based on median filtering and a Support Vector Machine, in which a median filter identifies impulsive noise in an image and an SVM corrects misclassifications to prevent image structure from being incorrectly classified as noise.

1.1 Support Vector Machine Classifiers

Support Vector Machines are a machine learning tool for performing such tasks as supervised classification, regression and novelty detection. They have been applied to a wide range of real-world problems including text categorisation, human face detection in images, hand-written character recognition, image retrieval, and the detection of microcalcifications in mammograms $[6, 2, 10, 5]$.

SVM classifiers learn a particular classification function from a set of labelled training examples. The training set consists of n training examples, with each example described by a d-dimensional vector, $\boldsymbol{x} \in \mathbf{R}^d$, labelled as belonging to one of two categories, $y \in \{-1, 1\}$ referred to as the "positive class" and the "negative class". Following training, the result is an SVM that is able to classify previously unseen and unlabelled instances, $\boldsymbol{x}$, into a category based on examples learnt from the training set.

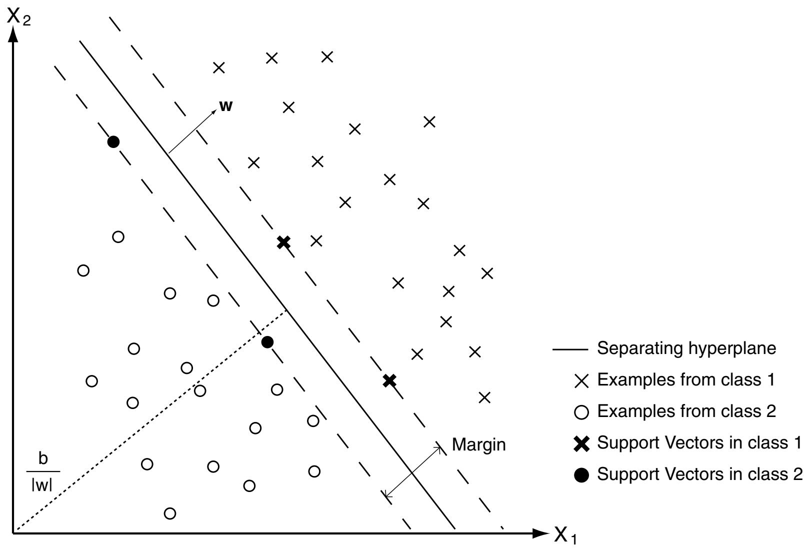

Fig. 1. SVM classification of two classes in a two-dimensional input space

Geometrically, the Support Vector Machine classifier aims to construct a hyperplane that divides $\mathbf{R}^d$ in two, with all training examples in the positive class on one side of the hyperplane, and all examples in the negative class on the other side (see Figure 1). Although there may exist infinitely many hyperplanes that correctly separate the training data, the best SVM classifier is obtained by finding the hyperplane with the maximum "margin". The margin is defined as the distance between the closest training examples in the positive and negative classes to the separating hyperplane [1]. The training examples that determine the margin are known as "support vectors". A trained Support Vector Machine classifies unlabelled points according to the side of the hyperplane on which they fall.

In a typical training set, there will be some examples that unavoidably fall on the wrong side of the hyperplane's decision boundary. In this situation, the separating hyperplane is found by simultaneously maximising the margin between the two classes while minimising the penalty associated with the misclassifications in the training set. The optimum hyperplane, defined by $(\boldsymbol{w} \cdot \boldsymbol{x}) + b = 0$, is found by solving the following quadratic programming problem:

$$\begin{aligned} \min_{w,b,\xi} \quad & \tfrac{1}{2}||\boldsymbol{w}||^2 + C\textstyle\sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(\boldsymbol{x}_i \cdot \boldsymbol{w} + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad i = 1, \ldots, n \end{aligned} \tag{1}$$

where $\xi_i, i = 1, \ldots, n$ are slack variables introduced to allow for examples that fall on the wrong side of the hyperplane, and C is a positive parameter that controls the trade-off between maximising the margin and minimising the training error. This is equivalent to solving the following Lagrangian dual problem, where $\alpha_i, i = 1, \ldots, n$ are the Lagrange multipliers:

$$\begin{array}{ll} \max\limits_{\alpha} & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j \\ \text{s.t.} & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & 0 \le \alpha_i \le C \quad i = 1, \ldots, n \end{array} \tag{2}$$

We evaluated two Support Vector Machine algorithms for training, an implementation of the Sequential Minimal Optimisation algorithm [9] and the SVM$^{\text{light}}$ package [8]. We obtained similar classification results with either algorithm. The results presented in this paper were obtained using SVM$^{\text{light}}$ with a linear kernel and with the Support Vector Machine parameter C determined by SVM$^{\text{light}}$.

2 SVM Detection of Impulsive Noise in Images

Support Vector Machine classifiers learn a particular classification function from a set of labelled training examples. We created training and testing datasets by adding between 1%–20% impulsive noise corruption to an existing image, with the assumption that the original image was initially free of noise and the only noise contained in the image was the impulsive noise that was added.

We examined the effects that a number of parameters had on the performance of the SVM noise classifier. For each set of variables to be evaluated, an array of ten noise-corrupted versions of the same underlying image was generated, in which each image contained an equal level of impulsive noise. The first two images in each set were used to train two Support Vector Machine classifiers independently, while the remaining eight images in the set were used to evaluate the performance of the two trained SVM classifiers. The relative performance of SVMs with different parameter settings was evaluated by calculating the "generalisation performance" of the SVM. The generalisation performance for each SVM consisted of the percentage of noisy pixels and valid pixels in the testing data that were misclassified by the SVM. The misclassification rate was an average of the misclassifications in the eight test images for the two SVM classifiers that were trained.

The examples in the training and test sets were described in a nine-dimensional input space that consisted of the values of each pixel and its eight surrounding neighbouring pixels. This input space was selected because it retained the relationship of each pixel to its neighbours, which we hoped would allow a Support Vector Machine to identify patterns that represented structure such as lines, edges, and corners, and enable these patterns to be distinguished from impulsive noise.

2.1 Training Set Selection

The performance of a Support Vector Machine classifier is dependent on its training and so it is important that the training dataset consists of examples that are representative of the points from the positive and negative classes. We faced two problems in the selection of our training data: reducing the size of the training set and balancing the proportion of training examples from the noise and uncorrupt classes.

The images used in training and testing had dimensions of 512×512 pixels. We found that it was infeasible to create the training set by including every pixel from an image of these dimensions due to the time required to train an SVM on a dataset of this size (over $250,000$ examples). An initial attempt to reduce the size of training dataset was to downsample the training image from 512×512 pixels to a smaller 64×64 pixels. Impulsive noise was then added to the resized image and the training data was generated by taking every pixel in this image to form a training set of roughly 4000 examples. However, this generated a training set with an uneven proportion of examples from the two classes—the proportion of training examples labelled as noise matched the proportion of impulsive noise that was added to the original image. A Support Vector Machine trained on such a biased dataset was found to get caught in a local optimum in which it classified every example into the class that dominated the training set; if the training image contained only 2% impulsive noise, the resulting SVM classified every test example as an uncorrupt pixel.

The problems of training set imbalance and training set size were both overcome by creating the training data by randomly selecting 2000 uncorrupt and 2000 noisy pixels from a full-sized image of dimensions 512×512 pixels. This led to a small training dataset containing 4000 examples with an equal number of examples coming from both classes. In addition, taking a subset of pixels from the full-sized image had the advantage that the spatial properties of pixels in the training image were not affected by resizing the image.

2.2 Improving Class Separation

The generalisation performance of a Support Vector Machine classifier is highly dependent on the positive and negative classes being linearly separable in the input space. If the training set contains two classes that are clearly separable the resulting model will contain few Support Vectors and we conjecture will generalise well to the test set.

Our datasets contained considerable overlap between the "noise" and "uncorrupt" classes. Since we were dealing with noise that takes on a random intensity, there is a probability that a pixel could be corrupted by noise with a value that is close to, or identical to, the original pixel value. This leads to training data that is inseparable, since the same example may be labelled as both a valid pixel and as noise in the dataset. Furthermore, it is somewhat subjective as to whether a pixel that differs only marginally from its neighbours represents noise or is a small feature within the image.

To overcome this problem, examples in the training and test data were labelled as noise only if the original pixel's value had been changed by more than a certain threshold. For example, if a pixel of value x was replaced by noise of value x', the example would be labelled in the dataset as follows (given threshold T):

$$\text{label }(x) = \begin{cases} \text{uncorrupt}, & |x - x'| < T \\ \text{noise}, & |x - x'| \geq T \end{cases} \tag{3}$$

For the remainder of this paper a noise threshold of 25 is used. Although this value excludes one-fifth of the possible values for impulsive noise in an 8-bit greyscale image, it was selected because we believe that impulsive noise with a value that differs from the original pixel value by 25 or less would be visually imperceptible in the image. The use of this threshold produced an SVM with good generalisation performance.

2.3 Reducing the Variation Within the Classes

The detection of impulsive noise in image data is complicated by the large amount of background variation within images, which requires a large training set to define. However, if both training and test images were filtered to remove background structure, while retaining the possible noise in the image, then the task of the Support Vector Machine would be greatly simplified and, we believe, the SVM would generalise better to images outside the training set.

We evaluated the use of a filter to remove the background structure of an image. Training and test images were filtered by either a highpass FIR filter or the Immerkær background-removal filter [7]. The highpass filter was selected for its ability to emphasise the impulsive noise in an image—which appears as high-frequency data in the frequency-domain—making it stand out against the background of the image. The Immerkær filter was proposed as part of a method for estimating the level of additive noise in an image in which the predictable image structure is first removed from the image, leaving only noise (and fine image detail) remaining. Image structure is removed by filtering the source image, X, with the following convolution kernel, resulting in the image Y:

$$Y = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix} \otimes X \tag{4}$$

The convolution kernel is based on two Laplacian filters, and has the effect of removing all constant, linear, and quadratic variation in the intensity of pixels within the local window.

The effect of applying the highpass filter and Immerkær filter on an initial noise-corrupted image is shown in Figure 2. Figure 4 shows the generalisation performance of a Support Vector Machine trained and tested on images with no filtering, with a highpass filter, and with an Immerkær filter. The median difference is described in the next section.

Fig. 2. Application of pre-processing filters on "Lena" containing 2% noise

2.4 Multistage Classification Using SVMs

Motivated by the significant improvement that the Immerkær background-removal filter had on the generalisation performance of a Support Vector Machine, we had the idea of processing images with the median filter. The median filter is a popular non-linear filter used for removing impulsive noise from images, and is used at the core of many noise removal algorithms. The median filter is capable of effectively suppressing impulsive noise, however, it does so at the expense of the non-corrupt pixels in the image [3, 4]. In particular, the median filter is unable to distinguish fine lines from impulsive noise, and so lines in the image are removed or distorted. We recognised that subtracting the median filtered image from its non-filtered counterpart would result in an image containing only the noise in the original image as well as any distortions introduced by the median filter where it misclassified image structure as noise. A Support Vector Machine classifier could then be trained to separate noisy pixels from features removed by the median filter, without the complication of background image variation.

The median filter replaces every pixel with the median value of a surrounding window of the image. For a $(2N + 1) \times (2N + 1)$ window centered around pixel $x_{i,j}$, the median filter performs the following operation:

$$y_{i,j} = \text{median}(x_{i-N,j-N}, \ldots, x_{i,j}, \ldots, x_{i+N,j+N}) \tag{5}$$

The "median difference filter" that we developed is described below for source image X,

$$Y = |X - \text{Median}(X)| \tag{6}$$

We used a median difference filter with a window size of 3×3 pixels in our research.

Figure 3 shows the result of applying the straight median filter to an initial noise-corrupted image, and the corresponding median difference image. Figure 4 presents the generalisation performance of a Support Vector Machine that is applied to the median difference image (corrupted with 2% impulsive noise) and shows that the median difference SVM classifier clearly outperforms other SVM approaches.

Fig. 3. Demonstration of median difference filter on "Lena". For illustrative purposes the image contains 10% noise

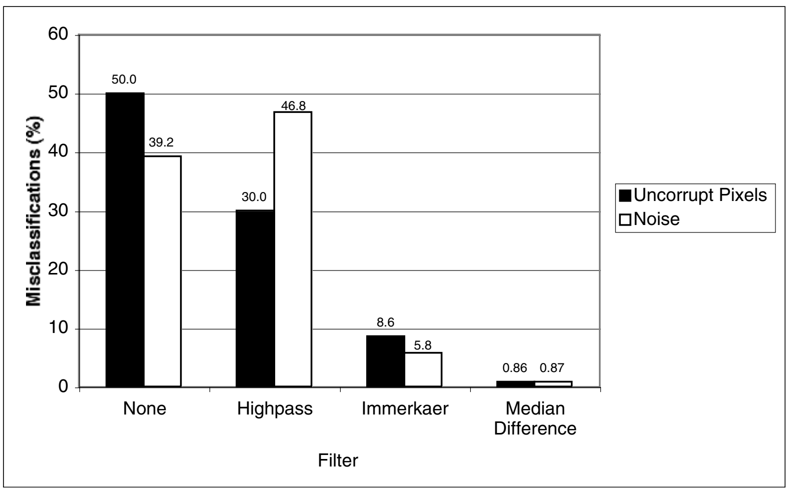

Fig. 4. Plot of generalised error rate for classification of uncorrupt and noisy pixels in test images containing 2% impulsive noise

Fig. 5. The set of test images used for evaluation of proposed SVM technique

3 Comparative Results

In this section we compare the performance of our proposed Support Vector Machine-based impulsive noise detection algorithm against a noise detector based upon the median filter for images corrupted by various levels of impulsive noise. The median filter classifier considers a pixel to be noise if the difference between its value and the median value of its neighbours is greater than a given threshold, T:

$$\text{class}(x_{i,j}) = \begin{cases} \text{noise}, & |x_{i,j} - \text{median}(i,j)| > T \\ \text{uncorrupt}, & |x_{i,j} - \text{median}(i,j)| \leq T \end{cases} \tag{7}$$

Many noise filtering algorithms, including those that treat the detection and removal of noise as a single problem, make use of the median filter—or median filter classifier—to perform implicit noise detection.

Our experiments were performed on 8-bit greyscale images of dimensions 512×512 pixels, which were corrupted with 1%, 2%, 5%, 10%, and 20% random-valued impulsive noise. The Support Vector Machine classifier was trained and tested on images that had been preprocessed with the median difference filter described in Section 2.4. The results in this paper were obtained using the image "Lena", but the trend observed with this image was also observed with other images.

Table 1 presents the 95% confidence interval for the percentage of misclassified pixels in the "noise" and "uncorrupt" classes, using both the proposed SVM classifier and the median filter classifier. These results show that the proposed SVM-based impulsive noise detector is able to perform well even when the level of impulsive noise corruption is high. A summary of the statistical significance of the proposed SVM classifier's better noise detection is given in Table 2. With the exception of the classification of image structure in images corrupted by 1% noise, all tests were statistically significant at below the 5×10^{-5} level of significance. We conjecture that even images with no noise corruption contain pixels that differ significantly from their neighbours, and these pixels are incorrectly classified as noise by the SVM, leading to the slightly higher misclassification rate for image structure in images containing 1% .

The orginal training dataset consisted of 4000 pixels from a 512×512 pixel image. For an image corrupted with 2% impulsive noise, the training set contains only 0.78% of the uncorrupt pixels from the image. Thus the results in Tables 1 and 2 show that the proposed SVM classifier is generalising well from a relatively small training set.

To verify that the Support Vector Machine was indeed learning to identify the characteristics of noise, and not simply learning to identify corruption in the training image, a Support Vector Machine that was trained on the image "Lena" was used to identify noise in the images "Boat" and "Mandrill" (see figure 5). Two Support Vector Machine classifiers were each trained on two separate noise-corrupted versions of the image "Lena" containing 2% impulsive noise. The two trained SVMs were then evaluated on noise classification of eight noise-corrupted versions of the images "Lena", "Boat" and "Mandrill". The median difference

Table 1. Comparison of median filter noise detector versus proposed SVM noise classifier

Noise Level	Median Filter Classifier		Proposed SVM Classifier	
	Misclassified Noise	Misclassified Structure	Misclassified Noise	Misclassified Structure
1%	3.65–4.37%	0.735–0.763%	0.692–0.995%	0.770–0.778%
2%	3.78–4.15%	0.996–1.05%	0.637–0.863%	0.914–0.928%
5%	4.75–5.20%	2.08–2.15%	0.829–0.975%	1.17–1.19%
10%	5.87–6.17%	4.79–4.92%	1.05–1.17%	1.46–1.50%
20%	7.81–8.04%	12.3–12.4%	1.65–1.76%	2.23–2.30%

Table 2. Tests for statistical significance of performance improvement of proposed SVM noise detector

Noise Level	P-Value (Noise)	P-Value (Structure)
1%	2.78×10^{-7}	9.98×10^{-1}
2%	3.99×10^{-10}	4.83×10^{-5}
5%	1.34×10^{-10}	4.85×10^{-11}
10%	5.97×10^{-13}	5.89×10^{-13}
20%	3.15×10^{-14}	3.42×10^{-16}

image was used for all training and test images. Table 3 shows the 95% confidence interval for the percentage of pixels misclassified by the proposed SVM. For comparison, we also include results for an SVM that has been trained and tested on the same underlying image. For example, an SVM that was trained on the image "Boat" containing 2% impulsive noise was tested on eight other versions of "Boat" containing 2% noise.

Table 3. Generalisation of proposed SVM to noise detection outside training image

Image	Trained on "Lena"		Trained on different noise-corrupted version of same image	
	Misclassified Noise	Misclassified Structure	Misclassified Noise	Misclassified Structure
Lena	0.637–0.863%	0.914–0.928%	0.637–0.863%	0.914–0.928%
Boat	1.41–1.62%	2.13–2.16%	1.63–1.86%	1.77–1.81%
Mandrill	5.95–6.48%	14.9–15.7%	11.4–12.1%	6.59–6.93%

We note that performance on "Lena" was better than performance on the other two images. The classifiers trained on "Lena" and "Boat" were more accurate in classifying noise than in classifying structure. However, the reverse was true for "Mandrill". "Mandrill is an image with a large amount of texture and detail. We conjecture that the high level of variation in the image structure in

"Mandrill" causes the classifiers trained on this image to be biased towards capturing structural details correctly. This may explain why the classifier's trained on "Lena" were more accurate in classifying noise on "Mandrill" than the classifiers trained on "Mandrill" itself. Also we conjecture that it may be necessary to include more attributes or to use an SVM with a non-linear kernel to improve the learning of the underlying image structure.

4 Conclusions

In this paper, we have approached the removal of impulsive noise from an image as a classification problem, and have proposed an impulsive noise detection algorithm based on Support Vector Machine classifiers. The performance of the Support Vector Machine classifier was improved by applying domain knowledge in order to generate a balanced training set, to reduce the overlap between the positive and negative classes, and to suppress the variation of examples within each class.

The use of domain knowledge led to a two stage process in which a traditional noise filtering algorithm, the median filter, was used as a first stage to identify impulsive noise in an image, and a Support Vector Machine was then used to correct misclassifications by the median filter. This composite noise classifier performed significantly better than either the median filter or a Support Vector Machine classifier individually. Further work could investigate the replacement of the median filter as a first-stage noise detector with a filter that is appropriate to the level of noise and type of noise in an image.

The results that we have obtained with multi-stage noise detection indicate that Support Vector Machines may be worthwhile as the second stage in a two-stage classification process. Such an approach could be applied to areas outside noise detection, where a Support Vector Machine is used to improve upon an existing technique which, as we have shown with the median filter, is simply used as a black box first-stage classifier. Further work could also investigate the use of other machine learning classifiers to implement the second-stage of the process.

References

1. Burges, C.: A tutorial on Support Vector Machines for pattern recognition. Data Mining and Knowledge Discovery **2** (1998) 121–167
2. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000)
3. Davies, E.R.: Edge location shifts produced by median filters: theoretical bounds and experimental results. Signal Process. **16** (1989) 83–96
4. Davies, E.R.: A remanent noise problem with the median filter. In: Proc. 11th IAPR International Conference on Image, Speech and Signal Analysis. Volume 3. (1992) 505–508

5. El-Naqa, I., Yang, Y., Wernick, M., Galatsanos, N., Nishikawa, R.: A Support Vector Machine approach for detection of microcalcifications. IEEE Transactions on Medical Imaging **21** (2002) 1552–1563
6. Hearst, M., Dumais, S., Osuna, E., Platt, J., Schölkopf, B.: Support Vector Machines. IEEE Intelligent Systems **13** (1998) 18–28
7. Immerkær, J.: Fast noise variance estimation. Computer vision and image understanding **64** (1996) 300–302
8. Joachims, T.: Making large-scale SVM learning practical. In Schölkopf, B., Burges, C., Smola, A., eds.: Advances in Kernel Methods - Support Vector Learning, MIT Press (1999) 169–184
9. Platt, J.: Chapter 12. In Schölkopf, B., Burges, C., Smola, A., eds.: Advances in Kernel Methods - Support Vector Learning, MIT Press (1999) 185–208
10. Tong, S., Chang, E.: Support Vector Machine active learning for image retrieval. In: Proceedings of the ninth ACM international conference on Multimedia, ACM Press (2001) 107–118