Mohammad Ehsan Momeni Heravi, 1 Mohammad Hossein Moattar2

1 Department of Textile and Fashion Design, Mashhad Branch, Islamic Azad University, Mashhad, Iran.
2 Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

Carpet Back Sizing Quality Assessment by Measuring the Amount of Resin Using Image Processing and Machine Learning Approaches

Ocenjevanje kakovosti utrjevanja hrbtnine strani preproge z določanjem količine zamreževala s pomočjo obdelave slik in strojneg učenja

Original scientific article/Izvirni znanstveni članek
Received/Prispelo 6-2023 • Accepted/Sprejeto 10-2023

Corresponding author/Korespondenčni avtor:
Mohammad Hossein Moattar, Associate Professor
Tel.: +989155030540
E-mail: moattar@mshdiau.ac.ir
ORCID: 0000-0002-8968-6744

Abstract

The mechanical properties of the carpet, such as dimensional stability, bending stiffness, handle and creeping on the surface during use, have a direct relationship with the amount of resin applied to the back of the carpet in the sizing process. In today’s factories, the optimal amount of resin and the mechanical quality of the carpet are controlled by the operator touching the carpet on the machine carpet finishing line or manually while rolling the carpet. Proposed in this paper is an automatic method based on the evaluation of the bending stiffness of the sized carpet that uses digital image processing and machine learning to measure the optimal amount of size concentration and control this index. For this purpose, during the final stage of carpet production, the carpet is folded in the middle, and two edges of the carpet are placed on top of each other. A side view image is then taken of the carpet. Using edge detection methods, the edges of the carpet are identified, and different features, such as the average, maximum and minimum statistics for the curve and contour angles, are then extracted. Different conventional machine learning approaches, such as KNN, CART and SVM, are applied. To evaluate the proposed method, a dataset containing 220 different images is used in a 10-fold cross-validation scheme. Different performance measures resulting from the evaluations demonstrate the effectiveness and applicability of the method.

Keywords: carpet quality assessment, carpet back sizing, digital image processing, machine learning, edge detection

Izvleček

Mehanske lastnosti preproge, kot so dimenzijaska stabilnost, upogibna to gost, otip in drsenje na površini med rabo, so neposredno povezane s količino zamreževala, nanesenega na hrbtno stran preproge pri procesu izdelave. Opti-
malno količino zareževala in mehanske lastnosti preproge pri izdelavi danes ocenjuje operater z ročnim otipom preproge v strojni liniji v fazi plemenitenja ali med njenim zvijanjem v svitek. V članku je predlagan samodejni postopek, ki temelji na ocenjevanju upogibne togosti utrjene preproge. Ta zajema obdelavo digitalne slike in strojneg učenja za določitev optimalne koncentracije zareževala in njenega nadzora. V ta namen je bila preproga v zadnji fazi izdelave na sredini prepognjena, pri čemer sta bila dva robova preproge položena drug na drugega. Nato so bile posnete fotografije preproge z stranskega pogleda. Z metodami zaznavanja robov so bili razpoznani robovi preproge, za krivulje obrisa in kotov krivulje obrisa pa določeni statistični parametri, kot so srednje, minimalne in maksimalne vrednosti. Uporabljeni so bili različni konvencionalni pristopi strojneg učenja, kot so KNN, CART in SVM. Za ovrednotenje predlagane metode z 10-kratno navzkrižno validacijo so bili uporabljeni podatki 220 različnih fotografij. Različna merila uspešnosti, ki izhajajo iz ocen, so dokazala učinkovitost in uporabnost metode.

Ključne besede: ocena kakovosti preproge, utrjevanje hrbtne strani preproge, digitalna obdelava slik, strojno učenje, zaznavanje robov

1 Introduction

Mechanical and visual properties significantly affect the quality of the carpet. Mechanical properties are used to explain dimensional stability, comfort while walking on the carpet and abrasion properties in use, while visual properties are used to explain the appearance of the carpet. To respond to the demand for measuring the visible properties of carpets, many methods and useful tools have been used in the past, including microscopy, photography, densitometry, reflectometry and image processing. The work done in evaluating the mechanical properties of the carpet often relates to the reversibility of the carpet under static or dynamic load. Another characteristic that has been studied in the evaluation of the mechanical properties of the carpet is the fatigue mechanism of the carpet. Fatigue mechanism refers to the durability of carpet piles against shear forces during use, which may cause the piles to come out of the carpet, and is evaluated by weighing the carpet before and after use or testing [1‒3].

Gentry [4] investigated the effect of the type of backing, the density of the tufting machine needle, and the density of the piles per surface unit in a study on carpets that had different primary and secondary backings. One of the important indicators of carpet efficiency is the durability of the carpet’s appearance during use. Among the things that can affect these characteristics is the type, quality and quantity of the latex (resin) on the back of the carpet [5, 6]. In today’s factories, the optimal amount of latex is controlled by touching the carpet with fingers on the production line or when rolling the carpet. The method of checking by a human agent is usually leads to defects and inherent errors that make this difficult. The need for automatic methods is thus greater than ever. Although machine carpet factories are in the midst of a technological transformation, the benefits of using computer-assisted manufacturing (CAM) methods are becoming established in this industry.

Because the amount of resin transferred to the back of the carpet on the finishing line is influenced by many variables and affects many variables of the mechanical properties of the carpet, the optimal amount of resin that is placed on the back of the carpet must be determined and controlled. Bending stiffness is effective in terms of characteristics such as handle, shrinkage and resistance to wrinkles [7]. In the past, many experimental methods have been developed to measure indicators that can quantitatively express the bending stiffness of textiles. Those methods can be divided into two categories. The first category comprises methods that deal with the measurement of the movement, force or energy required to cause bending deformation. The second category involves the measurement of the deformation of the fabric under the influence of its own weight. The
tests based on the second method are simpler and faster, provide better control of the final product and can even be shared online [8].

One of the methods of expressing bending stiffness using the deformation of the fabric under the influence of its own weight, which was invented by Pierce, is the measurement of bending length. Bending length is the length of fabric that bends under the influence of its own weight at a certain angle. A longer bending length means a fabric is harder. The authors of [9] stated that this height of the loop is proportional to bending length measurement and can be used to calculate bending length. In other words, harder fabrics create a higher loop than limper fabrics. Figure 1 shows the basis of this method. Zhou and Ghosh [9] measured the bending behaviour of fabric in terms of the sensitivity of their sizes to shape change. Four types of loops were used for this purpose. Type 1 loops were created by hanging the fabric vertically between two rollers, and type 2 loops were obtained by turning the fabric on itself while there was a separating plate between them. Type 3 and 4 loops were obtained by holding the fabric between two clamps. Each loop was described by several features.

![Figure 1: Basis of the fabric folded loop test [9]](image)

There are a few works that have proposed image processing or other automatic artificial intelligence-based approaches for the quality assessment of fabrics and carpet. In [10], Cassidy et al. used a bending box to measure the bending properties of fabrics whose bending length is difficult to calculate using Pierce’s method. They stated that the width and length of the fabric that are placed on top of each other have a negligible effect on the height of the loop. Xu [11] presented an approach based on image processing to measure the appearance of the carpet, in which the image parameters such as shape, direction and spatial patterns that the human agent uses visually when evaluating the carpet were automatically analysed. Quinones et al. [12] presented a solution based on geometric local binary patterns to examine carpet warp and weft images, which measured the carpet quality based on the balance of grey points in the images. Shady et al. [13] presented a method to evaluate fabric structure characteristics using a Wiener filter to detect weave patterns.

In [14], convolutional neural networks (CNNs) were used as a classification tool for the patterns and colours of the carpets. In [15], visual spectral comparator (VSC) and conventional CT scanning were used for the visual examination of the weaving details and present condition of two 17th-century knotted-pile carpets. An interesting work regarding the use of machine learning, in particular deep learning approaches, in the carpet industry is [16], which proposed deep convolutional generative adversarial networks (GANs) for automatic carpet pattern design and creation. In [17], a method for comparing two carpet photos and determining the best placement of several carpet images was proposed.

The present work, is the first study that attempts to propose a machine learning and image processing-based approach for estimating the amount of coated resin on the back of the carpet and find an automatic method to assess the mechanical quality of carpets. In this research, we first attempted to evaluate and measure the characteristics of the arc in the carpet as a reflection of the bending stiffness by hanging a part of the carpet between two rollers. After coming off the sizing line, the carpet was placed on a flat surface and turned over on itself, and an analysis was performed on the arc created in the carpet in this state. A series of transformations and operations were then performed to extract features, which seemed effective in characterizing the bending loop. Using the extracted features from the images, conventional machine learning classifiers were
trained to discriminate and decide on the amount of the coated resin and to classify carpets based on this aspect.

The rest of this paper is organized as follows. Section 2 presents materials and methods. Experimental setup and evaluation results are presented and discussed in Section 3. Finally, Section 4 presents conclusions and the bases for future works.

2 Materials and methods

2.1 Experimental data

In this research, bending stiffness was used as a feature that is influenced by the amount of resin on the back of the carpet for evaluating the mechanical properties thereof. After the carpet came off the sizing line and was placed on a flat surface, it was folded in the middle; the two edges of the carpet were placed on top of each other and the created curve was then photographed. The amount of protrusion and arch resulting from the folding of the carpet was measured. To evaluate the optimal amount of resin in carpet back sizing, samples of carpets with a density of 490 piles/cm (reed 700 dents/m, weft density 1400 ends/m) were selected. Other structural parameters were as follows: warp yarn count 29.5/4 tex, 80% polyester/20% cotton; weft yarn count 44.8/2 tex jute; pile yarn count 49.2/2 tex acrylic. The resin used was vinyl acetate homopolymer. The prepared samples were divided into five equal parts with a width of 60 cm and a length of 180 cm. One part was coated with the appropriate and optimal amount of resin, and four other parts were coated with 10% and 20% more and less resin than usual, as shown in Figure 2.

![Figure 2: Radius of curvature of the arc formed in the folded carpet a) before back sizing b) after back sizing](image)

The optimal and usual amount of resin to reach the appropriate density in the factory from which the sample was prepared was 135 grams of dry matter per square meter. A 12 Megapixel CCD camera was used for image capturing, where the working distance, or the distance between the objective lens and the folded carpet, was set at 60 cm. The colour temperature for the setting was between 5000 K and 5500 K, and diffuse LED lighting was used as the applied lighting configuration. A total of 220 photos were taken of each of sample, and were used for testing, verifying and comparing edge detection methods. In this way, carpets were classified according to quality in the following six predefined categories: negative 10 relative sizing, negative 20 relative sizing, normal sizing, positive 10 relative sizing, positive 20 relative sizing and pre-sizing.

2.2 Image processing and feature extraction procedure

The proposed image processing and feature extraction process is presented in this section. For this purpose, first the outer contour (outer edge) of the carpet was extracted, followed by the average, maximum and minimum statistics for the curve and angles of the contour. To extract features from folded carpet images, the edges of the carpet had to be detected in the images. Edges are local changes in
image brightness. On the other hand, shadow changes result in brightness changes but are not edges. For this reason, extracting the true working edges was difficult.

A large number of edge detection operators function based on the first derivative of the brightness intensity. Derivatives of an image show changes in brightness in both vertical and horizontal directions. Several operators were presented to calculate local derivatives. Sobel is the best and most general derivative calculation operator. From the convolution of windows and in the image, the derivatives in the x and y directions were calculated, respectively. In equations 1 and 2, represents the derivative in the x direction and represents the derivative in the y direction [18].

\[
G_x = \frac{1}{8} \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-3 & 0 & -1
\end{bmatrix}
\]

\[
G_y = \frac{1}{8} \begin{bmatrix}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & -1
\end{bmatrix}
\]

\[
I'_x(i,j) = (I(i-1,j-1)-2I(i,j-1)-I(i+1,j-1) + I(i-1,j+1) + 2I(i,j+1) + I(i+1,j+1))
\]

\[
I'_y(i,j) = (I(i-1,j-1)+2I(i-1,j)+I(i-1,j+1) - I(i+1,j-1) - 2I(i+1,j) - I(i+1,j+1))
\]

To calculate the magnitude of the gradient, the square root of the derivatives was calculated in two directions, as in equation 3. This relation was calculated for each pixel of the image.

\[
G_{Magnitude} = \sqrt{I'_x + I'_y}
\]

It should be noted that, as a rule, all the points of an image have some difference relative to their adjacent points. Thus, the magnitude of the gradient will not be zero for most points. On the other hand, there is a considerable difference with their neighbours at edge points. For this reason, thresholding can be used to distinguish edges from non-edges.

2.2.1 Fixed threshold limit edge detection

The simplest solution to determine the edge is fixed thresholding on the magnitude of the gradient. In other words, the magnitude of the gradient for all points of the images is compared with a predefined threshold. If the magnitude of the gradient is greater than the threshold, that point is located on the edge. In equation 4, represents the magnitude of the gradient of the pixel, T represents the threshold and E represents the existence of an edge point at

\[
E(i,j) = \begin{cases} 1 & \text{if } G_{Magnitude}(i,j) > T \\ 0 & \text{else} \end{cases}
\]

Considering the variety of the colour of carpets, as well as the different darkness and brightness of the upper and lower background of the carpet, using a fixed threshold to determine the edge does not provide a favourable result.

2.2.2 Canny edge detection

Canny is one of the most common edge detection algorithms. Since the images have noise and small edge changes, a smoothing filter can be applied to the image. A smoothing filter is a filter that removes small changes in the image. In this way, image noises are removed. Most smoothers consider the weighted average of the neighbouring points of each pixel as the new value of the pixel. Many smoothers are proposed according to the size of the neighbourhood and weight matrices. Canny edge detection uses a Gaussian smoothing filter (equation 5). This relation defines the weight of the neighbouring points, in which x and y are the coordinates of the neighbouring point and \( \sigma \) is the smoothing parameter. From the convolution of the window with the image, the smoothed image is obtained [19].

\[
G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}
\]

After applying the Gaussian filter, the gradient of the image was calculated using equation 3 to identify
In vertical pixel navigation, navigation starts from the top of the image and each pixel is compared with its bottom pixel. If their difference is greater than a certain threshold, it is an edge pixel. In equation 6, which determines the final edges of the image, abs() represents the absolute value function.

\[
E(i,j) = \begin{cases} 
1 & \text{if } \text{abs}(I(i,j) - I(i,j + 1)) > T \\
0 & \text{else} 
\end{cases} 
\]  

When the first edge point is seen in a column, the navigation of that column ends.

The use of edge detection methods in determining the edge of the carpet created many unwanted edges that cannot be easily removed. The edges resulting from canny edge detector are shown in Figure 3.

2.2.3 Pixel navigation

It is evident from Figure 3 that edges inside the carpet were also detected in the Canny method. Since the colour of the carpets has many variations, and black and white colours are also used within the carpets, Canny edge detection considers the top black background to be the same as the inside of the carpet. This may result in many false positives in edge point detection. Thus, to make the edge extraction simpler and more accurate, pixel navigation was used.

Since only the top and left edges of each carpet were needed, the pixel traversal method was suitable for obtaining these edges. In the available image, the background colour is very different from the edge of the carpet. Thus, by scrolling vertically from top to bottom, the image of the upper part of the contour was obtained, and by scrolling horizontally from left to right, the side part of the outer contour of the carpet was obtained. Figure 4 shows how to calculate the contour using the pixel navigation method.
2.2.4 Jump-based pixel navigation

In the pixel navigation method, the distance between the value of two neighbouring pixels, \( p \) and \( q \), is calculated at each step of the navigation, and if that distance is lower than the threshold, it means that the contour is not yet found. In this situation, the algorithm moves one pixel down in the column. This single pixel navigation method works very slowly in high resolution images. To improve the pixel navigation method, jump-based pixel navigation can be used.

In this approach, navigation is done in multi-pixel steps. A navigation step of \( s \) is used for this purpose, and each pixel is compared with a pixel in the column-wise spatial distance of \( s \). Moreover, to increase the resistance of the method to noise, the average values of five adjacent pixels are considered instead of comparing the values of two pixels. These two approaches for determining the edge point positions are denoted in equation 7.

Horizontal row-wise scrolling was also performed to calculate the left contour to increase the speed and accuracy, and the contour obtained from the previous step was used. Since the contour is continuous, the location of the contour in each row is close to the location of the contour in the upper row. Thus, horizontal scrolling was performed to obtain the exact location of the contour near the contour location of the upper row. Supposing that \( (r, c) \) was the location of the contour in the previous line, horizontal scrolling was performed in the spatial row-wise distance of \( (r, c-k) \) and \( (r, c-k) \), where \( k \) is an arbitrary number. Again in horizontal scrolling, instead of comparing the value of two adjacent pixels, the average difference, as calculated in equation 7, was used.

After determining the edges, probable noises were removed using a morphological thinning operation, which is a morphological closing followed by a morphological opening to repair possible edge discontinuities. After morphological operations, carpet contour with a thickness of one pixel is obtained. Figure 5 shows the contour calculation steps.

\[
E(i, j) = \begin{cases} 
1 & \text{if } \frac{1}{5} \sum_{n=-2}^{2} l(i, j + n) - \frac{1}{5} \sum_{n=-2}^{2} l(i, j + s + n) > T \\
0 & \text{else}
\end{cases}
\]  

(7)

2.2.5 Comparing the proposed edge detector

In this section, a comparison is made between the stated methods on real images. The results of applying different edge detection methods are shown in Figure 6. As is evident, the methods of fixed threshold and Canny (b and c) created many unwanted edges, while the pixel navigation method determined a pure edge contour for the carpet. The results of the application of pixel navigation and jump-based pixel navigation are similar in most cases, while the latter was faster.
As seen in the pictures, the jump-based pixel navigation method demonstrated good accuracy and its execution speed can be increased by adjusting the jumping parameter. The results of the implementation of this method on 220 carpet images show that it can accurately identify the edges of the carpet at a suitable speed. The results of the edge detection were then used for carpet classification based on the contour. Different features, such as carpet curvature, angle, length-to-width ratio and contour radius, were extracted and used for the quality classification of manufactured carpets. It is also evident that the lower edge was omitted, since the lower edge of the carpet does not possess any information regarding fold and curvature.

2.2.6 Feature extraction

For classification purposes, we needed to extract descriptive features from the curvatures resulting from the edge detection step. These features should express the curvature of the carpet. For this purpose, the features of carpet curvature, including the angle of the oblique line, length-to-width ratio and contour radius, were extracted, as shown in Figure 7.

The features of the length-to-width ratio and the angle of the oblique line that each carpet makes with the horizontal axis were independent of the distance of the camera from the carpet. The maximum and minimum column difference of the contour denotes the width, while the maximum and minimum row difference of the contour denotes the length of the curvature. Using the Hough transform, which is a transformation that finds geometric shapes in an image, the largest interior curvature of the carpet was obtained. To make it independent from the distance between the camera and the carpet, it was then divided by the length-to-width ratio feature.
Figure 7: First three primary features of the curvature: a) length-to-width ratio; b) the angle of the oblique line; c) contour radius

On the other hand, the contour of the carpet is a curve, and the main goal was to describe this curve in such a way that it could help the classification process. Hence, the coefficients of this curve were considered features of the contour. For this purpose, equation 8 was used to calculate the curvature or bending of a contour, where \( p \) and \( q \) are the pixel coordinates at the beginning and end of a contour segment, respectively. Moreover, equation 9 denotes the angle of curvature between two edge points \( p \) and \( q \). In fact, the curvature is the ratio of the Euclidean distance to the Manhattan distance of the first and last pixels of the sequence.

\[
\text{Curvature} = \frac{\text{Euclidean distance} (p, q)}{\text{Manhattan distance} (p, q)}
\]

(8)

\[
\text{Angle} = \arctan\left(\frac{q_y - p_y}{q_x - p_x}\right)
\]

(9)

where \( q_x \) and \( q_y \) represent the x and y coordinates of point \( q \). Since calculating a single value does not reflect the total shape of the curve, edge pixels were sorted from the right-most pixel to the left-most pixel. The array was then divided to \( C \) parts and, considering the points at the beginning and end of each segment, the curvature features were calculated for each one. The curvature and angle statistics of these \( C \) segments, including the maximum, minimum and mean values, were then determined as the final describing features. In summary, the extracted features for describing the contour of the folded carpet are as presented in Table 1.

<table>
<thead>
<tr>
<th>Feature</th>
<th>No. of coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length-to-width ratio</td>
<td>1</td>
</tr>
<tr>
<td>Angle of the oblique line</td>
<td>1</td>
</tr>
<tr>
<td>Folding radius</td>
<td>1</td>
</tr>
<tr>
<td>Curvature statistics (i.e. maximum, minimum and mean of the curvature)</td>
<td>3</td>
</tr>
<tr>
<td>Angle statistics (i.e. maximum, minimum and mean angle of the angle)</td>
<td>3</td>
</tr>
</tbody>
</table>

The initial size of the images was 3000 × 4000 pixels. However, to reduce the calculations, the size of the images was changed to 375 × 500 pixels. Also, other parameters of the approach were set according to Table 2.

3 Experimental results and discussion

After feature extraction, three different methods were used as learners and classifiers. In the following evaluations, three traditional classifiers, i.e. K nearest neighbour (KNN), classification and regression tree (CART) and support vector machine (SVM), are used. Although deep learning approaches are the most recent classification paradigms, they could not be applied because the problem was a low dimensional one and the size of the evaluation database was not large. Since SVM is a binary classifier, it was trained in a one-versus-all scheme.
All the experiments were performed using the Windows 10 Pro operating system with an i5 core (3.20 GHZ) CPU and 16 GB of RAM, while Python version 3.6 was used to implement the proposed method. Traditional 10-fold cross validation was used for evaluations and classification accuracy, which is the number of correct decisions divided by the total number of reported decisions as the evaluation metric.

Table 2: Values of parameters of specific approach

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>20</td>
</tr>
<tr>
<td>K</td>
<td>10</td>
</tr>
<tr>
<td>C</td>
<td>30</td>
</tr>
<tr>
<td>T</td>
<td>10</td>
</tr>
</tbody>
</table>

3.1 Discussion about features
The first step was to analysis the discriminating power of the extracted features. To do so, the histograms of the features were plotted and compared in Figures 8 and 9. Figure 8 illustrates the distribution of the folding radius feature for different classes, while Figure 9 presents a histogram of the angle of the oblique line for the five different classes. As seen in both figures, different classes have wide overlapping regions considering the two investigated features, while the features are not discriminative by themselves. For example, as seen in Figure 8, all neighbouring and similar classes have a decision region in common, which is approximately half of the width of the distribution. On the other hand, it can clearly be seen that even two distant classes have common decision regions. This means that the features are not independently good ways for differentiating between the classes. It is the main reason that classification models are trained using a set of nine different features and used for the decision.

Another measure for determining the discriminating power of the features is the Fisher ratio. The Fisher ratio can quantify the discriminativeness of the features and is the ratio of between-class scatter to within-class scatter. Since the problem was a multiclass classification problem, we had to average the value over all possible class combination as denoted in equation 10. The Fisher ratio values are summarized in Table 3 for each feature. As denoted, features such as the angle of the oblique line and folding radius had the highest Fisher ratio, which indicates the greater effectiveness of the features.

Figure 8: Distribution of folding radius feature for different classes. The overlap between every two classes is shown in a different colour.
3.2 Discussion about classification results

Since the features were not effective by themselves, a second phase of feature transformation using linear discriminant analysis (LDA) was performed on the initial feature vector of size 9. LDA is usually used as a dimension reduction approach, but here it was used for feature transformation. Table 4 indicates the classification performance of each machine learning model, with and without LDA. Moreover, since the evaluations were performed in a 10-fold cross validation scheme and the results were averaged, performance standard deviations are also included in the table.

The best accuracy on each assessment condition is shown in italics and the best overall accuracy is denoted in bold typeface. Moreover, for the better discussion of the results, the average accuracies of the classifiers are summarized in the last column. The number in parenthesis indicates the rank of each classifier based on its average performance. In this table, the classifiers are also evaluated with different parameters and settings. For the SVM classifier, two different traditional kernel functions, referred to as radial basis functions (RBF), and a linear kernel were investigated. KNN classifier is evaluated with different K values (i.e. 3, 5, 7). Since there were not

![Figure 9: Distribution of the folding angle of the oblique feature for different classes. The overlap between every two classes is shown in a different colour.](image)

\[
\text{Average}_{\text{FIS}} \quad \text{ratio} = \frac{2}{Nc(Nc - 1)} \sum_{i=1}^{Nc} \sum_{j=i+1}^{Nc} \frac{(m_i - m_j)^2}{v_i + v_j}
\]

(10)

<table>
<thead>
<tr>
<th>Feature category</th>
<th>Feature name</th>
<th>Average Fisher ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Length-to-width ratio</td>
<td>4.11</td>
</tr>
<tr>
<td></td>
<td>Angle of the oblique line</td>
<td>11.42</td>
</tr>
<tr>
<td></td>
<td>Folding radius</td>
<td>10.64</td>
</tr>
<tr>
<td>Curvature</td>
<td>Min. curvature</td>
<td>2.3</td>
</tr>
<tr>
<td></td>
<td>Max. curvature</td>
<td>9.21</td>
</tr>
<tr>
<td></td>
<td>Average curvature</td>
<td>7.43</td>
</tr>
<tr>
<td>Curvature angle statistics</td>
<td>Min. angle</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>Max. angle</td>
<td>14.5</td>
</tr>
<tr>
<td></td>
<td>Average angle</td>
<td>10.23</td>
</tr>
</tbody>
</table>

Table 3. Average Fisher ratio of each extracted feature
many samples in the classes, higher values for K are not useful. In addition, as seen in the table, the average performance of the KNN classifier degrades when K increases to 7.

**Table 4: Classification accuracy of the different classifiers on the proposed approach**

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Parameters</th>
<th>LDA</th>
<th>Accuracy (%)</th>
<th>Average accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>RBF kernel</td>
<td>No LDA</td>
<td>76.11±2.83</td>
<td>85.19 (1)</td>
</tr>
<tr>
<td></td>
<td>LDA (d = 3)</td>
<td>82.44±3.57</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDA (d = 4)</td>
<td><strong>91.69±2.85</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDA (d = 5)</td>
<td>90.54±3.01</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Linear</td>
<td>No LDA</td>
<td>69.13±3.37</td>
<td>79.77 (4)</td>
</tr>
<tr>
<td></td>
<td>kernel</td>
<td>LDA (d = 3)</td>
<td>82.13±3.60</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA (d = 4)</td>
<td><strong>88.08±3.4</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA (d = 5)</td>
<td>79.55±3.84</td>
<td></td>
</tr>
<tr>
<td>KNN</td>
<td>K = 3</td>
<td>No LDA</td>
<td>69.60±4.07</td>
<td>80.32 (3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA (d = 3)</td>
<td>79.78±7.18</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA (d = 4)</td>
<td><strong>86.51±4.37</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA (d = 5)</td>
<td>85.39±6.98</td>
<td></td>
</tr>
<tr>
<td>K = 5</td>
<td>No LDA</td>
<td>72.29±5.93</td>
<td></td>
<td>81.09 (2)</td>
</tr>
<tr>
<td></td>
<td>LDA (d = 3)</td>
<td>77.15±7.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDA (d = 4)</td>
<td><strong>88.94±7.53</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDA (d = 5)</td>
<td>85.99±4.46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K = 7</td>
<td>No LDA</td>
<td>68.28±7.92</td>
<td></td>
<td>78.46 (6)</td>
</tr>
<tr>
<td></td>
<td>LDA (d = 3)</td>
<td>79.66±5.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDA (d = 4)</td>
<td><strong>87.89±5.96</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDA (d = 5)</td>
<td>78.02±5.56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CART</td>
<td>No LDA</td>
<td>68.04±3.12</td>
<td></td>
<td>78.67 (5)</td>
</tr>
<tr>
<td></td>
<td>LDA (d = 3)</td>
<td>78.75±6.62</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDA (d = 4)</td>
<td>85.99±5.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LDA (d = 5)</td>
<td>81.93±6.9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The results of Table 4 are also summarized in Figure 10. This figure illustrates both the effectiveness of the SVM classifier and the LDA feature transformation. This figure shows that using LDA can considerably increase the classification accuracy of the approach, with SVM with an RBF kernel being the best choice on this problem from amongst the evaluated classifiers.

A lingering concern is the time complexity of the process, especially the proposed edge detection approach. Table 5 presents the running time of the edge detection process and shows the lower computational time of the proposed edge detector. It should be noted that the proposed edge detector is customized for this problem and its efficiency is due to the fact that the edge detection process is designed specifically for this problem.

**Table 5: Average processing time of the edge detection approaches per image**

<table>
<thead>
<tr>
<th>Method</th>
<th>Processing time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed threshold</td>
<td>0.0164</td>
</tr>
<tr>
<td>Canny edge detection</td>
<td>0.246</td>
</tr>
<tr>
<td>Pixel navigation</td>
<td>0.0047</td>
</tr>
<tr>
<td>Proposed jump-based pixel navigation</td>
<td>0.0041</td>
</tr>
</tbody>
</table>

**Figure 10: Classification accuracy of the different classifiers on the proposed approach**

**4 Conclusion**

This paper represents the first attempt at the mechanical quality assessment of the carpets based on the amount of their back sizing using an automatic image processing and machine learning based approach. The approach proposes a novel edge detection scheme for this purpose, which is followed by the extraction of representative features. A traditional LDA approach was the used for feature determination and was followed by some
traditional simple classifiers, from which SVM with an RBF kernel demonstrated the best results. The evaluations show the effectiveness of the proposed approach in determining the number of coatings and the achievement of a classification accuracy of 91.69 percent. Moreover, the proposed approach was evaluated from a computational time point of view, which proves the real time applicability of the method. For future works, using the back texture of the carpet is suggested for the purpose of quality assessment. More complex learning approaches, such as convolutional neural networks, may be better suited for this purpose.

Availability of data and materials: The datasets used and/or analysed during this study are available from the corresponding author upon reasonable request.

Competing interests: The authors hereby declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Funding: The authors hereby declare that no funding was received for this research.

References

12. QUINONES, Rolando, ORJUELA, Sergio A., ORTIZ-JARAMILLO, Benhur, VAN LANGEHOVE, Lieva, PHILIPS, Wilfried. Quantifying appearance retention in carpets using


