ABSTRACT:
Selecting most representative features among a huge feature set is considered in this paper. One feature extraction and five feature selection algorithms are tested on features obtained form steel surfaces. Two new feature selection algorithms are introduced in this paper. Performance measure for feature selection algorithms is also introduced as a new method to select the best performing feature selection algorithm. Steel surfaces are classified as rusty and clean according to selected features by each selection algorithm.

Keywords: Steel surfaces, rust grades of steel, feature selection and extraction, performance measure for feature selection algorithms.

1. INTRODUCTION
As the number of features to be fed into the classifier are beyond the permissible limits for exhaustive search, a preprocessing operation is needed to eliminate features having no discrimination power. Feature selection and extraction algorithms are needed at this point.

As the dimensions of the feature space increase beyond a certain limit, the problem ‘curse of dimensionality’ comes into play. Curse of dimensionality can be described as the situation where adding extra features to the classifier decreases the performance of the classifier. The limit for the dimension of the feature space depends on the classifier type and the nature of the problem.

Extracting each feature from the image requires additional work, which increases the both time requirements of the recognition operation and the complexity of the recognition system. In real time applications, the complexity issues are the most important ones, because the operation time and complexity of the system is directly proportional to the number of features used in the recognition system.

From the problems mentioned above, it is clear that feature selection methods are vital for pattern recognition problems.

One of the most important feature selection methods is Branch and Bound Algorithm introduced by Narendra and Fukunaga [1]. Kittler [2] studied several feature selection and extraction algorithms. Pudil et al. [3] used sequential forward floating and sequential backward floating selection methods for feature selection.

Another group of methods for feature dimension reduction is the projection algorithms. The projection algorithms map the data in the original feature space to a new feature space such that the classes are well-separated [4-6].

In this paper, one feature extraction and five feature selection algorithms are tested on feature sets formed in technical report-1 [7]. Feature extraction algorithms are considered in section two. Feature selection algorithms are considered in section three.
Performance measure for feature selection algorithms is given in section four. Implementation and results are given in section five.

2. FEATURE EXTRACTION

Feature extraction algorithms map the samples in the feature space to a new feature space. This mapping operation is done to satisfy certain conditions in the new feature space. One simple example for such a condition may be to increase the class separation in the new feature space. The main characteristics of the feature space to be mapped must also be preserved.

In addition to the mapping conditions to be satisfied, if the new feature space is lower dimensional than the initial feature space, then dimension reduction in the feature space is also achieved.

In this section, feature extraction algorithms are discussed for feature space dimension reduction while preserving the properties of the initial feature space.

An example for feature extraction operation is given in Figure 2.1


2.1 Principal Component Analysis (PCA)

PCA maps the original data set to a new space such that the maximum amount of variance of the original data set is preserved. If the mapped space is lower dimensional than the original space, then dimension reduction is achieved.

The coordinate axes of lower dimensional space are found as follows:

i- The covariance matrix of the data set is found.
ii- Eigenvalues of the covariance matrix are calculated and these eigenvalues are sorted from maximum to minimum.
iii- Eigenvectors corresponding to these eigenvalues are formed.
iv- New coordinate axes of the feature space are taken as the first ‘s’ eigenvectors. This operation is done to decrease the dimension and preserve maximum amount of representation energy.
v- Mapping operation is done to transform the samples to new space. New dimension for the feature space will be ‘s’, where $s \leq d$.

Since all the eigenvectors formed will be orthogonal to each other, the newly formed subspace will also be spanned by orthogonal vectors.

3. FEATURE SELECTION

Feature selection algorithms try to find the most representative features among a larger feature set. Unlike feature extraction algorithms considered in the previous section, the aim of these algorithms is only to select features that can be used for classification purposes.

Features are graded according to measures extracted from them. It is assumed that these measures give necessary information about the class distribution in the feature. According to the grades of features, the best ones are searched and selected.

An example of the operation of the feature selection operation is given in Figure 3.1

3.1 Entropy Measure

The entropy measure concerns with the information contained in the feature. To find a measure of information contained in the feature, its entropy is calculated. It is assumed that as the information contained in the feature is increased, the classes in that feature will be better discriminated.

To calculate the entropy measure of a feature, it is normalized such that the sum of the sample values in the feature will be unity and the values in the sample will be greater than zero.

Entropy of a feature \( ft \) having \( M \) samples can be calculated as:

\[
E = - \sum_{i=1}^{M} f^*(i) \log(f^*(i))
\]  

(1)

where \( f^*(i) = \frac{f(i) + \min(f)}{\sum_{j=1}^{M}(f(j) + \min(f))} \)

(2)

3.2 Selection by Shape Similarity

Shape of a feature can be taken as the distribution of the data samples, those are ordered in the same form for all of the features, in the feature. The aim of selection by shape similarity is to discard the features whose data samples resemble in distribution.
If two features have similar shapes, they have similar sample sets also. Discarding one of them from the feature set will not decrease the discriminative power of the feature set.

Similarity measure for two features is taken as the Euclidean distance between the smoothed and normalized versions of them. Smoothing is required to eliminate noise and add generality to the shapes of the features.

The similarity measure can be calculated as:

For two features, \( ft_1 \) and \( ft_2 \), having same index \( i \):

\[
\text{shape of } ft_1 \rightarrow ft'_1(i) = \frac{ft_1(i) + \min(ft_1)}{\sum_{i=1}^{M} (ft_1(i) + \min(ft_1))} \quad i = 1..M
\]

\[
\text{shape of } ft_2 \rightarrow ft'_2(i) = \frac{ft_2(i) + \min(ft_2)}{\sum_{i=1}^{M} (ft_2(i) + \min(ft_2))} \quad i = 1..M
\]

\( M \) is the total number of samples.

\[
\text{distance} = \sum_{i=1}^{M} |ft'_1(i) - ft'_2(i)|
\]  

To give an example for the similarity measure, two features extracted from GLCM [7] are given in Figures 3.2.1 and 3.2.2. ‘x’ axis corresponds to index of feature. ‘y’ axis corresponds to values of samples in the feature.

Before calculating distance between these two features, Wavelet Transform of them are taken for three levels and low channel of the transform is taken. Since three levels of Wavelet Transform is applied, the sample size is decreased from 1644 to 205. The features are then normalized for obtaining their shape. In Figure 3.2.3 shapes of these features are given. The difference of these two features is given as a new feature \( ft_3 \) in Figure 3.2.4.
Similarity between these two features will be calculated as:

\[ distance = \sum_{i=1}^{205} |ft_3(i)| \]  

(6)

To find the resembling features, hierarchical clustering operation is done on the difference measures found between features [14]. The number of features to be selected is taken as the final cluster number in hierarchical clustering. This way feature number to be selected directly adjusts the similarity between features.

3.3 Variance Measure

In this measure, the variance of the feature is taken as a measure to give insight of the feature.

The variance of a distribution gives information about the spanning of the data around the mean value. In this measure the distribution of the feature is considered. As all of the data contained in the feature becomes less diversified, the class separation with that feature will be more difficult. As the variance of the feature is decreased, it can be assumed that the feature will be less powerful in classification since for different classes the feature values would be similar to one another.

This assumption can easily be misleading in many cases. As the variance of the feature is increased, this does not indicate that the feature will be more powerful in classification. The feature can be formed such that, the data belonging to same class will have large variance.

To overcome this drawback of the variance measure, the class information must be taken into account. This issue is explored in the next section.
3.4 Fisher’s Measure

Linear Discriminant Analysis (LDA) is one of the projection algorithms. The projection is achieved to increase the class separation in the new space [14]. To find such a space, iterative methods are used to increase the measure introduced by Fisher:

\[ d = \frac{\text{between class scatter}}{\sum \text{within class scatter}} \]  

(7)

Fisher’s measure has two constraints to be satisfied:

i- The distance between class centers must be maximized.

ii- The scattering of the data belonging to the same class must be minimized.

In this section, Fisher’s measure is directly used for feature selection without projecting the data to a new space. For each feature, Fisher’s measure indicates the distribution of the data in the feature, taking class information into account.

For a two-class case, Fisher’s measure can be given as

\[ d = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \]  

(8)

3.5 Fisher’s Measure with Clustering

Fisher’s measure is related with the data distribution for each class and class separation. This measure is suitable for linearly separable classes. This measure will be useless if classes are not linearly separable.

In this section, a new measure is introduced related to the Fisher’s measure. The classes need not to be linearly separable in this measure. Non parametric classifiers can be used in this kind of class distributions.

Instead of calculating class centers, clustering algorithms are applied to find class centers in this newly suggested method. If the cluster centers to be found by clustering algorithms are taken above the total classes number, then linearly non-separable cases can be handled. If the classes are linearly separable then the cluster centers will converge to the same locations. Linearly separable cases will not cause any problem if the cluster numbers are taken above class numbers.

First of all desired points are introduced to calculate Fisher’s measure with clustering. Desired points are the ideal locations of cluster centers due to the range of data in feature space. Range of data corresponds to the distance between minimum and maximum points in all axes of the feature space. Desired points are arranged such that, all of the cluster centers are separated maximally in the feature space.

The measure is related to the distance between cluster centers found by clustering algorithms and desired points in the feature space. In selecting features, the features that have smallest distances are chosen.

In addition to the cost function of Fisher’s measure, the range of data is taken into account as a normalizing factor in this method.

Fisher’s Measure with Clustering for \( n \) cluster centers is:

\[ d = \frac{\sum_{i=1}^{n} \text{dist(cluster center}_i - \text{desired point}_i)}{\text{range of data}} \]  

(9)
To find the cluster centers in the data, two clustering algorithms K-Means Clustering and Self Organizing Map (SOM) are applied. The difference between SOM and K-Means Clustering is the cluster centers found.

### 3.5.1 Fisher’s Measure with K-Means Clustering (FKMC)

As the name implies, in K-Means Clustering only mean value of the cluster is assumed to be unknown. K-Means Clustering algorithm tries to find the cluster center from the data set given [14].

To give an example of Fisher’s Measure with K-Means Clustering: Two normally distributed data with mean values (-2, -2) and (2,2) are given in Figure 3.5.1.1 to represent two classes. Solid circles represent the cluster centers found by K Means Clustering method. Rectangles represent the desired points for the cluster centers in the feature space.

![Figure 3.5.1.1 Example for Fisher’s Measure with K-Means Clustering](image)

For this example Fisher’s Measure with Clustering can be calculated as:

\[
d = \frac{\sum_{i=1}^{2} dc_i}{\sqrt{(rx^2 + ry^2)}}
\]  

(10)

dc is the distance between cluster center found and the desired point in feature space.
rx is the range of data in x axis.
ry is the range of data in y axis.

Another example for Fisher’s Measure with K-Means Clustering is given in Figure 3.5.1.2. To give an example in two dimensions, three normally distributed data with mean values (-6, -6), (6,6) and (0,0) are given to represent two classes.
For this example Fisher’s Measure with Clustering can be calculated as:

\[ d = \frac{\sum_{i=1}^{3} dc_i}{\sqrt{(rx^2 + ry^2)}} \]  

(11)

3.5.2 Fisher’s Measure with Self Organizing Map (FSOM)

Self Organizing Map (SOM) is an unsupervised Artificial Neural Networks (ANN) algorithm that models the distribution of the data. Kohonen [15-17] proposed SOM.

SOM has a learning rule:

\[ w_i(t+1) = w_i(t) + \alpha(t)[x(t) - w_i(t)] \quad \text{if} \quad i \in N_c(t) \]

\[ w_i(t+1) = w_i(t) \quad \text{otherwise} \]  

(12)

\[ N_c(t) \] is the neighborhood range at time \( t \).

\( w_i \) is the weight of neurons, \( i \) is the index of neurons in \( N_c \).

\( \alpha(t) \) is the adaptation gain.

Weight of each neuron is adjusted to model the distribution of the data according to the location of the neuron. Each training data is taken as the desired output of the system. The modeling error is fed back to adjust the weights of the neurons.

As stated in the previous paragraph, the neurons model the distribution of the data according to their locations. In addition to updating the weight of neurons, the weights of the neighboring neurons are also updated in SOM. The neighborhood is decreased with time. As the neighborhood is decreased with time, the network specializes.

The same data distribution used in the previous example given in Figure 3.5.1.1 is tested with SOM in the following example. Solid circles represent the cluster centers found by SOM. Rectangles represent the desired points for the cluster centers in the feature space.
For this example Fisher’s Measure with Clustering can be calculated as:

\[ d = \frac{\sum_{i=1}^{2} dc_i}{\sqrt{(rx^2 + ry^2)}} \]  

(13)

The same example given in Figure 3.5.1.2 is tested with SOM in the following figure.

For this example Fisher’s Measure with Clustering can be calculated as:

\[ d = \frac{\sum_{i=1}^{3} dc_i}{\sqrt{(rx^2 + ry^2)}} \]  

(14)

4. PERFORMANCE MEASURE FOR FEATURE SELECTION ALGORITHMS
Feature selection algorithms must be graded according to their performances on the given feature set. In this paper, the data obtained from the texture analysis methods are used. In addition to grading of feature selection algorithms, grading of texture analysis methods can also be obtained with the same method.

The idea behind the performance measure is as follows:

Each classifier has its estimation for the class distribution. The classification success of the classifier is directly related to this estimation. According to this estimation difference between different classifiers, they have different decisions to the same sample set.

If the features are tested on as many classifiers as possible, the total tendency of the classifiers will give information on the discrimination power of the feature set. The total tendency of the classifiers will not be effected by the different estimations of different classifiers included.

Instead of testing as many classifiers as possible, two classifiers having totally different estimations about the class distributions are taken to simplify the calculations in this study: Minimum Distance Classifier (MDC) and KNN Classifier [14].

Minimum Distance Classifier (MDC) is the simplest case of Bayes classifier. Prior class distribution of each class is taken as Gaussian distribution. On the other hand, KNN has no prior assumption for the class distribution. KNN tries to estimate the class distribution directly from the training data set. They represent two different approaches in statistical pattern recognition theory.

If both classifiers decide in a similar way for the feature set given, it can be concluded that the effect of estimation of prior class distributions will be minimized.

If the feature sets selected by the feature selection algorithms are graded according to their classification successes, the performance measure of the feature selection algorithm can be obtained.

Performance measure is calculated as follows:

i- Minimum Distance Classifier and KNN classifier classify the feature set.

ii- To eliminate prior estimation effects on the discrimination power of the feature set, mean value and the variance of the classification success of two classifiers are calculated. Mean value of the classification successes represents the common classification success. Variance of the classification successes represents the degree of agreement of the two classifiers on the classification decision.

iii- Both mean and variance of the classification successes are weighted and summed up to find the performance of the feature set.

iv- Total performance value of the feature selection algorithm on the given feature sets is taken as the average performance of all feature sets.

The performance measure of a feature selection algorithm for \( n \) feature sets can be calculated as:

Let 
\[
C_m(f_i) \quad \text{represent the percent success of the Minimum Distance Classifier on the feature } f_i, \text{ that is selected by feature selection algorithm.}
\]
\[
C_k(f_i) \quad \text{represent the percent success of the KNN Classifier on the feature } f_i, \text{ that is selected by feature selection algorithm.}
\]

Mean value of the successes will be calculated as:

\[
\mu_i = \frac{C_m(f_i) + C_k(f_i)}{2} \quad (15)
\]
Variance of the successes will be calculated as:

\[ \sigma_i^2 = (C_m(f_i) - \mu_i)^2 + (C_k(f_i) - \mu_i)^2 \]  

(16)

If the feature selection algorithm is tested with \( N \) feature sets, performance measure for that feature selection algorithm will be calculated as:

\[ \text{Performance} = \frac{1}{N} \sum_{i=1}^{N} (w_i \ast \mu_i + w_2 \ast \sigma_i^2) \]  

(17)

5. IMPLEMENTATION AND RESULTS

In technical report 1, eight texture analysis methods are used to form features from the images of steel surfaces [7]. Texture analysis methods applied to steel surfaces and the features formed for each method are given in Table 5.1. As the number of extracted features are beyond the limits of exhaustive search for classification; feature selection methods are applied on the feature set to decrease the dimension of the feature space in this paper.

<table>
<thead>
<tr>
<th>Abbr.</th>
<th>Texture Analysis Method</th>
<th># Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFT</td>
<td>Fourier Transform</td>
<td>20</td>
</tr>
<tr>
<td>G T</td>
<td>Gabor Transform</td>
<td>120</td>
</tr>
<tr>
<td>GLCM</td>
<td>Gray Level Cooccurrence Matrices</td>
<td>30</td>
</tr>
<tr>
<td>HIST</td>
<td>Histogram Information</td>
<td>20</td>
</tr>
<tr>
<td>MRF</td>
<td>Markov Random Fields</td>
<td>125</td>
</tr>
<tr>
<td>R T</td>
<td>Radon Transform</td>
<td>150</td>
</tr>
<tr>
<td>S D</td>
<td>Surface Density Approach</td>
<td>240</td>
</tr>
<tr>
<td>W T</td>
<td>Wavelet Transform</td>
<td>1408</td>
</tr>
</tbody>
</table>

In this section two main sets of experiments are achieved:

i- The first experiment concentrates on the selected features by different selection algorithms.

ii- The second experiment deals with the performance of feature selection algorithms on different data sets.

5.1 Results of Feature Selection Algorithms

The feature selection and extraction algorithms explained in the previous sections are tested on the feature sets obtained by texture analysis methods. Each method is tested on the feature sets of eight texture analysis methods separately. The layout of the procedure applied in this section is given in Figure 5.1.1
In searching for the best features, the combinations of the features are not considered. The search time was the main limitation for such a search. Hence each feature is considered separately without its combinations with other features.

To force the same search criteria for all of the selection algorithms, the same number of features are selected by each method. Since the selection algorithms used do not consider the combination of features in searching, this enforcement can be accepted. To compare the features they select, such enforcement was also necessary.

The final number of features to be selected is taken as three. From the past experiments [18], selecting three features from each texture analysis method seems acceptable. The features selected will be used in classification algorithms. The curse of dimensionality can be encountered if the feature space is higher dimensional.

Feature selection is done in hierarchical manner except for PCA, since PCA requires compact feature space. The aim of hierarchical search was to decrease the search time in selection. The following procedure is applied in hierarchical selection:

i- For each texture analysis method, features formed from steel surfaces are grouped due to the color channels they are obtained from. Feature selection algorithms select four features from this group for each texture analysis method.

ii- The features selected from different color channels of the texture analysis methods are collected into one group and three features are selected from this group as the final representative of the texture analysis method.

iii- The final three features selected are taken as the result of selection operation from that texture analysis method.

To give an example for hierarchical operation: For the features extracted from Markov Random Fields approach, Figure 5.1.2 shows layout of the example.

There are 125 features extracted from the image set. At the first selection, the number of features is decreased to 20. At the second level, the total number of features is reduced to 3. The final number of features selected from MRF is 3.
In Fisher’s Measure with Clustering six cluster centers are assumed for two-class case. For the cases where the part of one class is located in the region of the second class is also considered.

In Shape Similarity Method, to find the smoothed shape of the feature, Wavelet Transform is applied, for three resolution levels and the low channel of the transform is taken as the smooth version of the shape of the feature. Daubechies low pass filter tap 32 is used in Wavelet Transform.

The features selected by each method are tested on Minimum Distance Classifier and KNN classifier. Half of the sample set, that has 1644 samples, is used for training classifiers and the other half of the sample set is used for testing the classifiers.

In this section, only two classes are taken for classification. Two classes are taken as rusty and clean. Correct classification rates are given in the tables below in hundred percent forms.

### Table 5.1.1. Classification successes of features selected by the Entropy Measure

<table>
<thead>
<tr>
<th></th>
<th>FT</th>
<th>GT</th>
<th>GLCM</th>
<th>HIST</th>
<th>MRF</th>
<th>RT</th>
<th>SD</th>
<th>WT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDC</td>
<td>87.83%</td>
<td>88.2%</td>
<td>94.77%</td>
<td>82.48%</td>
<td>83.09%</td>
<td>51.34%</td>
<td>75.18%</td>
<td>61.44%</td>
</tr>
<tr>
<td>KNN</td>
<td>98.83%</td>
<td>97.8%</td>
<td>99.2%</td>
<td>94.65%</td>
<td>98.18%</td>
<td>93.43%</td>
<td>91.24%</td>
<td>76.16%</td>
</tr>
</tbody>
</table>

### Table 5.1.2. Classification successes of features selected by the Fisher’s Measure

<table>
<thead>
<tr>
<th></th>
<th>FT</th>
<th>GT</th>
<th>GLCM</th>
<th>HIST</th>
<th>MRF</th>
<th>RT</th>
<th>SD</th>
<th>WT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDC</td>
<td>88.2%</td>
<td>88.32%</td>
<td>99.64%</td>
<td>84.06%</td>
<td>73.84%</td>
<td>91.85%</td>
<td>90.27%</td>
<td>61.8%</td>
</tr>
<tr>
<td>KNN</td>
<td>95.01%</td>
<td>95.26%</td>
<td>99.64%</td>
<td>98.66%</td>
<td>99.76%</td>
<td>93.92%</td>
<td>97.2%</td>
<td>90.02%</td>
</tr>
</tbody>
</table>

### Table 5.1.3. Classification successes of features selected by the FKMC

<table>
<thead>
<tr>
<th></th>
<th>FT</th>
<th>GT</th>
<th>GLCM</th>
<th>HIST</th>
<th>MRF</th>
<th>RT</th>
<th>SD</th>
<th>WT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDC</td>
<td>91.73%</td>
<td>66.06%</td>
<td>96.72%</td>
<td>88.44%</td>
<td>99.27%</td>
<td>55.11%</td>
<td>72.14%</td>
<td></td>
</tr>
<tr>
<td>KNN</td>
<td>97.45%</td>
<td>91.97%</td>
<td>98.78%</td>
<td>98.91%</td>
<td>99.64%</td>
<td>86.25%</td>
<td>91.73%</td>
<td></td>
</tr>
</tbody>
</table>

### Table 5.1.4. Classification successes of features selected by the FSOM

<table>
<thead>
<tr>
<th></th>
<th>FT</th>
<th>GT</th>
<th>GLCM</th>
<th>HIST</th>
<th>MRF</th>
<th>RT</th>
<th>SD</th>
<th>WT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDC</td>
<td>88.2%</td>
<td>88.93%</td>
<td>88.69%</td>
<td>77.49%</td>
<td>98.78%</td>
<td>91.12%</td>
<td>86.25%</td>
<td></td>
</tr>
<tr>
<td>KNN</td>
<td>95.01%</td>
<td>98.26%</td>
<td>94.77%</td>
<td>91.61%</td>
<td>99.64%</td>
<td>97.81%</td>
<td>94.77%</td>
<td></td>
</tr>
</tbody>
</table>

### Table 5.1.5. Classification successes of features extracted by PCA

<table>
<thead>
<tr>
<th></th>
<th>FT</th>
<th>GT</th>
<th>GLCM</th>
<th>HIST</th>
<th>MRF</th>
<th>RT</th>
<th>SD</th>
<th>WT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDC</td>
<td>67.64%</td>
<td>87.59%</td>
<td>93.19%</td>
<td>83.21%</td>
<td>99.39%</td>
<td>51.34%</td>
<td>42.58%</td>
<td>89.66%</td>
</tr>
<tr>
<td>KNN</td>
<td>98.66%</td>
<td>98.42%</td>
<td>99.76%</td>
<td>94.53%</td>
<td>99.51%</td>
<td>81.39%</td>
<td>86.13%</td>
<td>98.78%</td>
</tr>
</tbody>
</table>

### Table 5.1.6. Classification successes of features selected by the Shape Similarity

<table>
<thead>
<tr>
<th></th>
<th>FT</th>
<th>GT</th>
<th>GLCM</th>
<th>HIST</th>
<th>MRF</th>
<th>RT</th>
<th>SD</th>
<th>WT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDC</td>
<td>91.73%</td>
<td>88.56%</td>
<td>94.28%</td>
<td>86.25%</td>
<td>70.19%</td>
<td>51.34%</td>
<td>68.61%</td>
<td>63.26%</td>
</tr>
</tbody>
</table>
Table 5.1.7. Classification successes of features selected by the Variance Measure.

<table>
<thead>
<tr>
<th></th>
<th>FT</th>
<th>GT</th>
<th>GLCM</th>
<th>HIST</th>
<th>MRF</th>
<th>RT</th>
<th>SD</th>
<th>WT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDC</td>
<td>65.33%</td>
<td>65.57%</td>
<td>89.05%</td>
<td>83.33%</td>
<td>84.55%</td>
<td>51.34%</td>
<td>41.73%</td>
<td>86.62%</td>
</tr>
<tr>
<td>KNN</td>
<td>97.32%</td>
<td>93.8%</td>
<td>98.66%</td>
<td>92.82%</td>
<td>97.57%</td>
<td>85.65%</td>
<td>82.85%</td>
<td>98.78%</td>
</tr>
</tbody>
</table>

5.2 Results of the Performance Measure for Feature Selection Algorithms

In this section, performance measure of each feature selection algorithm is calculated.

In addition to calculating the performance measure of feature selection algorithms, the same idea is applied to texture analysis methods and their performances are also calculated.

To compare the results of the selection algorithms, three different sets of weights and thresholds are applied in performance measure. Applied weight sets and thresholds are given in Table 5.2.1.

Table 5.2.1 Weight Sets

<table>
<thead>
<tr>
<th></th>
<th>Weight₁</th>
<th>Weight₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set1</td>
<td>1</td>
<td>-0.08</td>
</tr>
<tr>
<td>Set2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Set3</td>
<td>1.2</td>
<td>-0.2</td>
</tr>
</tbody>
</table>

The features formed by Wavelet Transform are not tested on Fisher’s Measure with Clustering because of time requirements. The corresponding cells are left blank. In calculating the percent performances, these cells are not taken into calculations.

Performance measures that are calculated for three different weight sets are given tables below. For each weight set shaded cells correspond to features that performed better than the threshold value given for that weight set.

Table 5.2.2 Performance Results of weight set1

<table>
<thead>
<tr>
<th></th>
<th>Entropy</th>
<th>Fisher M.</th>
<th>FKMC</th>
<th>FSOM</th>
<th>PCA</th>
<th>Shape S.</th>
<th>Variance</th>
<th>Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT</td>
<td>90.91</td>
<td>90.68</td>
<td>93.94</td>
<td>90.68</td>
<td>63.91</td>
<td>93.94</td>
<td>60.86</td>
<td>83.56</td>
</tr>
<tr>
<td>GT</td>
<td>91.16</td>
<td>89.57</td>
<td>65.59</td>
<td>91.85</td>
<td>90.66</td>
<td>91.4</td>
<td>63.75</td>
<td>83.43</td>
</tr>
<tr>
<td>GLCM</td>
<td>96.6</td>
<td>98.93</td>
<td>97.67</td>
<td>89.98</td>
<td>95.61</td>
<td>96.24</td>
<td>92.01</td>
<td>95.29</td>
</tr>
<tr>
<td>HIST</td>
<td>85.57</td>
<td>87.01</td>
<td>91.48</td>
<td>77.49</td>
<td>86.31</td>
<td>88.81</td>
<td>86.27</td>
<td>86.13</td>
</tr>
<tr>
<td>MRF</td>
<td>86.09</td>
<td>73.36</td>
<td>99.45</td>
<td>99.2</td>
<td>99.45</td>
<td>68.44</td>
<td>87.67</td>
<td>87.66</td>
</tr>
<tr>
<td>RT</td>
<td>36.95</td>
<td>92.8</td>
<td>51.27</td>
<td>93.57</td>
<td>48.31</td>
<td>37.95</td>
<td>44.95</td>
<td>57.97</td>
</tr>
<tr>
<td>SD</td>
<td>78.05</td>
<td>92.77</td>
<td>74.26</td>
<td>89.06</td>
<td>26.42</td>
<td>70.44</td>
<td>28.47</td>
<td>65.64</td>
</tr>
<tr>
<td>WT</td>
<td>64.47</td>
<td>59.98</td>
<td>92.56</td>
<td>80.08</td>
<td>89.74</td>
<td>77.37</td>
<td>69.22</td>
<td></td>
</tr>
<tr>
<td>Perf.</td>
<td>78.73</td>
<td>85.63</td>
<td>81.95</td>
<td>90.26</td>
<td>75.4</td>
<td>78.41</td>
<td>69.22</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2.3 Performance Results of weight set2

<table>
<thead>
<tr>
<th></th>
<th>Entropy</th>
<th>Fisher M.</th>
<th>FKMC</th>
<th>FSOM</th>
<th>PCA</th>
<th>Shape S.</th>
<th>Variance</th>
<th>Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT</td>
<td>93.33</td>
<td>91.61</td>
<td>94.59</td>
<td>91.61</td>
<td>83.15</td>
<td>94.59</td>
<td>81.33</td>
<td>90.03</td>
</tr>
<tr>
<td>GT</td>
<td>92.22</td>
<td>91.79</td>
<td>79.02</td>
<td>92.1</td>
<td>93.01</td>
<td>92.94</td>
<td>79.69</td>
<td>88.68</td>
</tr>
<tr>
<td>GLCM</td>
<td>96.99</td>
<td>99.64</td>
<td>97.75</td>
<td>91.73</td>
<td>96.48</td>
<td>96.72</td>
<td>93.86</td>
<td>96.17</td>
</tr>
</tbody>
</table>
To compare the feature selection algorithms, their performances are given in hundred percent forms in Table 5.2.5.

<table>
<thead>
<tr>
<th>Set</th>
<th>Entropy</th>
<th>Fisher M</th>
<th>FKMC</th>
<th>FSOM</th>
<th>PCA</th>
<th>Shape S</th>
<th>Variance</th>
<th>Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set1</td>
<td>78.73%</td>
<td>85.63%</td>
<td>81.95%</td>
<td>90.26%</td>
<td>75.4%</td>
<td>78.41%</td>
<td>69.22%</td>
<td></td>
</tr>
<tr>
<td>Set2</td>
<td>85.77%</td>
<td>90.72%</td>
<td>88.16%</td>
<td>92.02%</td>
<td>85.71%</td>
<td>86.83%</td>
<td>82.19%</td>
<td></td>
</tr>
<tr>
<td>Set3</td>
<td>81.14%</td>
<td>80.94%</td>
<td>75.23%</td>
<td>89.4%</td>
<td>67.93%</td>
<td>68.61%</td>
<td>57.6%</td>
<td></td>
</tr>
</tbody>
</table>

In addition to the performances of feature selection algorithms, performances of texture analysis methods are also calculated by the total times they are selected. The percent results are given in Table 5.2.6.

<table>
<thead>
<tr>
<th>Set</th>
<th>FT</th>
<th>GT</th>
<th>GLCM</th>
<th>HIST</th>
<th>MRF</th>
<th>RT</th>
<th>SD</th>
<th>WT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set1</td>
<td>83.56%</td>
<td>83.43%</td>
<td>95.29%</td>
<td>86.13%</td>
<td>87.66%</td>
<td>57.97%</td>
<td>65.64%</td>
<td>77.37%</td>
</tr>
<tr>
<td>Set2</td>
<td>90.03%</td>
<td>88.68%</td>
<td>96.17%</td>
<td>89.28%</td>
<td>92.94%</td>
<td>77.04%</td>
<td>79.7%</td>
<td>82.82%</td>
</tr>
<tr>
<td>Set3</td>
<td>76.55%</td>
<td>77.96%</td>
<td>94.86%</td>
<td>83.68%</td>
<td>81.91%</td>
<td>37.63%</td>
<td>49.63%</td>
<td>71.46%</td>
</tr>
</tbody>
</table>

6. CONCLUSIONS

Feature selection algorithms are used in the classifier system design only. After the classifier is formed and the features to be fed into the classifier are decided, the feature selection algorithms will become obsolete. So the time requirements of the feature selection algorithms are not vital for the pattern recognition operation.

The feature selection algorithms are forced to select the best three features for each texture analysis method. Selecting three features by each method is forced to confirm the uniformity between the number of features selected from each feature selection method.
Some feature selection algorithms used in the literature have cost functions, such that they also decide on the number of features to be selected. In practical applications since the features will be fed into classifiers, the strengths and weaknesses of the classifiers must also be taken into account in feature selection. One important problem for classifiers is curse of dimensionality [14]. The problem is directly related to the number of features fed into the classifier. So it is better to limit the number of features to be selected.

In addition to the existing feature selection algorithms used in literature, two new feature selection algorithms are introduced in this paper. First of the newly formed methods is Selection by Shape Similarity. Although it is a relatively simple feature selection method, the results indicate that it can be used as a feature selection algorithm in other feature sets also.

Second of the new formed algorithms is a method derived from Fisher’s measure. In this new approach, instead of calculating the distance between class centers and scattering of the data for each class, clustering algorithms are used. The strength of this method over other feature selection algorithms is that it also concentrates on the cases where the part of one class is located in the region of another class. In many real time applications this kind of features are faced. For this purpose non-parametric classifiers are introduced in the literature. For most of the feature selection algorithms, the features are acceptable where classes are linearly separable. Fisher’s measure with clustering is designed to select features where the classes are not necessarily linearly separable. The requirement for selection is class separation, either linearly or non linearly.

The results obtained from the suggested method are also promising and outperformed many feature selection algorithms tested with the same feature sets.

The features selected by each method are fed into classifiers to test the classification success of rusty and clean steel surfaces without finding the rust grade and cleanliness degree. To find a common measure between the selection algorithms, rusty and clean type discrimination is done.

Classification of rusty and clean surfaces is one of the main results of the paper. If the recognition operation is not detailed and only a decision is required whether the surface is rusty or clean, results of this paper is enough for recognition operation.

For each feature selection method, the feature set that has classification success over 99.5 percent is given in shaded form. Having such a high degree of classification success is promising for real time adaptation of the study.

The classifiers used for this operation are KNN and Minimum Distance Classifier. The aim in choosing these two different kinds of classifiers is, to use the results of the classification success in the performance measure that is used in grading feature selection algorithms.

In this paper performance measure for feature selection algorithms is introduced as a new measure to grade feature selection algorithms. The idea is to find the average power classification power of the features selected by the feature selection algorithm. The results of the performance measure for feature selection algorithms are given in Table 5.2.5

From Table 5.2.5, the prepared performance measure indicates that, Fisher’s Measure with SOM outperformed other feature selection algorithms for all the weight sets.

Performance measure can also be used to grade texture analysis methods. For that reason the average classification power of each texture analysis method is given in the Table 5.2.6. The results indicate that GLCM performed better than other texture analysis methods in general sense.

To point out here performance measure introduced in this paper, concerns with the average performance of the feature selection algorithms and texture analysis methods. One
feature selected by a feature selection algorithm may be superior but other selected ones may decrease the performance of the selection algorithm.

REFERENCES:


