# Feature Selection from Huge Feature Sets

José Bins
Faculdade de Informática
Pontifícia Universidade Católica (RS)
Porto Alegre, RS 90619-900, Brazil
Bins@inf.pucrs.br

Bruce A. Draper
Computer Science Department
Colorado State University
Fort Collins, CO 80523, USA
Draper@cs.colostate.edu

## Abstract

*The number of features that can be computed over an image is, for practical purposes, limitless. Unfortunately, the number of features that can be computed and exploited by most computer vision systems is considerably less. As a result, it is important to develop techniques for selecting features from very large data sets that include many irrelevant or redundant features. This work addresses the feature selection problem by proposing a three-step algorithm. The first step uses a variation of the well known Relief algorithm [11] to remove irrelevance; the second step clusters features using K-means to remove redundancy; and the third step is a standard combinatorial feature selection algorithm. This three-step combination is shown to be more effective than standard feature selection algorithms for large data sets with lots of irrelevant and redundant features. It is also shown to be no worse than standard techniques for data sets that do not have these properties. Finally, we show a third experiment in which a data set with 4096 features is reduced to 5% of its original size with very little information loss.*

## 1. Introduction

The number of features that can be computed over an image is, for all practical purposes, limitless. Examples of common image features include Fourier coefficients, PCA eigenvectors, Zernike moments, Gabor energy functions, Wavelet responses, probe sets, and histogram/correlogram features, not to mention simple set features like the mean and standard deviation of an image. Moreover, this list of popular features barely scratches the surface; in [23], Tieu and Viola compute $45,000$ image features without using any of those mentioned above. The feature set available for object classification is therefore very large.

Unfortunately, large feature sets are problematic. Real-time systems cannot afford the time to compute or apply them. More relevent to this paper, statistical model fitting and/or supervised learning systems generally do not have enough labeled training instances to fit accurate models over very large feature spaces, due to finite sample effects [9]. At the same time, in many cases it is difficult or impossible to know without training which features are relevent to a given task, and which are effectively noise. As a result, the ability to select features from a huge feature set is critical for computer vision.

Given a feature set of size $M$, the feature selection problem is to find a feature subset of size $n$ ($n << M$) that maximizes the system's ability to classify object instances. Finding the optimal set of features is usually intractable [14], and many problems related to feature selection have been shown to be NP-hard ([2], [8]). For most practical problems, an optimal solution can only be guaranteed if a monotonic criterion for evaluating features can be found, but this assumption rarely holds in the real-world [13]. As a result, we are forced to find heuristic solutions that represent a trade-off between solution quality and time.

We are interested in the problem of feature selection in the context of computer vision. In particular, we are interested in problems with the following characteristics:

- Large numbers of features, on the order of thousands.
- Many irrelevant features, with regard to a given task.
- Many redundant features, with regard to a given task.
- Noisy data, since image noist will affect all measured features.
- Continuous data, since most features listed above produce continuous values.
- Small training sets, relative to the size of the initial feature set.

## 2. Previous Work

The literature covering feature selection is extensive and spread across many fields, including document classification, data mining, object recognition, biometrics, remote sensing and computer vision. It is relevent to any task where the number of features or attributes is larger than the number of training samples, or too large to be computationally feasible. Feature selection is also related to four other areas of research: dimensionality reduction [24]; space partitioning [17]; feature extraction and decision trees [21].

Many algorithms have been proposed for feature selection, from simple algorithms like Sequential Forward Selection (SFS) [18] to more complex algorithms such as neural net prunning [6] and genetic selection [22]. Surveys of feature selection algorithms are given by Kittler [12], Siedlecki and Sklansky [22] and Bins [1]. For this work, the most relevant algorithms are: Relief, proposed by Kira and Rendell [11] in 1992 and extended by Kononenko [15] to handle noisy, incomplete and multi-class data sets; and Sequential Floating Forward Selection (SFFS) and Sequential Floating Backward Selection (SFBS), proposed in 1994 by Pudil et al. [20]. Relief has been shown to detect relevance well, even when features interact [5], and SFFS/SFBS has been shown to be much faster than Branch and Bound (BB) while obtaining comparable results on at least some data sets [20].

## 3 The proposed system

### 3.1 A Three-step Algorithm

The goal of our system is to reduce a large set of features (on the order of thousands) to a small subset of features (on the order of tens), without significantly reducing the system's ability to approximate functions or classify objects. Our approach, shown in Figure 1, is a three step system: first the irrelevant features are removed, then the redundant features are removed, and finally a traditional combinatorial feature selection algorithm is applied to the remaining features. The idea is that each step is a filter that reduces the number of candidate features, until finally only a small subset remains. As will be discussed later, we combine these filters into three algorithms (R+K+B, R+K+F, and R+K), depending on the domain and task statement.

### 3.2 The Component Filters

The first filter removes irrelevant features using a modified form of the Relief algorithm [11, 15]. For readers who are not familiar with Relief, it assigns relevance values to features by treating training samples as points in feature space. For each sample, it finds the nearest “hit” (another sample of the same class) and “miss” (a sample of a different class), and adjusts the relevance value of each feature according to the square of the feature difference between the sample and the hit and miss. Kononenko [15] suggests several modifications to Relief to generalize it for continuous features and to make it more robust in the presence of noise. We adopt Kononenko's modifications, and modify Relief again to remove a bias against non-monotonic features, as described in [1].

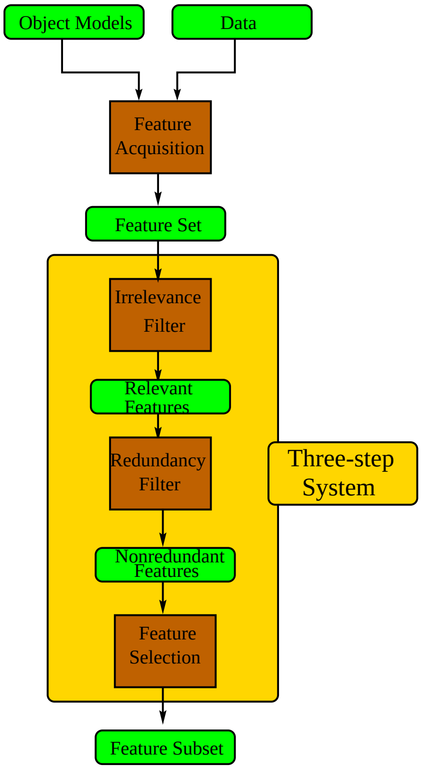

**Figure 1. High level system architecture.**

Within our feature selection system, Relief is used as a relevance filter. We therefore threshold the relevance values to divide the feature set into relevant and irrelevant features. This can be done either by thresholding the relevance value directly, or by selecting the highest $n$ values and discarding the remaining features. In either case, Relief does not detect redundancy, so the remaining feature set still contains redundant features.

The second step is a redundancy filter that uses the K-means algorithm [16] to cluster features according to how well they correlate to each other. When feature clusters are discovered, only the feature with the highest Relief score is kept; the other features in the cluster are removed from the feature set. This is an unusual application of K-means clustering, in that features are clustered (instead of samples), and correlation is used as the distance measure. A correlation threshold of 0.97 is used to detect when the features in a cluster are not sufficiently similar, in which case the cluster is split to make sure that potentially usefull features are not removed from the feature set.

The third and final filter is a combinatorial feature selection algorithm. When possible, we use the Sequential Floating Backward Selection (SFBS) algorithm [20]. Unfortunately, we find that SFBS is not feasible for feature sets of more than about 110 features. (Ng, et al, have previously reported a limit of 100 features [19].) Therefore, when the number of features remaining after the relevence and redundancy filters exceeds 110, we switch to the slightly less effective but more efficient Sequential Floating Forward Selection (SFFS) algorithm [20]. We use a Mahalanobis distance measure inside both SFFS and SFBS.

The order of these three filters is important. The asymptotic complexity of Relief is $O(s^2f)$, where $s$ is the number of training samples, and $f$ is the number of features. Since the number of features is much larger than the number of samples, the complexity of Relief is linear in terms of the most significant size factor. K-means, on the other hand, does more work per feature. The complexity of one iteration of K-means is $O(ksf)$, where $k$ is the number of clusters. Unfortunately, since we recursively split clusters using a very tight correlation threshold, $k$ is not a constant, but instead $k \approx f$. Thus the complexity of one iteration of K-means is $O(sf^2)$. (The number of iterations also increases with the number of features.) Since $f$ can be very large, it is important to filter irrelevent features before filtering redundant ones. Of course, SFFS and SFBS are more complex then either of the first two filters, and must be run last.

### 3.3 Running the system

There are two ways for an application to use our feature selection system. If the application does not require a fixed feature set size, but simply needs to ensure that there are no irrelevent or redundant features, then only the first two stages are run. We call this version R+K, and users need only to provide one parameter, to threshold the relief scores. (We have not encountered any reason to adjust the clustering threshold of 0.97.)

Alternatively, if an application requires a target number of features, then a third filter is used to reduce the set of relevent and non-redundant features to the target feature set size. If fewer than 110 features survive the first two filters, SFBS is used (we call this version R+K+B). Otherwise, the final filter is SFFS (creating R+K+F).

## 4. Evaluating the Feature Selection System

To evaluate our system, we conducted three experiments over three different data sets. The first data set contains regions of interest (i.e. subimages) extracted from aerial images of Fort Hood, Texas. Each region of interest is a hypothesized house, and the goal of the system is to judge which hypotheses are accurate. This is a regression task, since the accuracy of hypotheses are measured on a scale of zero to one. The second data set are features extracted from images of hand-written characters. This data set was first described by Breuklen et al. [3, 4], and has also been used by Jain et al. [10]. It has the disadvantage that it contains only 649 features[1], and that none of these features are completely irrelevent or redundant. The third data set contains images of cats and dogs, and has been previously used as a testbed to compare appearance-based recognition methods [25].

### 4.1. Experiment #1: aerial images

The first data set consist of 891 features computed over regions of interest extracted from aerial images of Fort Hood [7]. Typical features here are statistical features (mean, st. dev., etc.) computed over the raw image or first or second derivatives of the raw image, and repeated at different scales. Other features include eigenprojections, probes, histograms comparisons, etc. These features match the assumptions underlying our system, in that there are more features (891) then training samples (200), and many of the features are irrelevant for judging house hypotheses, or are redundant with other features.

The first experiment tests if all three steps of the algorithm are necessary. Since the system is composed of filters, any filter can be removed and the system will still work[2]. To test whether all three filter are needed, all feasible combinations of modules of the system were applied to the task of selecting the best subset of ten features.

Each combination of modules selects features in a different way, and consequently the number of features selected at each step may vary. Where possible, the Relief threshold was set to select the 300 most relevent features. By default, the clustering threshold was 0.97; however, when clustering was followed by SFBS, only the best 110 or fewer clusters according to their Relief score were used, and when K-means was the final filtering stage, only ten clusters were used. SFFS and SFBS were set to select exactly 10 features. Table 1 shows the eight feasible combinations of features and how many features were selected by each module.

For each feature set, 100 neural nets were trained and tested. When training a net, the data is randomly divided into three sets: a learning set (70%); a validation set (15%) and a test set (15%). Each net was trained for $15,000$ epochs, with an “early termination” routine saving the net state with the lowest MSE on the validation set. Each net is then evaluated in terms of its MSE on the test set. The average MSE for the 100 neural nets is shown in table 2 for each combination of filters. As can be seen, the best performance is achieved by using all three filters. The results are particularly good when the third filter is SFBS.

[1] The original images are no longer available, so we cannot compute additional features.

[2] SFBS is not practical if it is applied to more than 110 features.

| System | Original | R | K | F/B | Final set |
|---|---|---|---|---|---|
| R | 891 | 10 | — | — | 10 |
| F | 891 | — | — | 10 | 10 |
| R+K | 891 | 300 | 10 | — | 10 |
| R+F | 891 | 110 | — | 10 | 10 |
| R+B | 891 | 110 | — | 10 | 10 |
| K+F | 891 | — | — | 10 | 10 |
| R+K+F | 891 | 300 | 110 | 10 | 10 |
| R+K+B | 891 | 300 | 110 | 10 | 10 |

**Table 1. Number of features selected at each module of each filter combination tested. R, K, F, and B correspond to Relief, K-means, SFFS, and SFBS.**

| Average MSE for 100 nets (15000 epochs each) | |
|---|---|
| System | MSE |
| Relief | 0.0696 |
| SFFS | 0.0649 |
| R+K | 0.0269 |
| R+F | 0.0236 |
| R+B | 0.0247 |
| K+F | 0.0154 |
| R+K+F | 0.0156 |
| R+K+B | 0.0093 |

**Table 2. MSE for each tested combination of modules of the system.**

| System | | R | R+K | R+F | K+F | F |
|---|---|---|---|---|---|---|
| R+K+F | $H_0$ | F | F | F | T | F |
| | P | $\approx 0$ | 6.2e-15 | 1.5e-11 | 8.1e-01 | $\approx 0$ |

| System | | R+k+F | R+B | K+F |
|---|---|---|---|---|
| R+K+B | $H_0$ | F | F | F |
| | P | $\approx 0$ | $\approx 0$ | 2.2e-15 |

**Table 3. Result of the T-test ($H_0$) and probability of observing the given result by chance given that the null hypothesis is true (P) for pairs of filters. Values of $P \approx 0$ indicate that the probability is so small that it exceeds the capability of representation used by the statistical package.**

The differences between filter combinations are statistically relevant. Table 3 shows the result of a T-test comparing the combinations of filters. The null hypothesis ($H_O$) for the T-test is that the two samples have the same average MSE ($\alpha = 0.005$). The row of Table 3 labeled $H_0$ shows the result of the T-tests (true or false); the row labeled $P$ shows the probability of observing the given result by chance if the null hypothesis is true.

The first line of Table 3 compares the whole system using SFFS against other combinations using SFFS (or no combinatorial filter). This shows that it is significantly better to run the whole system than just part of the system. The only case where the difference is not statistically significant is in comparison to K+F. Even here, however, the three-step version outperformed the two-step version, although not by enough to rule out the possibility of a statistical fluke. More importantly, when the whole system was run using SFBS as the final filter, its performance greatly surpasses all other options. This indicates that when most features are irrelevant and/or redundant, the best option is to run the three steps of the system, i.e. Relief, K-means and SFBS.

Although the goal of this test was to check the necessity of every component, a second but equally important goal was achieved. Relief and SFFS, running alone, are two of the best feature selection algorithms in the literature. Thus, by showing that our system outperforms these two filters, we also show that our three-step algorithm outperforms its primary competitors.

### 4.2 Experiment #2: The Digits Data

The second data set contains features extracted from images of handwritten numerals ('0' - '9') extracted from a collection of Dutch utility maps. There are 200 samples per digit, for a total of 2,000 samples. Each sample has 649 features: 76 Fourier coefficients of the character shapes; 216 profile correlations; 64 Karhunen-Loève coefficients; 240 pixel averages over 2 x 3 windows; 47 Zernike moments; and 6 morphological features.

Unlike the aerial image data set, this data set does not obey the assumptions that motivated our system design. In particular, it has more training samples (2,000) than features (649), which makes it possible to train classifiers on the whole feature set, without performing feature selection. It also has no irrelevent features, and no two features are redundant.

The goal of the first part of this experiment was to test a "do no harm" philosophy. The digit data set is small enough to run SFFS on all 649 features. Since our three-step system uses SFFS, we wanted to be sure that the three-step algorithm did as well as SFFS alone, even when the assumptions underlying the first two filters were not met. Table 4 shows the results. The full system (R+K+F) actually performed slightly better than SFFS alone.

The full system with SFBS performed worse than SFFS alone, because we had to raise the Relief threshold artifi-

| System | # Feat | MSE |
|---|---|---|
| R+K+B | 649 | 0.0132 |
| R+K+F | 649 | 0.0049 |
| F | 649 | 0.0059 |

**Table 4. Average MSE for 100 nets (5000 epochs each) for combinations of the system on the original Digits data. R, K, F, and B correspond to Relief, K-means, SFFS, and SFBS.**

cially high in order to reduce the number of features prior to SFBS below 110. This is a warning: although Relief is good at detecting irrelevent features, it should not be relied on to select the best from among relevent features. The combinatorial selection algorithms are better at that. If there are more than 110 relevent and non-redundant features, use SFFS.

Having shown that our three-step algorithm does no harm, we next show that it filters redundant and irrelevent features when we know they are present. We modify the digits data set by creating a redundant feature set out of the 649 original digits features, plus a second copy of each of the original features with 10% Gaussian noise added, for a total of 1298 features. We then extended this to a redundant and irrelevent feature set by adding a third copy of each original feature, this time with 60% Gaussian noise added (rendering the new features almost useless). Finally, we created a mixed feature set out of the original 649 features plus four versions of each original feature with added noise ranging from 10% to 55%. The results of applying our three-step algorithm to each of these data sets is shown on Table 5.

| System | Feat. Set | # Feat | MSE |
|---|---|---|---|
| R+K+F | Redundant | 1298 | 0.0045 |
| R+K+F | Relevant | 1946 | 0.0046 |
| R+K+F | Mixed | 3245 | 0.0048 |
| F | Mixed | 3245 | 0.0086 |

**Table 5. Average MSE of 100 nets on the modified Digits data sets. R, K, F, and B correspond respectively to Relief, K-means, SFFS, and SFBS.**

Results from Table 5 show that the system was able to remove the irrelevant and redundant features, because its performance is statistically the same as the results for the original data set (see Table 4). In comparison, the performance of SFFS degrades when enough irrelevant and redundant features are included.

### 4.3 Experiment #3: Cats and Dogs

In the third experiment, we use the system to remove irrelevent and redundant features without specifying a target feature set size. As a result, the third filter (SFFS or SFBS) is discarded. Instead, we apply a “R+K” algorithm to remove irrelevent features and redundant features, and return as many relevent and unique features as the data will support.

In this experiment, the data set is composed of two hundred images of cat faces and dog faces. Each sample is a black-and-white 64x64 pixel image, and the images have been registered by aligning the eyes. The performance task is to distinguish cats from dogs. The standard technique, as described in [25], is to compute eigenvectors from a gallery of 160 out of the 200 images. The remaining 40 probe images are projected into the eigenspace, and the nearest gallery image to each is returned as its match. A match is said to be correct if it of the same species as the probe. Figure 2 shows 24 examples extracted from the cats and dogs data set.

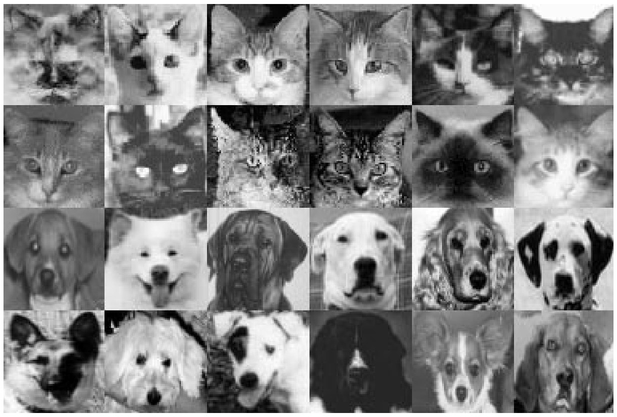

**Figure 2. Examples of images on the Cats and Dogs data set (12 cats and 12 dogs).**

Once again, the data contradicts one of our original assumptions, in this case the implicit assumption that features are expensive to compute and store. For principal components analysis, the features are the 4096 pixel values. Since these features do not have to be computed, and PCA can easily handle thousands of features, it is not necesssary to do feature selection for this task.

Nonetheless, there are two good reasons to select features from this data set. The first is provide an intuition about what pixels are important for this data set and task. The second is to allow PCA to be applied to much larger images. Current systems would be hard pressed to compute basis vectors for 512x512 images, but it would be much more feasible to select a few thousand pixels from such images and apply PCA to the reduced feature vectors.

This experiment also emphasizes the system's ability to detect relationships between features. The cats in the data set run the gamut of intensities from black to white; so do the dogs. As a result, the intensity of a single pixel is almost meaningless. It is only the differences between pixels that distinguish cats from dogs. Fortunately, Relief is sensitive enough to relations among features that it is still able to detect relevance.

Since the features are pixels, the results of Relief and K-means can be seen visually for this domain. Figure 3 shows images representing the result of the two filters. In these images, a pixel is zero (black) if it was eliminated from the feature set by the filter. Otherwise, the relevence value computed by Relief is used as the pixel's intensity.

Figure 3(a) is the image after the relevance filter. The relevence filter removes all but 615 pixels, reducing the volume of data by 85%. The remaining pixels highlight the differences betwee dogs and cats: dogs have relatively higher foreheads then cats, as well as wider muzzles. Cats have ears in the upper corners of the images. Cats' eyes are also larger, in proportion to their spacing.

Figure 3(b) shows the result of the redundancy filter, which further reduces the data set down to just 217 pixels. What the redundancy filter finds is that in many cases, the information in a pixel is the same as the information in the neighboring pixel. As a result, it produces a checkerboard effect, in which a pixel is selected and its neighbors are rejected.

The question, of course, is how much information has been lost by reducing the feature set from 4096 pixels to just 217. To test this, we applied a standard "eigenfaces" recogniton system to the complete set of 4096 pixels, and then again to the reduced set of 217 pixels. In both cases, we trained the system on 160 images, holding forty images out for testing. We repeated this process five times in a cross-validation mode, so that every image was used as a test image exactly once.

The classifier used was a PCA-based 1-nearest neighbor. This classifier computes all eigenvectors for a learning set and the sample projections over this basis. When a new sample is presented its projection over all eigen-vectors is computed and the result of the classification is the label (class) of the closest neighbor, in terms of Euclidean distance, to the sample projection.

To test if both data sets achieve the same performance, we used McNemar's test. This test is similar to a binomial test, but it considers only the difference in classification between the data sets. As a result, it is more sensitive to differences between algorithms than a binomial test. McNemar's test indicates that the difference in classification accuracy between the original pixel set and the reduced pixel set is not statistically significant. In other words, the data sets have approximately the same information despite the fact that the filtered version contains only 5% of the pixels in the original one.

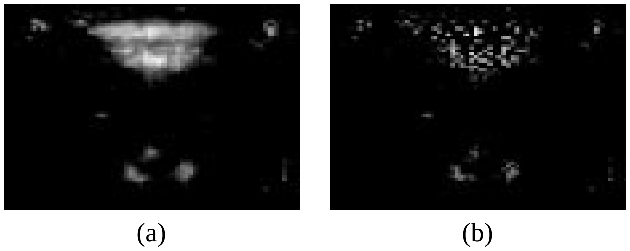

**Figure 3. Result of the Relief and Kmeans filters over the Cats and Dogs dataset. The value of each pixel is zero if it passed the filter or the Relief score it it did not passed the filter. a) after Relief filter (615 features/pixels selected); b) after k-means filter (217 features/pixels selected).**

## 5. Summary and Conclusions

The proposed system selects features from a large feature set via three filters. The first is based on the Relief algorithm ([11, 15]); it filters out irrelevant features. The second filter is based on K-means([16]); it filters out redundant features. The final filter is a combinatorial feature selection algorithm (SFFS or SFBS [20]); it selects the final subset of features. We recommend three algorithms, depending on the situation. The R+K+B algorihm is best for producing a feature set of a particular size, if the number of relevent and non-redundant features in the domain is less than 110. Otherwise, R+K+F is the best algorithm to select a feature subset of a specific size. If the size of the feature subset may vary, use R+K.

The system was tested with three different data sets. The Fort Hood aerial image hypothesis data set has many irrelevant and redundant features, and allowed us to show that R+K+B outperforms other algorithms from the literature, as well as other versions of this system. The digits data set [3, 10] is composed of hand-picked sets of features. It does not contain irrelevent or redundant features, and was used to show that the system "does no harm" when such features are not present. It also allowed us to demonstrate that the performance of the system does not degrade as irrelevent and redundant features are added. The third data set, composed of images of cats and dogs, allowed us to visualize the results of the first two filters, and to demonstrate that the system is sensitive to relationships between features.

The main contribution of this work is the system and the algorithms (R+K+B, R+K+F, R+K). As far as we know, these are the first algorithms (outside of text classification systems) that can handle such large data sets. It is also the first feature selection system which explicitly filters for irrelevance and redundancy.

## References

[1] J. Bins, *"Feature Selection of Huge Feature Sets in the Context of Computer Vision"*, Ph. D. Dissertation, Computer Science Department, Colorado State University, 2000.

[2] A. L. Blum and R. L. Rivest, *"Training a 3-node Neural Network is NP-complete"*, Neural Networks, 5:117-127, 1992.

[3] M. van Breukelen, R.P.W. Duin, D.M.J. Tax and J.E. den Hartog, *"Combining Classifiers for the Recognition of Handwritten digits"*, Kybernetika, 34(4):381-386, 1998.

[4] M. van Breukelen, R.P.W. Duin, D.M.J. Tax and J.E. den Hartog, *"Handwritten Digit Recognition by Combined Classifiers"*, $1^{st}$ IAPR Workshop on Statistical Techniques in Pattern Recognition, Prague, 1997, pp. 13-18.

[5] R. Caruana and D. Freitag, *"Greedy Attribute Selection"*, $11^{th}$ International Conference on Machine Learning, 1994.

[6] G. Castelano, A.M. Fanelli and M. Pelillo, *"An Iterative Pruning Algorithm for Feedforward Neural Networks"*, IEEE Transactions on Neural Networks, 8(3):519-531, 1997.

[7] B. A. Draper, J. Bins, K. Baek. *"ADORE: Adaptive Object Recognition"*, Videre 1(4):86-99, 2000.

[8] L. Hyafil and R. L. Rivest, *"Constructing Optimal Binary Decision Trees is NP-complete"*, Information Processing Letters, 5(1):15-17, 1976.

[9] A. Jain and B. Chandrasekaran, *"Dimensionality and Sample Size Considerations"*, Pattern Recognition Practice, V. 2, Krishnaiah and I.N. Kanal editors, pp. 835-855, North-Holland, 1982.

[10] A.K. Jain, R.P.W. Duin, and J. Mao, *"Statistical Pattern Recognition: A review"*, IEEE Transactions on Pattern Recognition and Machine Intelligence, 22(1), Jan. 2000.

[11] K. Kira and L.A. Rendell, *"The Feature Selection Problem: Traditional Methods and a New Algorithm"*, $10^{th}$ National Conference on Machine Intelligence, pp. 129-134, 1992.

[12] J. Kittler,*"Feature Set search Algorithms"*, Pattern Recognition and Signal Processing, C.H. Chen Editor, Sijthoff & Noordhoff, Alphen aan den Rijn, The Netherlands, 1978, pp. 41-60.

[13] R. Kohavi, *"Feature Subset Selection as Search with Probabilistic Estimates"*, AAAI Fall Symposium on Relevance, 1994.

[14] R. Kohavi and G.H. John, *"Wrappers for Feature Subset Selection"*, Artificial Intelligence Journal, 97(1-2):273-324, 1997.

[15] I. Kononenko, *"Estimation attributes: Analysis and Extensions of RELIEF"*, European Conference on Machine Learning, Catana, Italy, pp. 171-182, 1994.

[16] J. MacQueen, *"Some Methods for Classification and Analysys of Multivariate Observations"*, Proceedings of the Fifth Berkley Symposium on Mathematics, Statistics and Probability, L. M. LeCam and J. Neyman eds., pp. 281-297, University of California Press, Berkeley, 1967.

[17] D.P. Mandal, *"Partitioning of Feature Space For Pattern Classification* Pattern Recognition, 30(12):1971-1990, 1997.

[18] T. Marill and D.M. Green, *"On the Effectiveness of Receptors in Recognition Systems"*, IEEE transactions on Information Theory, 9:11-17, 1963.

[19] H.T. Ng, W.B. Goh and K.L. Low, *"Feature Selection, Perceptron Learning, and a Usability Case Study for text Categorization"*. 20th annual international ACM SIGIR conference on Research and development in information retrieval, July 27-31, Philadelphia, pp. 67-73, 1997.

[20] P. Pudil, J. Novovicova and J. Kittler, *"Floating Search Methods in Feature Selection"*, Pattern Recognition Letters, (15)1119-1125, 1994.

[21] J.R. Quinlan, *"Decision Trees and Multivalued Attributes"*, J. Richards, ed., Machine Intelligence, V. 11, Oxford, England, Oxford Univ. Press, pp. 305-318, 1988.

[22] W. Siedlecki and J. Sklansky, *"On Automatic Feature Selection"*, International Journal of Pattern Recognition and Artificial Intelligence, 2(2):197-220, 1988.

[23] K. Tieu and P. Viola. *"Boosting Image Retrieval"*, IEEE Conference on Computer Vision and Pattern Recognition, Hilton Head, June 13-15, 2000, pp. 228-235.

[24] M. Turk and A. Pentland, *"Eigenfaces for Recognition"*, Journal of Cognitive Neuroscience, 3(1), 1991.

[25] W. Yambor, *"Analysis of PCA-based and Fisher Discriminant-Based Image Recognition Algorithms"*, M.S. Thesis, Department of Computer Science, Colorado State University, May 2000.