

Pattern Recognition 2010/ 2011

Wood Species Classification Project

Final Report

Miroslav Radojević, Peter Rennert, Babak Rezaï, Isabel Rodas, Chengjia Wang

Abstract—This work intends to contribute to the study of wood texture classification by implementing and evaluating the performance of several feature extraction methods applied in combination with a variety of classification techniques. An exotic wood texture images dataset has been used to test the generated code. Implemented feature extraction methods comprise gray level cooccurrence matrix, mathematical morphology, ranklets, curvelets, wavelets and local binary patterns; tested classification techniques include k-nearest neighbour, linear discriminant classifiers, quadratic discriminant classifiers, neural networks, and support vector machines. Results have been evaluated based on computation time and classification accuracy, the highest success rate having been achieved by a novel scheme integrating X and X features introduced in this report.

I. INTRODUCTION

Texture analysis can be a challenging task in many respects, especially in terms of generalisation, as generally tailoring to the required application is needed. Indeed, no general method exists that can be applied to any kind of texture [1]. Many different methods for feature extraction, clustering and classification exist in different areas of use with proven good results. Textural properties carry useful information for discrimination, and they are particularly interesting in applications requiring discrimination between materials, as their texture is the fundamental property to be classified.

In this work, classification of wood texture is undertaken. An exploration on the different feature extraction techniques and their applicability to this particular case of classification has been developed, as well as the subsequent study of different classification algorithms and their efficiency when applied in combination with the implemented texture analysis techniques.

For the evaluation of results, different objectives have been stated in the search for an optimum discriminative and fast method. Firstly, the tested techniques have been evaluated in terms of their efficiency and error rate, and in relation to their computational requirements. Secondly, it has also been an aim to achieve robustness to pose and scale variation, as well as it has been attempted to generate a method that can deal with illumination variations.

This report is organised as follows. Section § IV is devoted to the different feature extraction methods that have been applied, initially presenting their general

characteristics, and then describing the variations and improvements generated. Section § VI is dedicated to the different classifiers that have been used, with a brief description for all of them. Section § VII focuses on the implementation and the results that have been achieved, and finally Section § IX presents the conclusions that have been drawn.

II. DATABASE

The provided database contained 23 classes that could be divided into further subclasses of wood species. The dataset was acquired by a specially designed device to capture a close up of a wood section. Initially the dataset had about 4000 training and 512 test images. However after a first inspection of the dataset duplications were discovered. With a MD5 checksum script all duplications were detected and deleted, leaving a training set of 1180 images and a test set of 517 images.

After further inspection of the dataset was discovered that the set of training images of each subclass seemed to be taken from a single sample with an enormous overlap between the images, the same hold true for the test set exemplary, figure 1 illustrates this fact on the Balau 1 class. Even if the training and test data seem to be from different samples and do not show obvious overlapping, the partially great variety between the subclasses make this classification task nearly a one-to-one classification. The classifier will be trained on the features of mostly one image and tested against the features of only one image. This assumption is supported by the fact, that the classification with ranklets for instance is nearly as good for subclass specific classification as for classification on the 23 main classes.

After the discovery of the duplications and the low variety within the classes, another dataset was made available. This dataset showed similar overlapping then the first set and was organized in a different way. Instead of an Excel file indicating the the scientific and common names for each class, this dataset is ordered by the scientific family name only. Moreover this dataset was not separated into a training and test set and not all classes of the first set were contained in the second class. We decided to do not use this dataset, because we were not sure how to merge them correctly.

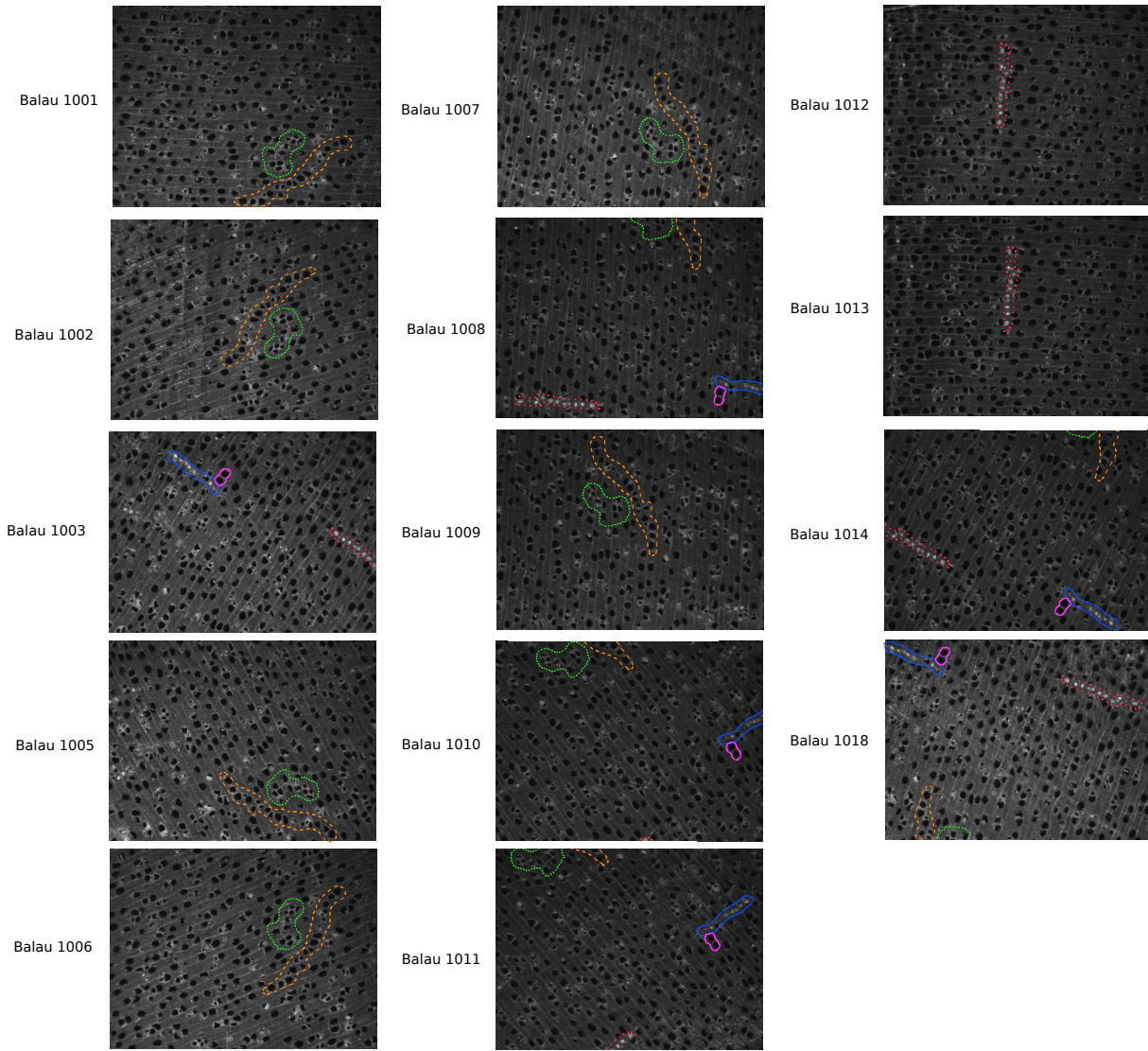


Fig. 1: The huge overlap of the images in the database shown on the example of all available training images from Balau 1. The coloured markers indicate distinguishable holes, which can be find in many of the images. Please note, that not the positions of the markers is the most relevant, but the depicted area in between them. This area is to great extent the same. Please note also, that images with numbers missing in this collection have been removed, because they had the same MD5 checksum as some images shown here.

It has also to be stated out that since this second dataset was not separated into a training an test set, the use of a N- fold cross validation as done in [2] (while working on this dataset) is very questionable, since it is very likely that one trains the classifier that has huge overlaps with the test images. Moreover, the use of the separated data from the first dataset will also not lead to interpretable results, because the training and test set are very limited. Especially a highly adaptive classifier such as a SVM with radial kernel or an

neuronal network with many layers or nodes might be overfitted to the test data, if tuned too much. Even a simple feature reduction method lead to overfitting if the amount of features is tuned too much, as shown later.

The dataset has several other drawbacks that have to be corrected or taken into account. Firstly, the illumination is very uneven between the images of one class and a strong non-uniform illumination can be found on the images. Additionally, some images have a strong blur.

This happens mostly at the borders, but some images are completely blurry.

III. PREPROCESSING

A. Illumination Correction

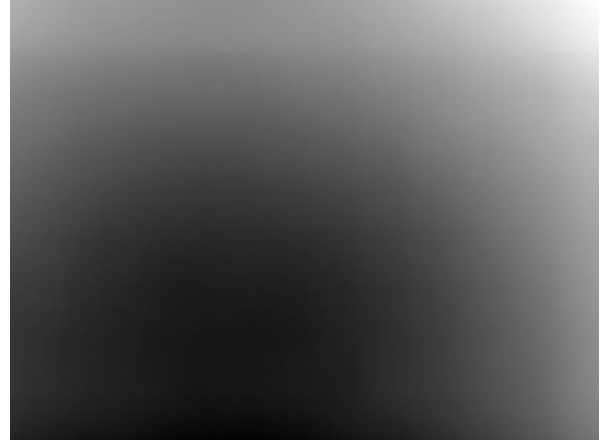
1) *Using Gaussian Filter*: The idea of illumination correction using a Gaussian filter is to approximate the overall illumination of the image by convolving the image with a Gaussian of a big standard deviation and kernel size. Doing so, the image gets smoothed out by the low pass filter and only the light as a global structure remains. This light pattern can then be simply subtracted from the original image to obtain an illumination corrected version. As a side effect, the average image intensity is normalized as well. This can be a particular problem if different classes are mainly distinct by there brightness.

The disadvantage of this method is that a convolution with a large kernel is very time consuming and a single convolution will often not lead to good results, because coarse structures such as big holes might still be distinguishable in the illumination approximation. Therefore, one needs to convolve the image several times with a large scale Gaussian filter, in case of this project good results were obtained by applying iteratively 4 Gaussian filters with a standard deviation of 500 and a kernel size of 50x50 pixel.

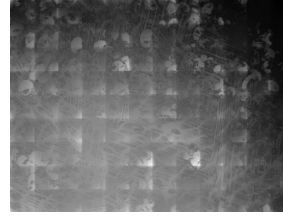
2) *Using Wavelet Decomposition*: Similar to the Gaussian approach, the illumination of an image can be approximated by the use of wavelet composition to a large scale. Since the approximation coefficients of a wavelet decomposition are obtained by a low pass filter, if they are removed only details will remain. If the decomposition is done with enough levels the approximation coefficients will only encode the illumination. Therefore, to remove the influence of the illumination, simply the approximation coefficients have to be set to zero before the image is composed again.

In order to get good results, it is crucial to use a wavelet with appropriate vanishing moments, that are able to model the illumination source. A wavelet can approximate a polynomial function with order of $n - 1$ by n vanishing moments. It is therefore necessary to choose a wavelet function with few vanishing moments. On the other hand low vanishing moments lead to artefacts in the composition if the approximation is removed.

For this work, the wavelet composition was done with the symmetric wavelet 2 from the Matlab toolbox, with 7 levels. The wavelet based illumination correction on the images takes about 0.4 and so is twice as fast as the convolution with Gaussian and the results are much better, even visually. The images are much more equalized and more sharp then after applying the Gaussians. All cluster segmentation based methods increased their performance significantly after the illumination correction.



(a) Global Illumination model computed from the training set images (combination of wavelet and Gaussian result)



(b) Wavelet difference MIP



(c) Gaussian difference MIP

Fig. 2: Illumination correction. 2a shows the estimation of the global illumination. That all images lay under the influence of similar illumination conditions is shown with the maximum intensity projection in figures 2b and 2c. Details can be found in the text.

B. Illumination Model

Even if the images of the dataset were acquired by a specially for this purpose designed device, supposedly screened from external illumination influences, the images of the dataset show all a strong non-uniform illumination. However, due to the screening there is a strong suggestion that there is an underlying illumination model that is the same for all images. This hypothesis was not verified into detail, since all available images have the dominating wood pattern which makes errors in the estimation of the illumination model very likely. Nevertheless simple tests were performed, which support this thesis. In figure 2 the results of these tests are shown. Very important are figures 2b and 2c. These images show the maximum intensity projection (MIP) of the difference of between estimated model and the illumination corrected result.

The MIP show the maximum values that were changed from original version to the illumination corrected one in any of the training images. For both methods a black upper right corner is visible. This means in *none* of the 1080 training images this area was corrected significantly. This is expected in a constant light distribution, since both methods, Gaussian and wavelet approach, remove the bright parts of the

image, the image is normalized to its darker regions. In the other parts are changed according to the surface properties of the wood. Interesting are also the artefacts produced by the wavelet decomposition.

Once it is verified that a general illumination model exist and a proper estimation would be done, it could be simply subtracted from each wood image which would make a efficient illumination correction possible that can be performed in an instant.

C. Rotation Alignment

As shown in section II, all wood types have year rings which appear nearly as lines due to the magnification. These lines can be used for an easy angle adjustment with the objective to make rotation variant feature extraction methods rotation invariant and to get access to orientation based features.

1) *Line Estimation Hough Transform*: Even if the now implemented version is a very straight forward strategy, more complex approaches were tested at the beginning. The most exhaustive tries were done with Hough transforms for lines, that are provided as Matlab toolbox. However, any attempts to solve the problems were unfruitful, the Hough transform turned out to have a very low robustness on the wood dataset. Problems that arose with Hough transform reached from the detection of too small line segments in arbitrary direction, or even showing a systematic direction that was somehow not correlated with the year rings at all. All approaches to fix that bad behaviours, including tuning of the function parameters, pre and post processing with mathematical morphology operations and image sharpening and smoothing filters failed.

In order to increase the strength and continuity of the gradients, which were seen as reason for the low performance, finally segmentation of the image intensity values by k-means clustering was tested. Even if the segmentation results looked promising, the lines of the wood are seemingly not straight enough for a Hough transform. Other line detectors found, where also not announced to be robust enough. The opposite is often the case in the literature, where modern researchers focus on detection of very straight, but intersected or interrupted lines. It was considered to use a adapted snake model to find the lines before the following method was discovered.

2) *Gradient Based Orientation Estimation*: Finally a very robust solution in the orientation estimation could be found by simply calculating the angle of the gradient of each pixel and finding the peak in the gradient histogram.

The main idea behind this approach is that the wood images contain mainly lines and circular holes. While along the lines many pixel will have a similar gradient direction, the boundaries of the holes will not have dominating gradient direction since they equalize each other around the circular edge. If now the histogram

of the gradient directions is done, it will show a peak perpendicular to the lines.

The implementation is very simple. The direction of the gradients for each pixel are calculated by the use of sobel filters in horizontal Δ_y and vertical Δ_x direction. Then both results are normalized by their magnitude and finally the angle is computed with $\tan^{-1} \frac{\Delta_y}{\Delta_x}$. The maximum of the histogram is detected by simply taking the highest value and interpolating its neighbourhood with a second order polynomial.

The resulting histograms look as exemplary in figure 3b and 3d. Note the huge difference in the histogram, which are used as feature as explained later in section IV-J. The other two images in 3 show the approximated direction of the lines. Note here that the lines in this particular wood types are not straight (figure 3a) even though they are extremely clear or hardly visible (figure 3d). This images were chosen to underline the robustness of this approach. An exhaustive control of the results on the training set shows that there only a few significant failures (if a main part of the image is blurry). For the rest of the images the error is estimated to be maximal at 5°.

This method is not implemented very efficiently at the moment, but the calculation of the angles could be made very fast by the use of look up tables instead of the tangent calculation and other tricks from the robotics and embedded community.

D. Segmentation

1) *K-Means Segmentation*: An completely unsupervised and fast segmentation method was implemented via a k-means clustering based on the intensity values of the histogram. Since it is histogram based the running time is very stable and the image size has nearly no influence on it. On the other hand, the simple version of a intensity based segmentation is very prone to non-uniform illumination. Therefore, it is very crucial to do a illumination correction as described above before the use of k-means segmentation.

The original k-means algorithm uses random seeds as cluster center. This cannot be tolerated in a automatic segmentation, because each run the clusters would belong to different regions of the image. To get a stable result, the seeds are simply fixed. For the training images this version produces stable results in about 0.07 seconds per image. The number of clusters used here is 6, to allow some clusters for noise or transitions and also to give a reasonable cluster amount for the most complex types of wood.

The segmentation result is not perfect in terms of giving perfect segmented lines and holes of the wood any time, but the segmentation is good and stable enough to generate strong gradients for the angle estimation and some feature extraction as described later. 4

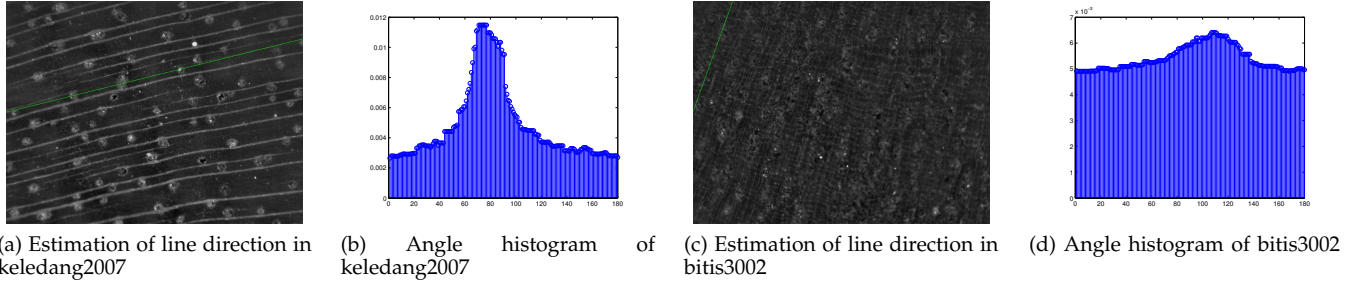


Fig. 3: Angle estimation, the green line indicates the main line direction. These images have a high resolution, zoom in for a better inspection.

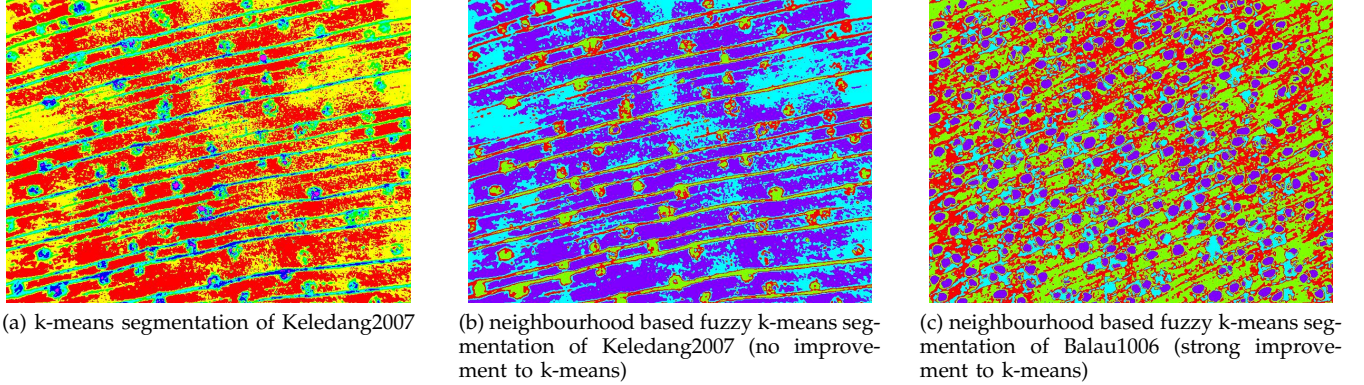


Fig. 4: Comparison between k-means and neighbourhood based fuzzy k-means

2) *Fuzzy K-Means Segmentation*: Later in the project the fcm fuzzy c-means function of Matlab was discovered and tested. It gives not much nicer segmentation results at the same computation time as the traditional k-means segmentation, if used on histograms as well. On the other hand fuzzy k-means gives one segmentation image per cluster. Each of them could be evaluated in the same way as wavelets or other image transforms are processed to obtain features.

However, for the reasons stated above, stable seeds are necessary to get repeatable results and to avoid outliers. This is not possible with the Matlab function. Time constraints did not allow to realize an augmentation of the k-means clustering and is strongly advised for future, to improve the features which are based on k-means clustering and the angle estimation (section IV-J).

3) *Cluster Segmentation using Neighbourhoods*: Another way to employ cluster based segmentation is by taking not only the pixel intensity of a single pixel into account, but also of its neighbourhood. Unfortunately this cannot be implemented as histogram based version, and therefore the computation takes very long. However tests on the training images showed, that it is enough to use only the neighbours in north, east, south and west, even without taking the pixel of interest directly into account. Fuzzy k-means clusters

here very good, separating in the most cases holes, lines and wood texture. Even in difficult cases it keeps giving meaningful results, that can be used for feature extraction as explained later.

This approach was not followed further, because even with optimizations it takes about more then 10 seconds per image, which is considered to be too much for a simple preprocessing.

IV. FEATURE EXTRACTION

A. Ranklets

Ranklets introduce a different way of treating pixel intensities. Ranklets technique analyses the rank of the pixels instead of the pixel intensity itself. This way, information about the intensity is transformed into relative placement of a certain pixel intensity within a window. Namely, essential feature of ranklets is the invariance to monotonic transformations of brightness where they prove to be robust.

1) *Ranklet transform*: Ranklet transform implies calculation of pixel rankings within windows of various orientations and size. Features are extracted from ranklet images obtained from different resolutions and orientations of the ranklet transform [3]. Feature extraction consists of doing ranklet transform - calculating ranklet coefficients, and using proposed method [3] to extract descriptive features from the collection of

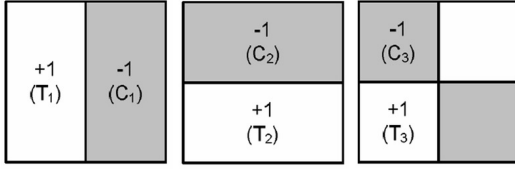


Fig. 5: Vertical, horizontal and diagonal sub-sets of an image crop.

coefficients. The aim of the feature extraction task is to code the variations of grey-scale intensity. Ranklet transform accomplishes that by analysing image in multiresolution and for different orientations [4]. Moreover, the analysis is non-parametric. Eventually, the output of the transform is number of non-parametric ranklet images - depending on resolutions and orientations considered. More resolutions or orientations leads to more image crops, therefore more ranklet images [3]. Sensitivity to orientation for each crop is achieved by dividing the crop into sub-crops, subsets T and C (figure 5). Subsets are defined for different orientations, strategically aligned and positioned so that total score of each crop describes sensitivity of pixel rankings to horizontal (T_1, C_1), vertical (T_2, C_2) and diagonal orientation (T_3, C_3). Similarly as in Haar basis functions of Mallat and square sub-regions used in face detection algorithm by Viola and Jones, regions are opposed against each other, making total score orientation sensitive. Ranklet coefficient of a crop of a certain size ($R_i, i = 1(hor), 2(ver), 3(diag)$) is obtained by counting the sum of ranks of pixels from T region for different orientations. The score is normalized so that it converges to +1 if all pixels from $T_i, i = 1(hor), 2(ver), 3(diag)$ have higher ranking than pixels in C_i . Likewise, the score values close to -1 if as many pixels from C_i have higher ranking than the pixels from T_j . Hence, close to zero score means weak horizontal, vertical or diagonal variation in ranking. Rankings are calculated within each crop. Instead of brute force direct comparison, which is computationally demanding, relative rank of pixels' grey-level values is calculated instead of doing "one-on-one" comparisons, giving as output a normalized value taking the interval $[-1, +1]$, as shown in [3]. This way, computational complexity becomes $O(N \log N)$ instead of $O(N^2)$. Such approach uses *Quicksort* algorithm to compute the Wilcoxon statistic of Haar-like ranklet scores [5]. Generally, computational expense is the main obstacle in application of ranklets due to the usage of sorting algorithms. Fast algorithm for computation of ranklets was introduced in works of Smeraldi ([5]), with computational complexity $O(\sqrt{N} + k)$ giving significant speed-up in calculation time. This algorithm is based on Distribution Counting algorithm for calculating Ranklets with linear complexity [5].

2) *Features extraction*: To sum up - ranklet transform takes out image crops for different scales calculates

vertical, horizontal and diagonal ranklet coefficients (R_1, R_2, R_3) ranging from -1 to +1. Next step is extraction of significant features from quantized ranklet images. Quantization is carried through by quantizing coefficients so that they take values ranging from -1 to +1, with step 0.1 resulting in 21 bin as suggested in [3]. Each ranklet image ends up having 11 features obtained from ranklet histogram (RH), equation 1 and ranklet co-occurrence matrix, (RCM), equation 2) of ranklet coefficients as suggested in [3] and [6].

$$RH(bin) = \frac{n(bin)}{\sum_{j=1}^{21} n_j}, bin = 1, \dots, 21 \quad (1)$$

$$RCM(bin_1, bin_2) = \frac{n_{d,\theta}(bin_1, bin_2)}{\sum_{i,j=1}^{21} n_{d,\theta}(i, j)}, bin_1, bin_2 = 1, \dots, 21 \quad (2)$$

where n is the number of coefficients taking one of the discrete bin levels, $n_{d,\theta}$ number of occurrences of two bin values d pixels apart, along angular rotation θ . Details about co-occurrence matrix (for grey-levels) are available in section (§ ??). Generally, histogram expresses probability distribution function of binned ranklet coefficient values, while the ranklet co-occurrence matrix (RCM) does the same for possible transitions between discrete pairs. A short overview of the eleven extracted features, according to [3] is shown in section § ?. Average of four co-occurrence matrices with ranklet coefficient was used in order to be rotation invariant, at least at 45 degrees. Therefore, four ranklet coefficient CMs are averaged: those that correspond to angular rotations of $\theta = 0^\circ, 45^\circ, 90^\circ$ and 135° . $rcmAvg = \frac{1}{4}(RCM_{1,0 \text{ deg}} + RCM_{1,45 \text{ deg}} + RCM_{1,90 \text{ deg}} + RCM_{1,135 \text{ deg}})$ Distance (d) is fixed to 1 and the information extracted is qualitatively statistical - mostly moments of first and second order. There is no need for higher values of d because multi-resolution is already supported.

3) *Ranklet features*: labelsec:rankFeat

- **Mean convergence** is expressed as $\sum_{i=1}^{21} \frac{1}{\sigma} |bin(i)RH(i) - \mu|$ where $bin(i) = [-1, 0.9, \dots, 0.9, 1]$ is discrete ranklet value μ , σ are mean and standard deviation of the ranklet coefficients, respectively. This feature uses the ranklet histogram (RH).
- **Code variance**, $\sum_{i=1}^{21} (bin(i) - \mu)^2 RH(i)$ is variance of ranklet coefficients.
- **Code entropy**, $\sum_{i,j=1}^{21} RCM(i, j) \log RCM(i, j)$
- **Uniformity**, $\sum_{i,j=1}^{21} RCM(i, j)^2$
- **First order element difference model**, $\sum_{i,j=1}^{21} |i - j| RCM(i, j)$ gives an estimate of the transition probability, weighted by the signed intensity of transition
- **Second order element difference model**, $\sum_{i,j=1}^{21} (i - j)^2 RCM(i, j)$ gives an estimate of the transition probability, weighted by the unsigned intensity of transition

TABLE I: Classification accuracies obtained using eleven Ranklet features (section § ??) for different resolutions. RW8, RW16 and RW32 represent features obtained using window sizes 8×8 , 16×16 and 32×32 , respectively. W8 + 16 represent concatenated features of W8 and W16, similar stands for W16 + 32 and W8 + 16 + 32.

Features	NB	1NN	3NN	5NN	QDC	LIN
RW8	0.238	0.150	0.162	0.158	0.609	0.497
RW16	0.286	0.207	0.219	0.230	0.412	0.402
RW32	0.228	0.143	0.161	0.145	0.240	0.250
RW8+16	0.398	0.333	0.340	0.356	0.648	0.594
RW16+32	0.346	0.288	0.288	0.279	0.449	0.387
RW8+16+32	0.420	0.364	0.375	0.389	0.573	0.594

- First order inverse element difference model, $\sum_{i,j=1}^{21} \frac{1}{1+|i-j|} RCM(i, j)$, gives an estimate of the transition probability, weighted by the inverse signed intensity of transition
- Second order inverse element difference model, $\sum_{i,j=1}^{21} \frac{1}{1+(i-j)^2} RCM(i, j)$, gives an estimate of the transition probability, weighted by the inverse unsigned intensity of transition
- Energy distribution of the ranklet CM, $ed_1 = \sum_{i=9}^{13} \sum_{j=9}^{13} RCM(i, j)$ takes values from the certain range - expresses transition energy in particular band.
- Energy distribution of the ranklet CM, $ed_2 = \sum_{i=7}^{15} \sum_{j=7}^{15} RCM(i, j) - ed_1$ takes values from the certain range - expresses energy in particular band.
- Energy distribution of the ranklet CM, $ed_3 = \sum_{i=3}^{19} \sum_{j=3}^{19} RCM(i, j) - ed_1 - ed_2$ takes values from the certain range - expresses energy in particular band.

Since the ranklet coefficients are computed from their relative ranking within an image sub-window - these features are robust to variations of gray-scale. Therefore - different types of grey-scale transformations (linear, gamma,...) do not affect these features. This characteristic can be advantage or disadvantage, depending on the type of texture.

B. Ranklets - Results

Ranklet features calculated using window sizes 8, 16, and 32 have been tested with different classifiers and with the train and test dataset described in introduction, without taking into account subclasses. Obtained accuracies are shown in table I. General conclusion is that quadratic classifier performs better than the others with ranklet features. Accuracies become better as features involving at least two (or more) window sizes are used. This can be treated as extension to multi-resolution, therefore recommendation to use ranklet features together. Best accuracy is obtained using PRTools4.0 quadratic classifier for concatenated features of window size 8 and 16, giving 64.8% accuracy.

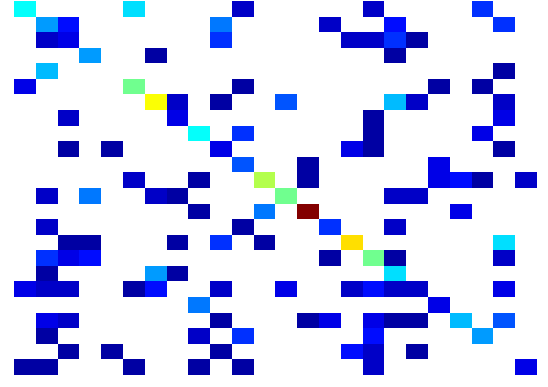


Fig. 6: Confusion matrix for the classification of 24 classes using Naive Bayes and concatenated ranklet features for windows 8, 16 and 32.

It is important to mention computational time performance of ranklet feature extraction. Due to sorting algorithms used, feature extraction for one 576×768 pixel image can last several minutes. To overcome this, Fast algorithm for computation of ranklets was used [5]. MATLAB code for ranklet transform was borrowed from professor F. Smeraldi for academic purposes. Moreover, images were resized to 256×256 size before feature extraction. All of these steps resulted in 66 seconds of execution time needed for one image on 2GHz CPU, which was significant improvement.

Generally ranklets did not perform well with other classifiers apart from linear and quadratic. Naïve Bayes had the best performance with the usage of all three window sizes (42%), having multi-resolution information available. Reason for such performance can be in one of the most important quality of ranklets - robustness to grey-scale transformations. Overall difference in brightness could be useful in classifying given wood textures. For instance, some wood species have similar texture, but one is brighter, or reflecting more light than the other. This can cause misclassification as shown in confusion matrix for the case of RW8+16+32 features and Naïve Bayes classifier (figure 6). Since ranklets do not analyse shapes and morphology, this brightness difference can be missed discriminative characteristic.

It might be useful to mention that there was an attempt to individually make a C MEX file that would speed up the execution time due to many for loops in Matlab. However, the code did not manage to perform significantly faster and the output was not the same as the output of the Matlab code in all the cases, hence the transform itself was carried out using prof. Smeraldi's code.

C. Curvelets

Following the concept of Fourier transform, wavelets use basis functions to transfer signal from one space to another trying to localize signal's frequency and the moment when frequency occurs. Wavelets could,

therefore, be considered as the generalization of Fourier transform - an attempt to localize signal in both time and frequency. One of the disadvantages of this method is "blindness" of classic wavelets to sense direction: edges, geometry. Solution for the problem is usage of direction-sensitive (anisotropic) basis functions. Ridgelet tend to be a good tool for straight lines, however, lines can be curved as well. To cover wider range of discontinuities, we need additional generalization.

Moreover, directional wavelet transforms use basis functions that are sensitive to orientation, so that transformation does localization in orientation, too. Curvelet transform goes even further. The idea is that image or function is represented in different scales (multiscale approach, already widely implemented in image processing or compression). In this case, localization of orientation in orientation depends on scale - anisotropy changes with the scale applied [7]. Furthermore, the whole transform is non-adaptive.

Curvelet transform works similarly as various other wavelet-like transformations. Discrete transform is used to obtain curvelet coefficients. Coefficients have been obtained using MATLAB curvelet toolbox CurveLab¹. For the purpose of extracting texture features, computational speed plays an important role. Therefore, C++ MEX implementation available from CurveLab was used. The texture features are derived from the Discrete Curvelet Transform [7] and the discretization of the continuous curvelet transform that uses "wrapping" algorithm [7].

Several works have focused on texture classification using curvelets and its improvement. Most of the implementations have shown good performance compared to ridgelets or other wavelet families that take into consideration the geometry or the curves [8]. Since the wood texture given contains such curve-like shapes, such feature extraction seemed like an interesting choice. However, most of the implementations of curvelet coefficients were accomplished in medical imaging - computer tomography [8], space imaging, satellite imaging, or on general collection of textures like bricks, clouds, grass, etc. [9], [10]. Generally speaking, the extraction pattern stays similar in all the applications:

- Candes and Donoho introduced a novel multi-scale transform -designed to handle curved singularities. Main idea is to zoom into an image patch, so that curves eventually become "straight". Transformation is applied with different scales and different angle orientations set as parameter of the transform (figure ??). What makes the curvelets special is that - the more we go to finer scales, the more the transform becomes sensitive to angle orientations - therefore establishing good frequency-space localization, better sparsity, less coefficients

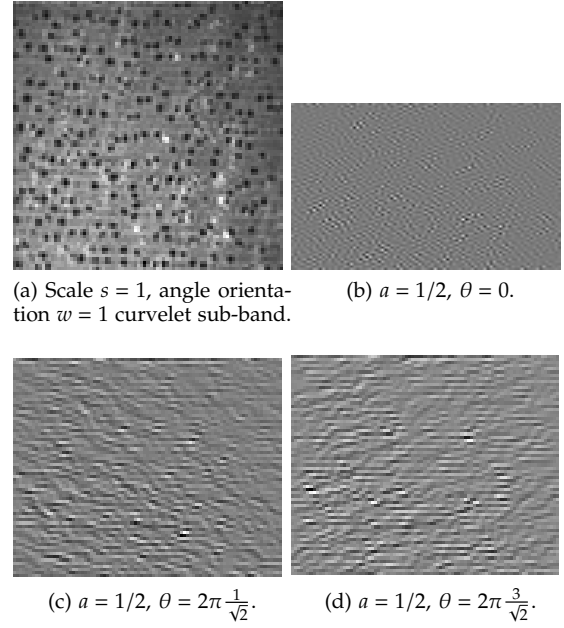


Fig. 7: Curvelet coefficients of different scales (s) and orientations (w) in spatial domain. Many features are extracted from texture characteristics of such images.

for a given accuracy. Sparsity could cause the problem in this application, however a couple of feature-parameters was revealed in [10] that actually adopts to sparsity giving quite discriminative features as outputs.

- Discrete curvelet transform is applied on set of texture images. Discrete transform means - discrete set of scales and angles for each scale. The output of the transform is sub-band decomposition: image is decomposed into sub-band images of curvelet coefficients. Each scale is represented with the group of coefficients - each one for a different discrete angular orientation.
- Images with curvelet coefficients (figure 7) need to be characterized in some way. The final outcome of that process is a feature, a description. Although some papers propose different approach [9], we will follow the classic approach where obtained curvelet coefficient images for different scales and orientations are described with already established co-occurrence matrices features, similarly as it was the case with ranklets. On top of that, additional features, such as parameters of an Generalized Gaussian Distribution (GGD) are calculated - α and β . Inspiration for using such parameters as features comes from the work of Gomez [10]. They proved to be quite useful in wood classification task that was given.

D. Curvelet Features

It is convenient to separate the features into three groups:

¹<http://www.curvelet.org>

- **Curvelet statistical features**
 - Mean (M) - mean value of the coefficients in sub-images (first order statistics)
 - Standard deviation (SD) of the coefficients in sub-images (second order statistics)
- **Curvelet Co-Occurrence features** Co-Occurrence matrix is formed for each sub-band of the Discrete Curvelet Transform (figure 7). As with grey-level co-occurrence matrices, spatial information about the distribution of the values is taken out using following features:
 - Energy and Entropy (EE)
 - Contrast (Con)
 - Cluster Shade (CS)
 - Cluster Prominence (CP)
 - Homogeneity (H)
 - Local Homogeneity (LH)

Formulas that express calculation of these parameters are available in literature [11], and the well known paper of Haralick that introduced co-occurrence matrices.

- **Generalized Gaussian Density Parameters** [10] Each sub-band of the curvelet space (figure 7) is used for estimating the parameters of a statistical model. The role of a statistical model is to capture the distributions in several parameters, similarly as mean and standard deviation fully describe Gaussian distribution. Curvelet coefficients are distributed sparsely - therefore a need for a parameter that would be better at modelling marginal distribution. It is suggested in work of Gomez [10] that the information about the edges is well captured in the moments of Generalized Gaussian Density (GGD). The aim is to calculate α (variance) and β (parameter describing the decreasing rate of the GGD) from the GGD definition $p(x; \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} \exp(-(\frac{x}{\alpha})^\beta)$ where Γ represents already defined mathematical function [10]. Two parameters are estimated by fitting the given data to $p(x; \alpha, \beta)$ distribution using Maximum Likelihood optimization (figure 8).

- α (first GGDcoeff parameter)
- β (second GGDcoeff parameter)

Suggestion (/citeGomez09) was that the textures should be classified by measuring the distance between them (as it is the case with euclidean 1NN or kNN), just that the distance used is “KullbackLeibler”. Definition of the distance is available in paper that proposes the usage of GGD coefficients. After applying it on two GGD coefficients obtained for different scales and angle orientations, to be used in classification, the performance was not better than it was when using ordinary euclidean 1NN - around 40%. However the option to use such distance measure is still available in *myKNN.m*.

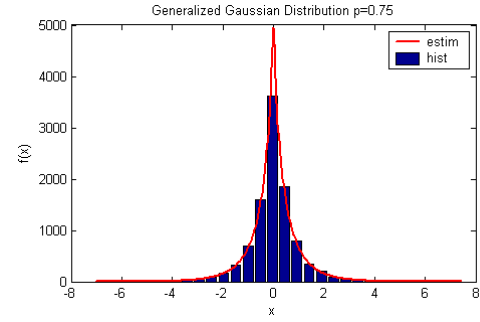


Fig. 8: Histogram of curvelet coefficients from the sub-band. GGD distribution was used as model. Red approximation of blue distribution was plotted using estimated parameters.

To summarize: image is transformed into collection of images (sub-band images) containing curvelet coefficients (figure 7), and features are extracted from each of the sub-images. Hence, amount of features depends on number of sub-images - therefore number of discrete scales and angles. Going deeper into scale or discretizing more the angular direction has as a consequence extremely high number of features.

E. Curvelets - Results

Features were extracted using Discrete Curvelet Transform (“wrapping method”) [7] with

- 3 scales and 16 angles of the second coarsest level
- 5 scales and 16 angles of the second coarsest level

as parameters. Due to computation time costs, each image was resized to 256×256 before the transform was executed. Results were similar for both scales. Table II showcases accuracies obtained for those feature combinations that had the best performance when combined together. Curvelet based features yielded accuracies ranging from 20% up to 66% for combination of 8 features and Linear classifier (table II). Since the results were not as expected, additional feature selection has been done. Total of 162 features that performed the best were reduced using PCA and Fisher mapping. With a simple 1NN classifier applied, accuracy was calculated for different feature reductions. The outcome is improvement of accuracy for certain reductions. As figure 9 shows, maximal accuracy was obtained when reducing to 95 features.

F. Gray-Level Co-Occurrence Matrix (GLCM)

Much has been said about Gray-Level Co-Occurrence Matrices (GLCM). They were often used for texture characterization, however not as often in wood classification ([12], [13]). Some references are available in case of the given dataset ([12]) or some other datasets [14]. Algorithmically, GLCM is a statistical tool, textures are characterized with statistics of different orders. More details on construction of the GLCM, parameters,

TABLE II: Classification accuracies obtained using curvelet transform features (section § IV-D). Transformation was accomplished with 3 scales and 16 angles at the coarse level. Image was resized to 256×256 prior to transformation.

Features selected	NB	1NN	3NN	5NN	QDC	LIN
GGDcoeff	0.373	0.298	0.304	0.292	0.205	0.489
SD	0.251	0.416	0.422	0.406	0.435	0.306
GGDcoeff+M	0.377	0.338	0.315	0.309	0.186	0.530
CORR+SD	0.340	0.416	0.422	0.406	0.338	0.424
M+CORR+LH	0.466	0.191	0.193	0.174	0.068	0.429
SD+Con+CS	0.338	0.470	0.470	0.474	0.056	0.460
EE+SD+CORR+CS	0.476	0.435	0.439	0.435	0.081	0.516
M+SD+Con+CS	0.340	0.484	0.468	0.455	0.052	0.491
GGDcoeff+SD+CORR+LH+CS	0.474	0.460	0.468	0.447	0.118	0.632
M+SD+Con+H+CS	0.360	0.484	0.468	0.455	0.037	0.513
GGDcoeff+EE+M+SD+CORR+CS	0.474	0.453	0.460	0.466	0.106	0.646
GGDcoeff+M+SD+CORR+LH+H+CS	0.474	0.453	0.460	0.466	0.097	0.654
GGDcoeff+M+SD+CORR+LH+CP+H+CS	0.472	0.259	0.271	0.279	0.041	0.663

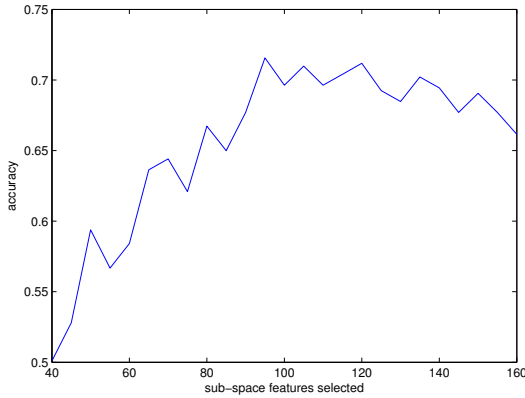


Fig. 9: Total number of 162 features that gave the best performance in curvelet classification were reduced to N features using PCA and LDA (fisher mapping) reduction 1NN classification. For each reduction (x axis), accuracy was plotted (y axis). Finally, the best accuracy (72%) was obtained with 95 features out of 162.

features is available in [15]. With GLCM an assumption is made that the characteristics of the texture are contained in spatial relationship of the pixel grey level values one with another [citekhalid2008design]. Spatial relationship implies probability of the occurrence of same pixel values at particular distance and particular angular direction. For this work, GLCM with distances of $d = 1, 3, 5, 7, \text{and } 9$ have been implemented together with angular directions of $0, 45, 90, \text{and } 135^\circ$. Several values of distances have been used to have some kind of multiresolution. GLCMs for different directions are averaged so that the rotational invariance of 45° is present. One of the issues that was not explored deeper in this work is the possibility of multiresolution - choosing appropriate d to deal with particular texture. Several papers are proposing a solution for this.

Features extracted from Gray level co-occurrence matrix are:

TABLE III: Classification accuracies obtained using combinations of ten available GLCM features calculated for distances $d = 1, 3, 5, 7, 9$ (section § IV-F). Short names represent features concatenated together. Number of features that characterize one image is 45.

Features	NB	QDC	LIN	1NN	3NN
ASM+Corr+Ent+CS+IDM+Var+Con	0.338	0.603	0.677	0.304	0.319
ASM+Corr+CS+IDM+Var+Con	0.346	0.569	0.646	0.304	0.319

- Angular Second Moment (ASM) ([12])
- Contrast (Con)
- Correlation (Corr)([12])
- Entropy (Ent)
- Inverse Difference Moment (IDM)([12])
- Inertia (Ine)([13])
- Local homogeneity (LH) ([13])
- Maximum probability (MP) ([13])
- Cluster Shade (CS)([13])
- Cluster Prominence (CP) ([13])

More details about the formulas that are atypical can be found in cited references. GLCM components are averaged for all directions. Components for different distances are concatenated together (attempt to have multiresolution).

G. GLCM - Results

GLCM showed significant performance with certain selection of features. As previously said, ten types of features were calculated. To sum up those that had the best performance when combined together are shown in table III. Meanings of the short names can be read in section § IV-F).

Naturally, accuracies could be improved by increasing number of used distances, however such acrobatics was not the purpose of this study, but to check possible performance of such features. Possible improvement could be finding a way to optimally estimate the

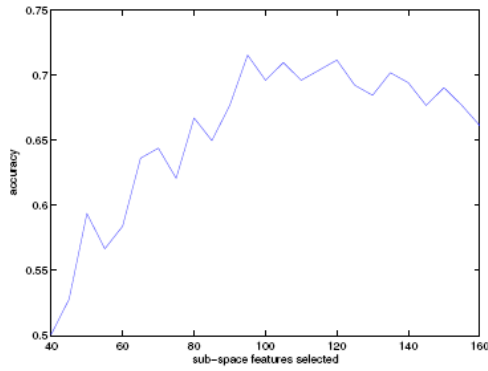


Fig. 10: Using PCA reduction, Fisher mapping and 1NN classification – accuracy of classification using GLCM reaches 72% when reducing the set of features to 95 features at the PCA reduction stage.

d parameter. Moreover, with the application of PCA reduction, LDA reduction and simple 1NN classifier, accuracy improves up to 72% (figure 10).

H. Wavelets

Wavelets are wave-like oscillations that are described by a function referred to as mother wavelet. It has been shown that a signal or image (i.e. 2D signal) can be expressed in terms of the details and approximation derived using dilation and translation of a wavelet. Wavelets have powerful capability of providing a multi-scale analysis of a signal in special-frequencies space. In other terms, what let provide a good approximation of “what is where” in a signal or image. As textures can be characterized with their intrinsic frequencies and the location of each frequency , wavelets are appropriate tools to extract such characteristics from texture. For characterization of a texture, it is decomposed into sub-bands. Each of the sub-bands represents the response of the texture to the high-pass and low- pass filter in certain direction (horizontal, vertical, diagonal) at a certain scale (see figure 11 for illustration). It means each sub band contain information about different frequencies their location and their direction. It is common to decompose wavelets using high pass and low pass filters (Figure 11) and extract features from decomposed sub-bands using some non-linear functions.

These features can express the scale dependant characteristic of texture in that sub-band. In a texture classification paradigm there are several factors one needs to consider when extraction features from texture using wavelet. These factor which affect the distinctiveness of feature space are : choice of mother wavelet, method of decomposition, number of levels of decomposition and the method used for extracting features from sub-bands.

1) *Choice of mother wavelets*: The feature are extracted from the response to certain filters (high pass and low-

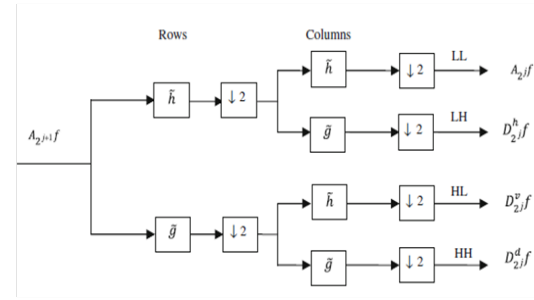


Fig. 11: wavelet decomposition into Sub-bands using high pass and low pass filters.

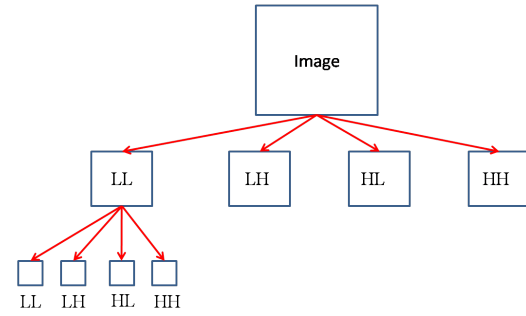


Fig. 12: Pyramid Decomposition.

pass) which are determined by mother wavelet. Thus, the choice of mother wavelet is important as the filter should be in suitable providing a proper localization of frequency elements. As it is not clear how much distinctive the response of a certain wavelet is before evaluation of classifier. In this work various families of wavelet has been used to study the correlation of features features extracted using each one. It has been suggested that the most suitable wavelets for texture classification are “texture matched wavelets”. These wavelet can be derived to maximize the discriminatory power over a training set of textures.

2) *Method of decomposition*: Another factor which needs to be explored is the method of decomposition. The most common way of decomposition is using wavelet pyramid in which one sub band from a higher level is decomposed to form the sub-band of a lower level. An example of such decomposition for 3 levels is shown in Figure 12 . Most commonly the LL sub band is decomposed further. But variant of pyramid decomposition select the sub band with highest energy for further decomposition.

Alternatively, it is possible to decompose all sub-bands further and derive an over complete basis this is referred to as wavelet packet decomposition (see Figure 13). This method provides a higher definition of sub-bands spectrum but it also yields feature redundancy. As this method yields a high feature vector dimension is it common to use feature reduction methods such as PCA and LDA to remove redundant features which do not contribute to classification.

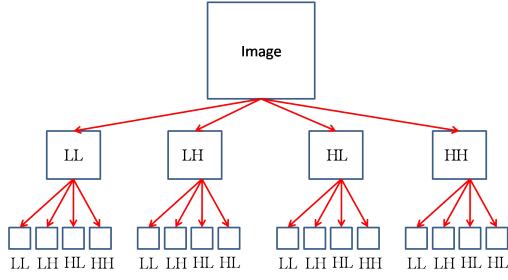


Fig. 13: Wavelet Packet Decomposition.

Norm-2 energy	$e_1 = \frac{1}{N^2} \sum_{k=1}^N C_k ^2$
Norm-1 / Average energy	$e_2 = \frac{1}{N^2} \sum_{k=1}^N C_k $
Standard deviation	$e_3 = \sqrt{\frac{1}{N^2} \sum_{k=1}^N (C_k - \mu)^2}$
Average residual	$e_4 = \sum_{k=1}^N C_k - \mu $
Entropy	$e_5 = -\frac{1}{N^2} \sum_{k=1}^N C_k ^2 \log C_k ^2$

Fig. 14: Feature generation from the wavelet coefficients

3) *Feature computation*: As we are dealing with a classification problem we need one feature vector per texture. There are several method for extracting features based on wavelet decomposition. The most common is using energy based features. For example norm Energy feature represents the average magnitude of the response of filter on that particular sub-band and direction. Several common energy based features are given below .

As above feature are derived from sub-bands which correspond to different directions (horizontal-vertical , diagonal) they are not rotation invariant. It has been shown that rotation invariance can be achieved by averaging the features in different direction (averaging features of HL,LH, HH sub bands). Other than single value representation of sub-band (energy based features), histograms can be used to measure the occurrence of different responses to the filter at each scale and orientation. For even richer and more expressive representation of response of wavelet at certain scale and orientation we can use co-occurrence matrix of sub-bands. This method yield a feature vector for each sub-band using different statistics on co-occurrence matrix (see GLCM section for more information). Such feature vector is much h more representative of sub-band thus when this feature vector are concatenate they provide a high dimensional but more descriptive of feature space for the texture. One should note that this process is highly time consuming and yield a bulky feature vector. Alternative to feature computation it is possible to use the value of the co-efficient from each image as feature vectors. For this purpose

TABLE IV: Summary of Results of individual Feature extraction methods

Feature extraction	KNN-1	QDC
Bior F1	37.71	52.224
Bior F2	15.47	5.802
Bior F3	40.81	28.433
Coif F1	39.26	47.775
Coif F2	15.86	5.802
Coif F3	40.61	10.251
Db F1	33.46	49.129
Db F2	15.66	1.934
Db F3	40.81	27.079
Haar F1	40.23	51.512
WPT Bior	28.67	4.835
WPT coif4	32.8	11.605
WPT db8	36.75	7.54
WPT sym8	33.07	12.37

In order to find the best features for texture representation different method have been implemented. Various mother wavelets families (e.g. Haar, Daubechies, Bior, Coifet, Symlet) have been tried out to see if there is a correlation between features derived from different mother wavelets. In each families various versions have been used (e.g. Daubechies 6-8, Bior 2.6 & Bior 3.7 etc.) Both normal and rotation invariant features has been implemented. Wavelet packet and wavelet pyramid decomposition have been studies with different levels of decomposition. The aim is to span the feature space as much as possible then select the set of features which can complement each other for better classification. it is important to study the correlation of features and remove the features that are highly correlated. After the feature extraction correlation matrix of all features is computed. The correlation matrix shows that the features extracted from same wavelet families are highly correlated(e.g.Bior 3.7 & Bior 2.5). Further, There exist rather high correlation between wavelet families such as Coifet and Bior and Daubechies. In order to select best set of feature subspace feature selection method has been used which is explained in following.

4) *Wavelet Experiments*: As the first experiment we evaluate the performance of each feature extraction method. Feature extraction method are grouped based on the feature computation method and wavelet families. F1 feature are normal energy features extracted from different scales. F2 Features are rotation invariant features and F3 are the maximum coefficients at a each level of decomposition. Several classifiers such as Quadratic , K-NN. The results show that different feature extraction methods have variable performances on different classifiers. The best performance is achieved using quadratic classifiers and Bior-F2 that is 52%. Table below shows the summary of the result this experiment.

Using the table IV we can have the first clue of which feature extraction methods is more suitable. However one should note that even the method with low perfor-

mance can have provide complementary information when combined with other methods. That is why we do the feature selection step. In this step all feature are concatenated and the PCA-LDA projection is made to derive the most discriminative features. As the next experiment we concatenate all features above together and perform PCA projection. This experiment shows that the result can be improved to 55% with 100 eigenvectors (i.e. a feature space dimension reduced from 2729 to 100). However, the performance plateaus and does not increase with higher feature dimensions. As the next Experiment we use the PCA+LDA projection and using 1-NN classifier the performance increases to 70%(200 eigenvectors). combining the results with morphological features and curvelet features and a subsequent PCA+LDA projection a performance of 80% was reached using 1-NN classifier. Overall from the experiment we can conclude that while being quick to compute (0.05s for F1) wavelet feature extraction alone is not a appropriate method for this texture classification problem. This is due to the fact that several classes of features have similar responses to wavelet families. however as some texture classes are classified using this method it is possible that the complementary information can be provided when combined with other features.

I. Local Binary Pattern

1) *State-of-the-art*: Local Binary Pattern (LBP) is a texture analysis operator that has been widely used for feature extraction and texture classification in computer vision, due to its simplicity and its computational efficiency [?]. It was first described in 1996 by Ojala et al. [?] and has since then enjoyed a wide popularity. Based on the binary comparison of a centre pixel value against its neighbours, the original algorithm used a 3x3 neighbourhood, thus giving 8 neighbours $P_8 = p_1, p_2 \dots p_8$ for each studied pixel p_c . Then, the following array of values is generated:

$$t(s(p_1 - p_c), s(p_2 - p_c), \dots, s(p_8 - p_c)) \quad (3)$$

where

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

This definition makes LBP invariant to monotonic grayscale transformations, as it uses the sign of change between the considered pixel and its neighbours and not the values themselves [1]. Each binary array is then converted to decimal notation by choosing a starting pixel and weighting.

$$p_c^{new} = \sum_{n=1}^N s(p_n - p_c) 2^{n-1}$$

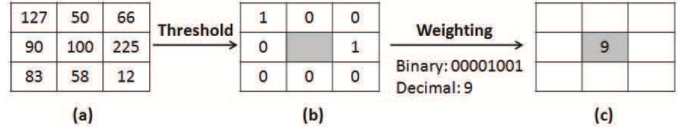


Fig. 15: Thresholding and weighting in LBP

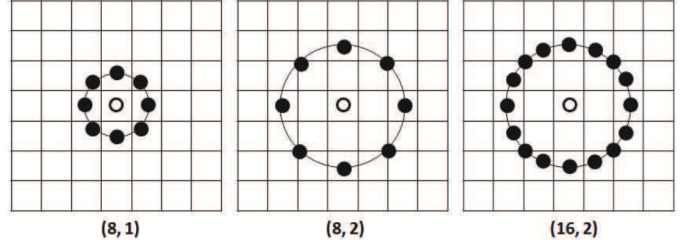


Fig. 16: Neighbourhoods of different number and radius

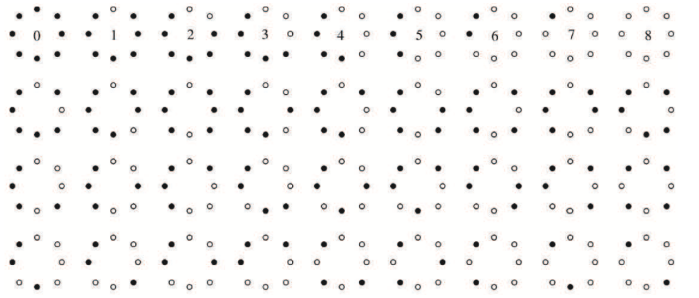


Fig. 17: LBP patterns

Finally, the obtained value is assigned to the central pixel that generated the array. Figure 15 from [2] illustrates the described process. A 3x3 neighbourhood is used in (a), and the resulting thresholding with the central pixel results in (b). Weighting the obtained array, starting from the right pixel in counterclockwise direction, gives the new value of the central pixel (c) as shown in figure 15.

Multiresolution can be attained with LBP by using neighbourhoods (N) of different sizes and interpolating pixel coordinates in order to allow any radius (R) [16]. The notation (N, R) then summarises the adopted approach, with $LBP(8, 1)$ being similar to the original LBP. Figure 16 illustrates different (N, R) combinations:

2) *Rotation invariance*: The original LBP can be made rotation invariant by choosing as a starting point in the circular neighbourhood that element yielding the binary array of minimum possible value. This approach needs many descriptors [2], as exemplified in Figure 17 for a neighbourhood of 8, which can be expressed as $LBP_{8,R}^{ri}$, with ri meaning rotation invariance. These detectors correspond to image patterns like bright and dark spots (0 and 8, respectively) and edges (4).

3) *Uniform patterns*: Introduced in [17], the use of uniform patterns applies the rotation invariance de-

scribed before solving the storage issue by means of reducing the number of patterns to be used. Instead of patterns, spatial transitions (bitwise 0/1 changes) [?] between binary values are applied, with categorization into classes for patterns with up to two transitions, and assignation to a miscellaneous class for those having more. As noted by [?] a properly selected set of patterns forms an efficient texture description improving the classification rates of the whole LBP histogram. The notation used here is $LBP_{N,R}^{riu2}$ with $N+2$ pattern classes for N neighbours [16]:

$$LBP_{8,R}^{riu2} = \begin{cases} \sum_{n=1}^N s(p_n - p_c), & \text{if } U(LBP_{N,R}) \geq 2 \\ N+1, & \text{otherwise} \end{cases}$$

where

$$U(LBP_{N,R}) = |s(p_N - p_c) - s(p_1 - p_c)| + \sum_{n=2}^N |s(p_n - p_c) - s(p_{n-1} - p_c)|$$

4) *Adaptive LBP*: Adaptive LBP [?] also incorporates directional statistical features to improve the efficiency of LBP. In particular, mean and standard deviation of the local absolute differences are applied, together with least square estimation to minimize the local difference. Given an $n \times m$ image, a central pixel g_c , and its P neighbours g_p , these present an orientation $2\pi i * p/P$ with radius R , where p is the orientation. The first and second order directional statistics of $|g_c - g_p|$ along orientation $2\pi p/P$ are μ_p and σ_p , and thus the mean vector and the standard deviation vectors are $\bar{\mu} = [\mu_0, \mu_2, \dots, \mu_{p-1}]$ and $\bar{\sigma} = [\sigma_0, \sigma_1, \dots, \sigma_{p-1}]$, where:

$$\mu_p = \sum_{i=1}^N \sum_{j=1}^M |g_c(i, j) - g_p(i, j)| / (M * N)$$

$$\sigma_p = \sqrt{\sum_{i=1}^N \sum_{j=1}^M (|g_c(i, j) - g_p(i, j)| - \mu_p)^2 / (M * N)}$$

These directional statistical features carry useful information for texture discrimination, and thus can be used to improve the classification results. Figure 18 illustrates the shift in the obtained vectors corresponding to a shift in the rotation of the texture image in $LBP(8, 1)$ [?].

Based on these vectors, the ALBP algorithm is developed by introducing a parameter w_p so that the directional difference $|g_c - w_p * g_p|$ can be minimized, which is solved using a least square estimation technique where \bar{g}_c is a column vector containing all g_c pixels, and \bar{g}_p the corresponding one for all g_p pixels.

$$w_p = \operatorname{argmin} \left(\sum_{i=1}^N \sum_{j=1}^M |g_c(i, j) - w_p g_p(i, j)|^2 \right)$$

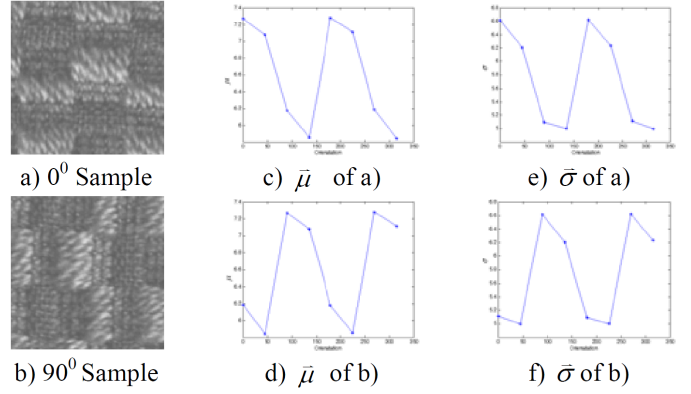


Fig. 18: Rotation of texture image and vectors

TABLE V: Global statistical texture measures

$m = \sum_{i=0}^{L-1} z_i p(z_i)$	$\sigma = \sqrt{\mu_2(z)}$	$R = 1 - (1/1 + \sigma^2)$
$\mu_3 = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i)$	$U = \sum_{i=0}^{L-1} p^2(z_i)$	$e = -\sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i)$

$$w_p = (\bar{g}_p^T \bar{g}_c) / (\bar{g}_p^T \bar{g}_p)$$

Finally, ALBP is defined as:

$$ALBP_{P,R} = \sum_{p=0}^{P-1} s(g_p * w_p - g_c) 2^p$$

5) *Proposed method: statistical measure incorporation to LBP*: A variant of LBP has been devised in an attempt to improve the accuracy of the attained classifications, with successful results. This algorithm implements the multiresolution rotation invariant uniform pattern version of LBP in combination with the use of a selection of different extracted global statistical texture measures, the latter as described in [?], and listed in table V.

In this way, different global measures are used to complement the locality of LBP, improving the classification obtained with other versions of LBP. As grey levels are important in the application of image moments, illumination correction has been undertaken before the extraction. The first measures are the mean, a measure of the average intensity, and the standard deviation, a measure of average contrast. Then smoothness is calculated, and the third moment, giving the skewness of the image histogram. Uniformity can also be estimated, and finally entropy, a measure of randomness. All of these have been calculated and implemented according to the formulation shown in Table V.

6) *Other developments*: Due to its efficiency, simplicity, multiresolution possibilities and both grayscale and rotation invariance, LBP has been used in a variety of applications, such as texture classification, product inspection and face analysis [2]. In [18], LBP is extended to colour images by computation of texture features for every band and statistical measures application. In [?],

TABLE VI: Computing times, in seconds, for each configuration

LBP 8	LBP 16	LBP 24	LBP 8-16	LBP 8-24	LBP 16-24	ALBP	SLBP
0.2207	0.4282	0.6872	0.6334	0.6354	0.8488	0.1901	1.0896

TABLE VII: Error rates of classification for each configuration

	knn-1	knn-3	knn-5	knn-7	knn-9	knn-11	Bayes-1	Parzen	Bayes-2
LBP 8	0.547	0.540	0.544	0.542	0.545	0.565	0.569	0.538	0.445
LBP 16	0.410	0.419	0.417	0.415	0.425	0.413	0.292	0.396	0.348
LBP 24	0.364	0.385	0.406	0.412	0.429	0.435	0.366	0.379	0.362
LBP 8-16	0.406	0.429	0.458	0.449	0.466	0.472	0.304	0.431	0.429
LBP 8-24	0.294	0.346	0.335	0.364	0.368	0.379	0.292	0.302	0.458
LBP 16-24	0.400	0.462	0.499	0.507	0.511	0.536	0.219	0.458	0.687
ALBP	0.580	0.598	0.603	0.613	0.607	0.629	0.534	0.574	0.942
SLBP	0.279	0.296	0.311	0.323	0.340	0.348	0.164	0.277	0.870

an LBP with Fourier features is presented, computing image descriptors from discrete Fourier transforms of LBP histograms, with good results. Monogenic-LBP, presented by [?], integrates LBP with two rotation invariant measures: local phase and local surface type computed by the first and second order Riesz transforms, improving the results of other state-of-the-art methods. Finally, [?] present a hybrid scheme, with global rotation invariant matching and local variant LBP texture features. The methods mentioned here have not been implemented in the present project, and remain as options to be explored in the future.

7) *Implementation*: The implementation here undertaken has made use of different versions of the LBP algorithm. The standard version has been used, with 8, 16, and 24 neighbourhoods (LBP 8, LBP 16, LBP 24), and also combinations of them have been applied to implement a multiresolution approach by making use of different radiuses (LBP 8-16, LBP 8-24, LBP 16-24). The uniform patterns method has been applied, as described in the previous subsection (*riu2*), as well as the adaptive LBP algorithm (ALBP), together with the generated statistical LBP method (SLBP) also presented in 5.3. Running times required by each feature extraction method to extract the features of one image (balau1001.bmp, previously converted to grayscale), including the whole LBP process, from image load to storage of results, are shown in Table VII :

The generated code implements the following process, for each algorithm configuration:

- extraction of features for all images in the training set
- generation of classifiers from the training set feature data
- extraction of features for all images in the testing set
- application of the classifiers to the testing set feature data
- estimation of classification error

Code has been organised following the scheme above. The extraction of features uses the LBP code

v.0.3.2 by Heikkilä and Ahonen to generate 8, 16 and 24 neighbourhood maps, using the *riu2* mapping type for uniform rotation-invariant LBP. Different configurations have been studied and tested, as explained before, using multiresolution, a novel algorithm, and the ALBP version by Guo et al. [?] v.1.0. In the proposed algorithm, a set of statistical measures have been tested in combination with the best configuration for multiresolution LBP, after applying illumination correction. These measures have been extracted as suggested by [?], and comprise, as explained, the following measures: mean, standard deviation, smoothness, third moment, uniformity, and entropy. Results were, as expected, improved with this approach. Indeed, the best classification rates of all those attempted have been attained by this method, as it is shown by error rates computation.

Once features are extracted for each configuration, values are labelled and stored in .mat files, for both normalised and non-normalised versions. The classifiers are trained using the .mat files corresponding to the training images, and then used to classify the data in the .mat files corresponding to the testing images for each method implemented. Finally, error rates are calculated. Applied classification techniques include k-nearest neighbour (knn) classifiers, Bayes-1, Bayes-2 and Parzen, which are not described here due to space limitations. The use of the new SLBP with most of these classifiers has proven to be more accurate than most of the rest of current-day approaches.

8) *Results*: Error rates have been computed for each of the LBP feature extraction methods in combination with different classifiers. Table VII shows the obtained results:

As it can be seen in the results of table VII the novel algorithm here introduced presents very good results in comparison to the rest of present-day studied methods, outperforming most of them in combination with most of the tested classifiers. Indeed, SLBP seems to offer the best configuration among the LBP-based algorithms for the classification of the wood textures in the given

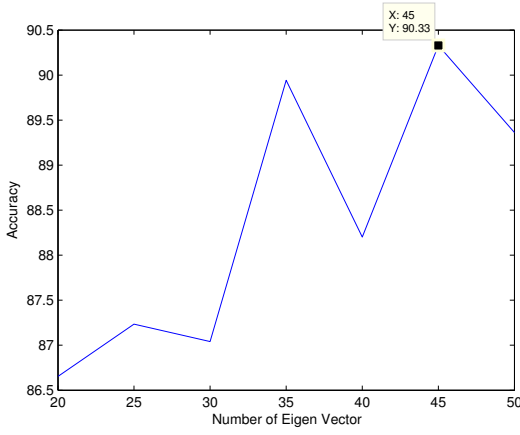


Fig. 19: Performance of LBP. This plot shows the result of feature extraction. The features were reduced to N features using PCA and LDA (fisher mapping) reduction and classified with 1NN.

dataset. Two factors are taken into account to reach this conclusion, both the processing speed for training and feature extraction, and the accuracy of the results. It must be noted, however, that the time needed by this method is slightly superior to the rest, of around 1 second. In terms of accuracy, on the other hand, it outperforms all other methods when applied with all the tested k-nearest neighbour algorithms, and with Bayes-1 and Parzen. Only with Bayes-2 the error rate does not decrease with the implemented method. As it can be seen, error rates are as low as 0.16 with Bayes-2 and around 0.77 with Parzen and some of the k-nearest neighbours. In conclusion, the proposed algorithm seems to offer a good compromise between timing and accuracy, which may be further investigated in the future.

The use of a feature reduction with PCA and LDA (fisher mapping) reduction and final classification with 1NN raised the performance up to 90% (see figure 19). The plot oscillates, which was not expected. This however, might be a result of overfitting to the test set. Normally it would be expected that the classifier becomes better, the more features are added and reaches one peak and finally the performance declines from there.

J. Angle Based Features

The observation that different classes of wood show a very different angle histogram (compare for instance figure 3b with 3d), led to the implementation of descriptors of this histograms. Based on inspection of different histograms five simple descriptors were chosen.

Figure 20 summarizes the extracted features. First a haar like scheme, inspired from the famous Viola and Jones detector [19], is used to compute the differences between different regions of the histogram. To compute this features, first sums between the regions A, B, C of

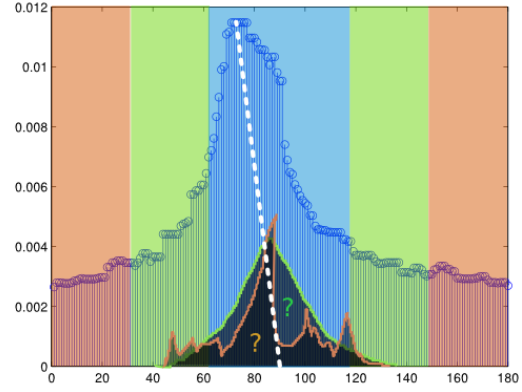


Fig. 20: Features extracted from an angle histogram at the example of keledang2007 (see fig. 3a). First all the values within the equal coloured regions are summed and the permute differences of these sums are taken as first features. Additionally kurtosis (variance from flat histogram or peaky shape) and skewness (represented by white line) are computed as features.

the same colour shown in figure 20. The orange region, for instance, sums the bins from 1 to 30 and 91 to 180. The three different summarized regions are now compared with each other as follows

$$\begin{aligned} \text{feature}_A &= \sum A + \sum B - \sum C \\ \text{feature}_B &= \sum A - \sum B + \sum C \\ \text{feature}_C &= \sum A - \sum B - \sum C \end{aligned} \quad (4)$$

To make the description more complete additionally the skewness ($\frac{\mu_3}{\sigma^3}$) and kurtosis ($\frac{\mu_4}{\sigma^4} - 3$) are calculated, where μ_n is the n^{th} central moment and σ the standard deviation and used as features 4 and 5.

K. Orientation Based Segment Projection (OBSP)

The orientation based segment projection was designed to create a simple descriptor of the wood, based on its main features observed by human perception. These are the lines, which vary in thickness, intensity, frequency of repetition and straightness, further there are holes which can appear black or bright and vary in amount, size, regularity, shape and connectivity. Finally the space between can be more rough or flat.

In this approach several properties of the wood are described together. The segments of an image are processed by simply counting the amount of pixel of the picked segment on a ray that is laid over the image. By shifting this ray over the image, a kind of histogram is created, which is then described by its statistical values.

For computational reasons, the implementation provided projects the cluster values only horizontal, vertical and in both diagonal directions. However, it seems that with this four angles the most of the properties of the sample image are extracted.

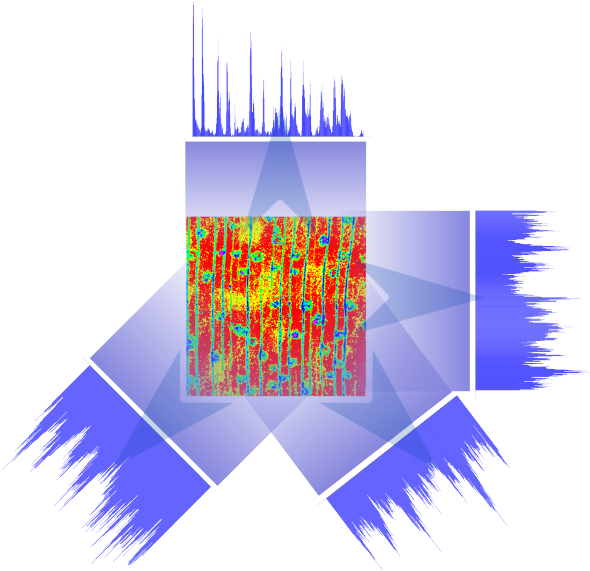


Fig. 21: Projection (and summation) of the value of the blue cluster. The histogram like projections are then bases of statistical feature extraction.

To build the final descriptor the following entropy and sparsity are extracted from each of the generated histograms as ninth feature the normalized ratio of each cluster is taken. While the entropy is calculated with the usual formula, the sparsity checks how many elements in the histogram that lay above a adaptive threshold have neighbours that fall below the threshold. This measure will therefore give the highest value when the histogram values alternate with a high frequency.

No exhaustive research has been done, but many more statical measurements could be though of to be calculated from the histograms.

This method relies heavily on a either (visually) good or (intra class) distinctive clustering and a very good orientation adjustment. Especially if the lines are an important feature the proper angle alignment is very necessary to preserve the peakiness of the histograms of this cluster.

The implementation takes with clustering, but without necessary illumination correction and orientation detection about 1.25 seconds on a 2.13GHz computer. Even if the code is written in Matlab, except with the use of compiler options, it is not realistic to improve the speed much more, since as much vectorization as possible was used and the main computations fall on the rotation of the image.

L. Morphological Spectrum

At the beginning of the project, morphological spectra [20] were considered as feature extraction method. Morphological spectra are an implementation of a discrete convolution of a signal with itself. Doing so, valuable shape features can be extracted. Even if not correct at the end, primary it was assumed that a

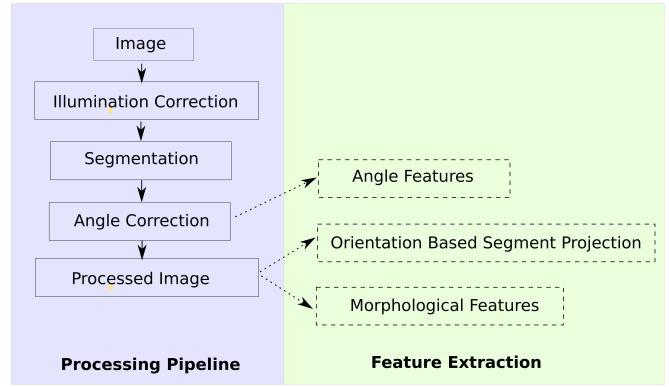


Fig. 22: Workflow of the feature extraction of the angle features, orientation based segmentation features and morphological features.

segmentation is needed to generate a morphological spectrum, which led us solve the segmentation problem first. Another reason, why the decision was made to turn first to segmentation was, that the wood holes could be used as important feature, following [21], where morphological spectra are used to classify cells that can be hardly distinct even by a trained human. Over the segmentation task and new opportunities the morphological spectra have been forgotten for a while.

However, at the end phase of the project the idea was followed again, but without time for an exhaustive literature review. One important paper seems to be [22], which can be used for further exploration of this topic.

Even if a true realization was not possible, a small attempt has been made, by simply generating features from the dilation of the clusters, obtained by k-means clustering with different shapes and sizes of kernels. This turned out to give 10 % more accuracy with Naive Bayes at the time of union with the angle based features and the features from the orientation based segment projection. On the other hand this different simple features show a very high correlation.

It can be clearly stated that it is highly advised to do research in this field, it seems to be very promising and it is used in many fields successfully.

M. Combination of Angle Based, OBSP and morphological features

The final feature was combined from the angle based feature extraction, orientation based segment projection and the morphological features. It gave with naïve bayes classifier about 63%, after processing with PCA and LDA and the use of the first nearest neighbour classifier 82% (see figure 23 on page 19).

N. Overview of the Used Features

This section is intended to give an overview of the feature extractors we experimented with. The features are summarized in table VIII, but in general it can be said that complex feature extractors such as wavelets,

TABLE VIII: Overview over the used feature extractors, summarizing their pros and cons that should be considered in a future implementation

Feature Extractor	Speed	Pros	Contras
Curvelets	5-15 sec	<ul style="list-style-type: none"> ✓ emulate 2d curves well ✓ multi-scale ✓ sensitive to edge direction ✓ take into account the geometry 	<ul style="list-style-type: none"> ✗ number of features can be quite high ✗ features amount depends on number of scales & angles ✗ complex to implement the transform ✗ long computation time ✗ computation time increases with the size of the image
Ranklets	60 sec	<ul style="list-style-type: none"> ✓ illumination invariant ✓ 45° rotation invariant ✓ brightness transform invariance 	<ul style="list-style-type: none"> ✗ extremely time consuming ✗ multi-resolution is not inherited ✗ do not give information about shape
GLCM	2-5 sec	<ul style="list-style-type: none"> ✓ extracts spatial relationships ✓ acceptable computation time ✓ distinctive accuracy ✓ easy for implementation ✓ has a certain degree of rotational invariance 	<ul style="list-style-type: none"> ✗ do not give information about shape ✗ multi-resolution is not inherited
Wavelets	0.5 - 1 sec	<ul style="list-style-type: none"> ✓ Multi scale & direction analysis ✓ Fast feature computation ✓ Scale invariant(not used) ✓ Rotation invariant 	<ul style="list-style-type: none"> ✗ Low discriminatory power ✗ No guideline for setting Mother wavelet ✗ No guideline for setting N-levels ✗ many parameters
LBP	0.2 - 1 sec	<ul style="list-style-type: none"> ✓ easy to implement ✓ very fast 	
Angle Features	1.2 sec	<ul style="list-style-type: none"> ✓ integrated in angle detection ✓ brightness invariant ✓ scale invariant ✓ few features ✓ histogram descriptors extendable ✓ easy to implement ✓ no parameters (except for segmentation) 	<ul style="list-style-type: none"> ✗ relies on segmentation result ✗ at current state not very discriminant
OBSP features	2 sec	<ul style="list-style-type: none"> ✓ very fast ✓ few optimized features ✓ rotation invariant ✓ histogram descriptors extendible ✓ no parameters (except for segmentation) 	<ul style="list-style-type: none"> ✗ relies on segmentation result ✗ only images center used ✗ not very scale invariant
Morphological features	0.4 sec	<ul style="list-style-type: none"> ✓ very fast ✓ easy to implement ✓ rotation invariant 	<ul style="list-style-type: none"> ✗ not very descriptive alone ✗ not very scale invariant ✗ does not extend by use of different masks
Gabor Filter	1 sec	<ul style="list-style-type: none"> ✓ ✓ 	<ul style="list-style-type: none"> ✗ ✗

curvelets and ranklets did not perform as well on this dataset, then more simple approaches. Especially the combination of LBP, angle features, OBSP features and morphological features perform very well and can be computed together in less then 5 seconds, where the later three show a low correlation and complement each other quite well. However, they can improve the very good performance of LPB only within a few per cents.

Saying this it has still to kept in mind, that the dataset is very special and the low inner-class variance makes it very difficult to draw final conclusions.

V. FEATURE SELECTION

Feature selection is applied mainly for two purposes, First: to reduce the diminution of data: as a

feature space with high dimensionality may contain high amount of redundant features which do not contribute to the classification. In fact high dimension of features lead to higher computation time and in some cases more complex decision boundaries. In such case classifier will have higher error rate. Second reason for feature selection is not only to reduce dimension for reasons mentioned previously but also to find features which can provide a more separable feature sub-space. An optimum feature space is the one which the samples belonging to same class have low scatter and different classes are as far away as possible from each other. In other term feature space has la maximum ratio of between class scatter and within class scatter for the whole data. Several approaches can be taken for

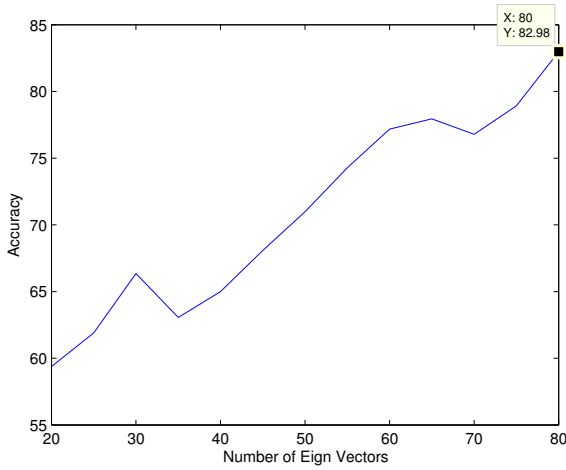


Fig. 23: Combination of angle based features with the features from the orientation based segment projection and the morphological features. This plot shows the result of feature extraction. The features were reduced to N features using PCA and LDA (fisher mapping) reduction and classified with 1NN.



Fig. 24: Confusion matrix of the combined features with the naïve bayes classifier. It shows that some classes have a weaker classification performance then others, but that from all classes at least some samples are classified correctly. However it also shows that only class 2 and 20 are confused regularly.

selection of suitable features for classification.

A. Naive Approach

As the First and the most trivial, it is possible to study the performance of individual set of features and discard those which perform poorly. However, this approach does not take in to account the fact that even features with low performance can provide complement information if combined correctly. Further

it is possible that selected features with individual high performance have high correlation thus simple concatenation of these features can lead to performance loss.

B. Correlation

The second approach is to analyze the correlation between features and remove features that are highly correlated. The problem with this approach is that it is time consuming, especially if the dimension of feature vector is of high. For example wavelets and curvelet feature extraction method produce a high dimensional feature vector (approx 2700 and 7000 receptively). Thus computation of correlation matrix of the combined feature vector is computationally expensive and the analysis is laborious.

C. Principal Component Analysis (PCA)

Another way to remove the correlation between is using Principal Component Analysis (PCA). This method finds the linear combination of features that can be used to express the feature space in a lower dimension. The dimension of feature vector after projection is dependent on the number of eigenvectors of the covariance matrix of the original space used. Eigenvectors associated with highest Eigen-value represent the feature with highest variability. The eigenvectors are sorted with respect to their eigenvalues in a descending order and the first N eigenvectors are used as transformation matrix to project feature space in to the reduced sub space with no correlation.

In practice the optimum number of eigenvectors is found empirically. We start from a specific number of eigenvectors (minimum dimension) and increase the number of eigenvectors selected and evaluate the performance of projected feature space. we plot the performance over different number of features. An example of these performance plot be seen in figure 25. If the plot start to plateau after a certain number of eigenvectors it means that the feature projection is stable and the this number of features can be used reliably for classification. on the other hand if the performance oscillates for different number of eigenvectors this means that selecting the number of eigenvector that give the best performance will not reflect the actual capability of the system. But rather the system is over fitted towards this specific test set.

Moreover, it should be noted that PCA has the issue of using the covariance matrix to compute the basis(i.e. transformation matrix for projection to sub space). Thus the basis derived is dependent on variability of data and that the subspace derived is not necessarily optimal for classification. To achieve a subspace which can separated the data in more suitable way. Linear Discriminant Analysis (LDA) can be used.

D. Linear Discriminant Analysis (LDA)

LDA attempts to derive the subspace that maximizes the ratio between class scatter and within class scatter, thus deriving a subspace that can separate features of different classes more optimally. For Derivation of LDA subspace we need to find $C-1$ eigenvectors of $S_w^{-1}S_b$ where C is the number of classes. This computation is not possible if the within class scatter matrix (S_w) is not invertible (i.e. it is rank deficient). In order to assure that (S_w) is invertible we should project data in PCA subspace first. With projection of features into PCA and LDA subspace respectively we can derive the subspace which is suggested by many studies to be optimal for classification. In order to test this hypothesis several experiments have been carried out.

The problem with PCA-LDA projection is that PCA is not compatible with LDAs Dimension reduction criteria thus the dimensions which may be important for better class separation might be discarded due to PCA projection. The solution to this problem is given by Direct LDA (DLDA). In this approach the null space of (S_b) – that does not contain any useful information for classification – is discarded and the null space of (S_w) – that contains critical discriminative information – is persevered. DLDA achieves this by first diagonalizing (S_b) first and then (S_w) that is the reverse order of the traditional LDA. DLDA is particularly helpful when the of (S_w) is singular (which is the case for most of our feature extraction methods). For implementation of DLDA and guides on techniques for dealing with high dimensional of (S_b) and (S_w) refer to [23].

VI. CLASSIFIERS

At the end of the classification pipeline, after preprocessing, feature extraction, feature selection and feature extraction stands the classification itself. As stated in the very beginning of the report, the variety of the dataset is very low and the use of very adaptive classifiers was considered as very useful. The reason is, that a tuning of classifiers as well as an intensive feature selection might lead to a good result with the provided test set, but at the cost of being overfitted to that very dataset. Cross validation is not applicable as well, because of the danger to test on data which the system was trained with, as stated in the section ?? Dataset.

Therefore we did not concentrate our research on finding an optimal classifier, rather then using simple settings to test the potential of the extracted features. The following sections will introduce briefly into the used classifiers and their advantages.

A. Naïve Bayes

One of the most simple classifiers is the naïve bayes classifier. It is based on Bayes probability law and relies on a statistical independence of the features and needs a inter- class variance above zero. However classifier is

very easy to use, has multi class classification abilities and is very fast in training and testing and does not need any tuning parameters, making it to an ideal testing classifier during the development of a feature extraction. Therefore the preliminary results of the single feature extraction methods, presented above, were calculated with the use of the naïve bayes classifier. In the final classification set-up it was replaced by more sophisticated mechanisms as described below.

The used naïve bayes was the one of the statistical toolbox of Matlab.

B. k-Nearest-Neighbour

The k-nearest-neighbour classifier is again a very simple classifier, that searches for a given test sample for the training sample with the closest distance over the complete feature space. The class is then determined with the class label that is assigned to the majority of the k closest training samples. It is therefore without modification able to do a multi class classification.

With its simplicity some disadvantages appear. The biggest disadvantages are that first the classification process takes longer with increasing feature space and with the amount of different samples. The second drawback is that it suffers much from the curse of dimensionality.

Despite its simplicity, it was used in the final classification set-up, which reduced the amount of features to 24, making the feature space suitable for a 1-nearest-neighbour classification.

As for the naïve bayes the Matlab implementation was used.

C. Support Vector Machine

Support vector machines (SVM) are a very powerful classification tool, natively developed to solve two-class problems. A SVM maps the feature space into higher dimensions until it is possible to separate the space with a specified kernel function. Typically SVM uses a linear function to separate the space, but also other functions are possible. Very common kernel functions are also the non-linear radial functions.

Despite its powerfulness, the SVM is a very tricky classifier. From previous project we knew that the SVM depends heavily on the type of feature vectors it is trained on. In a bag-of-words approach with a very high dimensional feature vector for instance, the trained SVM gave much better results, if each feature was only binary (showing either 0 or 1) instead of counted integers.

Another problem of SVMs are their limitation to two-class problems. In order to solve a multi-class problem it is necessary to adopt the standard SVM with an one-against-one or an one-against-all scheme.

Even if there was no focus onto SVM in this work, several libraries have been tested. The PRtools toolbox

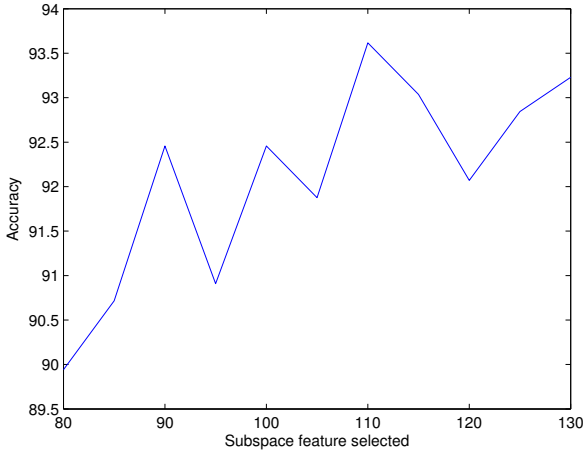


Fig. 25: Combination of angle based features with the features from the orientation based segment projection and the morphological features. This plot shows the result of feature extraction. The features were reduced to N features using PCA and LDA (fisher mapping) reduction and classified with 1NN.

comes with a multi-class SVM that was used and also a simple interface for SVMperf has been programmed and adopted with an one-against-all scheme. Both versions gave slightly better results than the simple naïve bayes and k-nearest-neighbour classifiers, but were still far away from the final result. Most probably due to a not very good feature post-processing and the not very extensive tuning attempts.

VII. FINAL SYSTEM

Based on their good basic performance and their computational effective generation, LBP and the combined features from the angle based segmentation, OBSP features and the morphological features were combined to one feature vector. As it can be seen in figure 25, after feature reduction to 109 features, this combination is slightly better (with 94%) than LBP alone which gave a maximum classification result of about 90% (compare figure 19). This curve shows also an not expected oscillation, as the curve for LBP does. As for LBP we think, that this is an result of a kind of overfitting of the classifier to the test data.

This result can be achieved in below 5 seconds on a single image, a speed that seems to be applicable in a real world application, if the classification result turns out to be as good as here on other test images.

VIII. FUTURE WORK

During the work on this dataset many approaches were experimented with, while constantly ideas arose that could not be followed, because of time or competency constraints. We would like to propose this ideas for a future work as starting point for later projects.

A. Database

Very important would be the standardization of the different available datasets, so that they become easily interchangeable. A selection of a training and test set for all datasets is crucial, due to the huge overlaps of the existing data, it is not possible to overcome the lack of a proper test set by use a cross validation.

The processing of the dataset could be made faster and more accurate, if the acquisition device would be calibrated in terms of generating a valid illumination model or an improvement of the device itself, providing a diffuse illumination. It seems that some wood shows some speckle effects, which make some holes appear white instead of black. But this has to be verified by an expert.

Apart from this adjustments of the dataset, it would be a future task, to acquire a new dataset on bigger wood samples and higher varieties. It is also questionable, if the wood inspected in an industrial context shows the same crafting features as the wood used for the dataset creation, which seems to have a polished wood.

B. Pre-Processing

To obtain better segmentation results a fuzzy-k-means that uses fixed initialization points could be implemented. Good segmentation results could be also expected if LBP would be used as segmentation basis which could be seen as a kind of neighbourhood based segmentation, just being much faster.

C. Feature Extractors

For both, the angle histograms and the OBSP histograms, more statistical values could be extracted to enhance the description abilities of both methods. Also it could be tested how much the diagonal projection of OBSP contributes to a classification, it might be considered to be left away to speed up the process by about the half of the time.

The exploration of morphological spectra should take a big part of future research, since they seem to be very promising. A morphological spectra approach could be also tried on the different histograms, by simply shifting them and extracting the differences between the shifts.

IX. CONCLUSION

Even if the conclusions that can be made on the used dataset are limited, we are here summarizing our results.

The work on this database showed that wood seems not to have very strong shape features which are targeted by curvelet, ranklet and wavelet approaches. On the other hand the shape based morphological approach showed potential. However, overall the simple and well known texture descriptor LBP showed a

very high performance. Considering its computational speed, practical use is very possible.

Some of the features we implemented based on the angle histogram and the summed projection of segmented areas horizontally, vertically and diagonal can be also computed very fast and increase the final classification performance.

As for the curvelet transform, coefficients of the General Gaussian Distribution individually performed better than the statistical and co-occurrence features. There is a potential in exploring which distance measure could improve their performance.

In future work we proposed the enhancement of the dataset and feature descriptors as well as advice the directed research on morphological spectra.

REFERENCES

- [1] Prasetyo, "A comparative study of feature extraction methods for wood texture classification," Master's thesis, LE2I Université de Bourgogne, 2010.
- [2] Prasetyo, M. Khalid, R. Yusof, and F. Meriaudeau, "A comparative study of feature extraction methods for wood texture classification," 2010, pp. 23–29.
- [3] M. Masotti and R. Campanini, "Texture classification using invariant ranklet features," *Pattern Recogn. Lett.*, vol. 29, pp. 1980–1986, October 2008.
- [4] F. Smeraldi, "Ranklets: Orientation selective non-parametric features applied to face detection," 2002.
- [5] —, "Fast algorithms for the computation of ranklets," in *ICIP*, 2009, pp. 3969–3972.
- [6] R. Xu, X. Zhao, X. Li, and C. I. Chang, "Target detection with improved image texture feature coding method and support vector machine."
- [7] E. Candès, L. Demanet, D. Donoho, and L. Ying, "Fast discrete curvelet transforms," 2005.
- [8] L. Dettori and L. Semler, "A comparison of wavelet, ridgelet, and curvelet-based texture classification algorithms in computed tomography," *Comput. Biol. Med.*, vol. 37, pp. 486–498, April 2007.
- [9] S. Arivazhagan, L. Ganesan, and T. G. S. Kumar, "Texture classification using curvelet statistical and co-occurrence features," in *Proceedings of the 18th International Conference on Pattern Recognition - Volume 02*, ser. ICPR '06, 2006, pp. 938–941.
- [10] F. Gómez and E. Romero, "Texture characterization using a curvelet based descriptor," in *Proceedings of the 14th Iberoamerican Conference on Pattern Recognition: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, ser. CIARP '09, 2009, pp. 113–120.
- [11] S. Arivazhagan, L. Ganesan, and T. G. S. Kumar, "Texture classification using ridgelet transform," *Pattern Recogn. Lett.*, vol. 27, pp. 1875–1883, December 2006. [Online]. Available: <http://portal.acm.org/citation.cfm?id=1228544.1228545>
- [12] M. Khalid, E. Lee, R. Yusof, and M. Nadaraj, "Design of an intelligent wood species recognition system," *Center for Artificial Intelligence and Robotics (CAIRO), Malaysia*, pp. 9–17, 2008.
- [13] P. Bautista and M. Lambino, "Co-occurrence matrices for wood texture classification," *Electronics and Communication Department, College of Engineering, MSU-Iligan Institute of Technology*, 2001.
- [14] R. Bremananth, B. Nithya, and R. Saipriya, "Wood Species Recognition Using GLCM and Correlation," in *Advances in Recent Technologies in Communication and Computing*, 2009. ART-Com'09. *International Conference on*. IEEE, 2009, pp. 615–619.
- [15] R. M. Haralick, Dinstein, and K. Shanmugam, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, pp. 610–621, November 1973.
- [16] O. T., Pietikäinen, M., and T. Maenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24(7), pp. 971–987, 2002.
- [17] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, pp. 51–59, 1996.
- [18] M. Akhloufi, X. Maldague, and W. Larbi, "A new color-texture approach for industrial products inspection," *Journal of Multimedia*, vol. 3, p. 44–50, 2008.
- [19] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. Citeseer, 2001.
- [20] J. Serra, "Courses on Mathematical Morphology – Chapter XXII The Covariance," <http://cmm.enscm.fr/~serra/cours/index.htm>, 2000. [Online]. Available: <http://cmm.enscm.fr/~serra/cours/index.htm>
- [21] P. Gómez-Gil, M. Ramírez-Cortés, J. González-Bernal, A. Pedrero, C. Prieto-Castro, D. Valencia, R. Lobato, and J. Alonso, "A Feature Extraction Method Based on Morphological Operators for Automatic Classification of Leukocytes," in *Artificial Intelligence, 2008. MICAI'08. Seventh Mexican International Conference on*. IEEE, 2008, pp. 227–232.
- [22] P. Maragos, "Pattern spectrum and multiscale shape representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 7, pp. 701–716, 2002.
- [23] J. Lu, K. Plataniotis, and A. Venetsanopoulos, "Face recognition using LDA-based algorithms," *Neural Networks, IEEE Transactions on*, vol. 14, no. 1, pp. 195–200, 2003.