Development of a data fusion model for electronic component recognition and automatic generation of life-cycle inventory PCB models

BY

Bernhard Föllmer

Department of Machine Tools and Factory Management

Chair of Industrial Information Technology

Submitted in partial fulfillment of the requirements

for the degree of

Master of Science in

Physical Engineering Science

at the

Technical University of Berlin

13.10.2014

Contents

List of figures

List of tables

Abbreviations

DFT
Discrete fourier transform, 10
FFT
Fast fourier transform, 10
LoG
Laplacian of Gaussion, 38

1. Introduction

ewt

1.1 Object recognition from 2D Images

Dg

1.1 Recycling potential of electronic waste

Asd

2. Recognition of electronic components

Erdt

2.1 Image preprocessing

S

Asf

2.1.1 Image rotation correction

To bypass the restriction of rotation invariant features for object recognition, the rotation angle of the printed circuit board images were determined. Since there is no fixed printed circuit

board orientation, the orientation is set by invariants of 90 degree whereas most of the electronic parts are horizontal or vertical alligned. The whole process is based on the assumption that Conducter tracks and electronic parts are mostly horizontal or vertical aligned and there structure and borders producing more horizontal and vertical edges than edges with different orientations. The rotation angle estimation is based on the rotation property of a discrete Fourier transform. The DFT of an image rotated by and angle Θ is the DFT of the unrotated image, rotated by the same angle Θ. The rotation property of a DFT is derived in (Maria Petrou, Costas Petrou, 2010) and therefore omitted here. The image rotation correction process is shown in Figure 1.

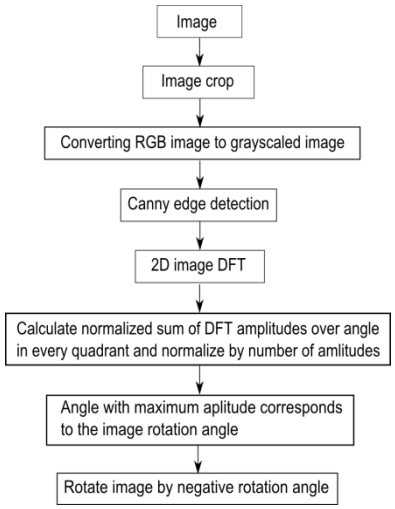


Figure 1: Image rotation correction process

At first the Image is cropped to a squared image [2000 x 2000] to reduce the process runtime. The RGB image is converted to grayscaled image and canny edge detection is applied. Afterward a 2D DFT is computed from the edge image. To estimate the rotation angle, the amplitude of the shifted 2D FFT image is summed up over discretized angles and normalized by number of amplitudes per angle step. The discretization is done in steps of 0.25 degree from 0 to 360 degree which results in a discretization error of 0.125 degree. The maximum of the normalized sum of amplitudes over the angle corresponds to the image rotation angle. With this process the rotation angle can be estimated with invariants of 90 degree image rotation.

An example of a rotated image by 3 degree, the edge image and amplitude of discrete Fourier transform is shown in Figure 2, Figure 3 and Figure 4. The accuracy of the angle estimation was not investigated in detail but inaccuracy could not be determined by eye.

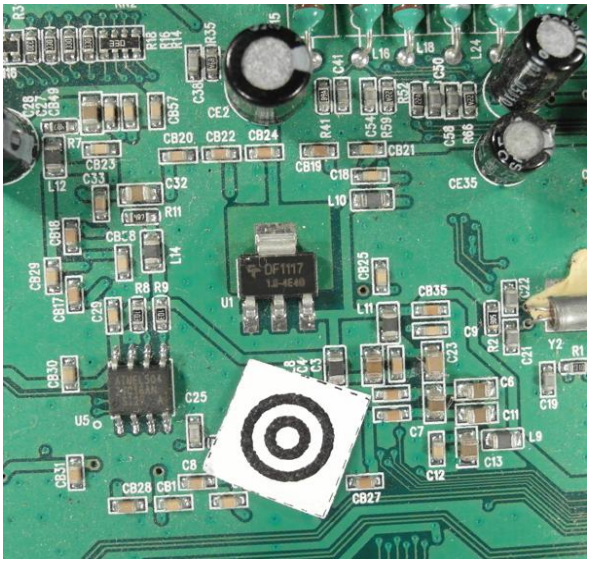


Figure 2: Image rotated by 3.0 degree

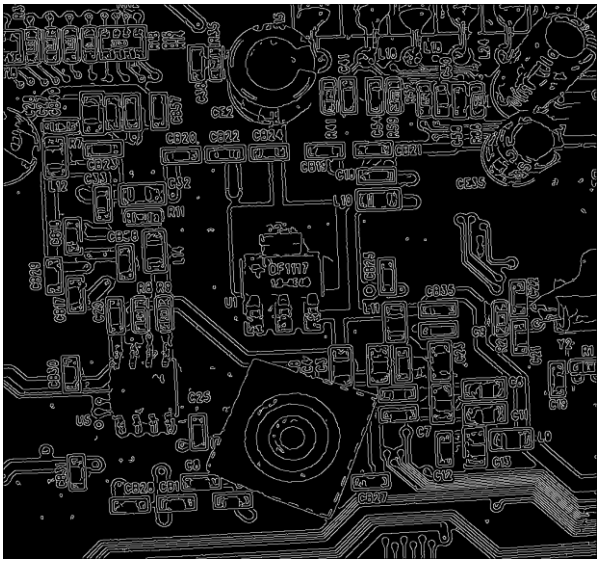


Figure 3: Canny edge image of the rotated image

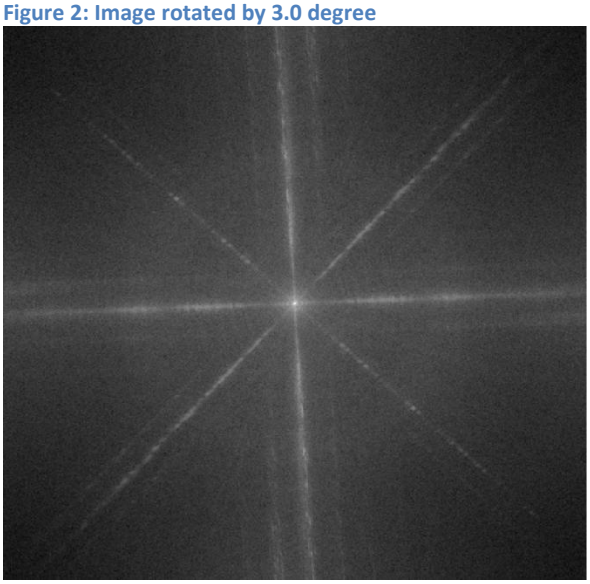


Figure 4: Shifted DFT of the rotated image (logarithmic representation)

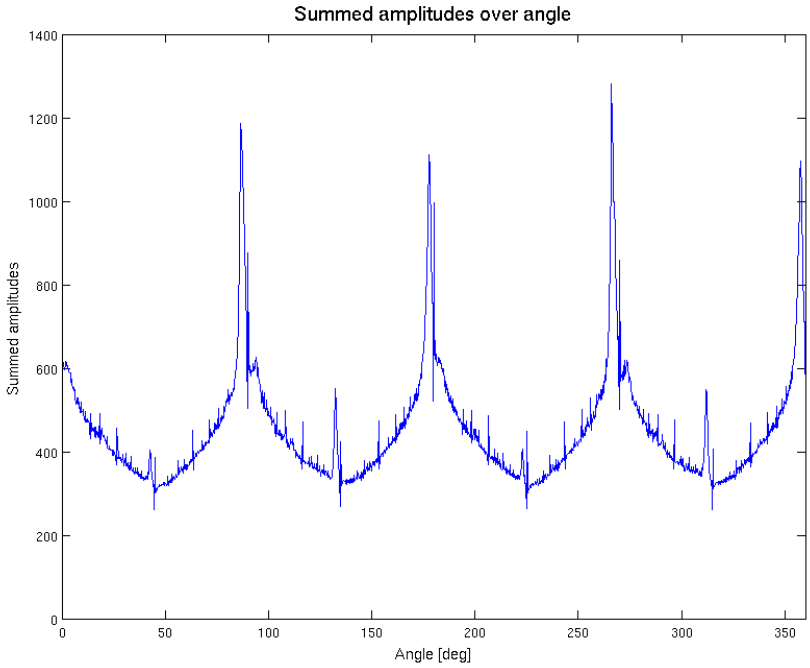


Figure 5: Summed amplitude over angle (invariants by 90 degree)

- Linien werdan auf punkte abgebildet

2.1.2 Scaling determination based on scaling symbol

To bypass the restriction of scale invariant features for object recognition, the scaling of the printed circuit board images were determined using a scaling symbol.

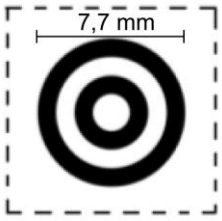


Figure 6: Scale symbol

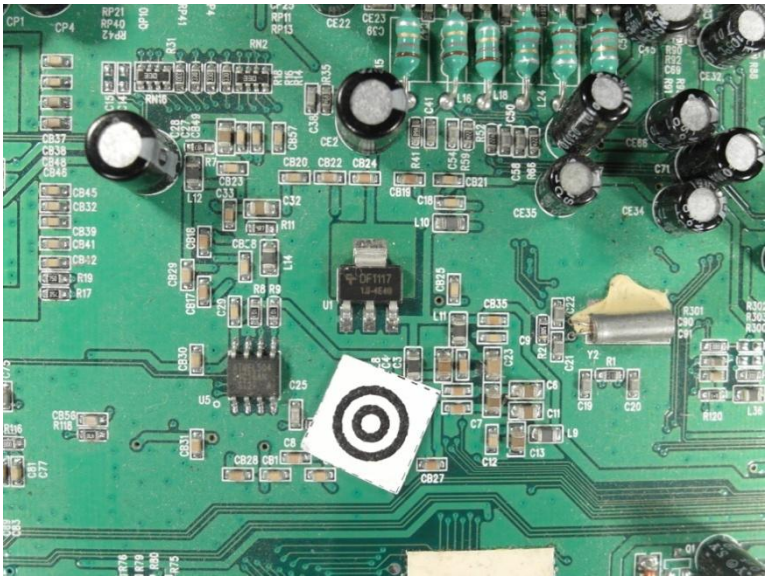


Figure 7: Scale symbol placed on the board

The scaling symbol is shown in Figure 6. The whole scaling determination process is shown in Figure 8.

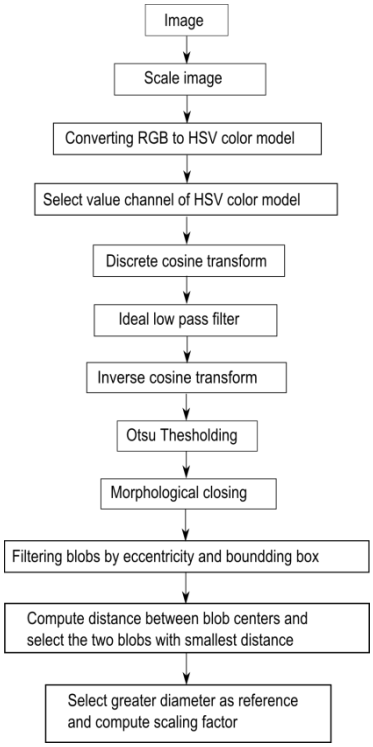


Figure 8: Scaling determination process

At first the image is converted from the RGB color model to the HSV color model and the brightness channel (value channel) is used to make a discrete cosine transform. The discrete cosine transform is frequently used in image compression such as the JPEG format. The discrete cosine transform is similar to the discrete Fourier transform but uses only cosine functions as kernels. The discrete cosine transform is shown in Equation (1) and (2) (Rafael C.Gonzalez, 2008).

$$T(u,v) = \sum_{x=0}^{n-1}\sum_{y=0}^{n-1} g(x,y)\alpha(u)\alpha(v)\cos\left[\frac{(2x+1)u\pi}{2n}\right]\cos\left[\frac{(2y+1)v\pi}{2n}\right] \quad (1)$$

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{n}} & for\ u = 0 \\ \sqrt{\frac{2}{n}} & for\ u = 1, 2, \dots, n-1 \end{cases} \quad (2)$$

$$\alpha(v) = \begin{cases} \sqrt{\frac{1}{n}} & for\ \ v = 0 \\ \sqrt{\frac{2}{n}} & for\ \ v = 1, 2, \dots, n-1 \end{cases} \tag{3}$$

To suppress illumination changes, an ideal low pass filter is applied in the frequency domain in which the first 10 x 10 cosine coefficients were discarded. Afterwards the inverse cosine transform is applied to get the image in time-domain. To extract the two dark circles of the scaling symbol, Otsu's method is used to automatically perform thresholding. To avoid salt and pepper noise, a morphological closing operator (5x5) is applied. The image is inverted and the eccentricity and bounding boxes are determined of the blobs. All blobs inside the eccentricity interval and inside the diameter interval are maintained, all others are discarded.

$$Maintained\ blobs = \{blobs,\ eccentricity_{min} < eccentricity\ \wedge\ diameter_{min} < diameter < diameter_{max}\} \tag{4}$$

- eccentrcity min angeben

To find the center of the scaling symbol, the distances between the centers of all blobs are calculated and the two blobs with the smallest distance are the inner and outer dark rings of the scaling symbol. The outer diameter of the larger blob is used as reference to calculate the image scale.

$$imagescale = \frac{diameter\ [pixel]}{diameter\ [mm]} \tag{5}$$

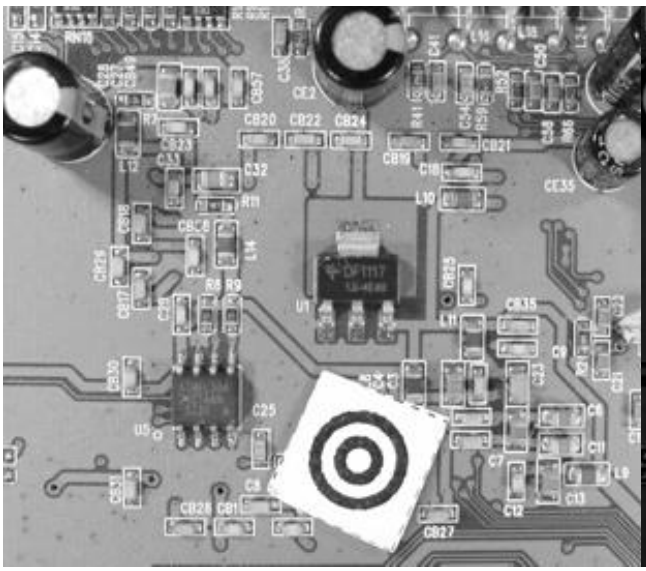


Figure 9: Value channel (brightness) of HSV color image

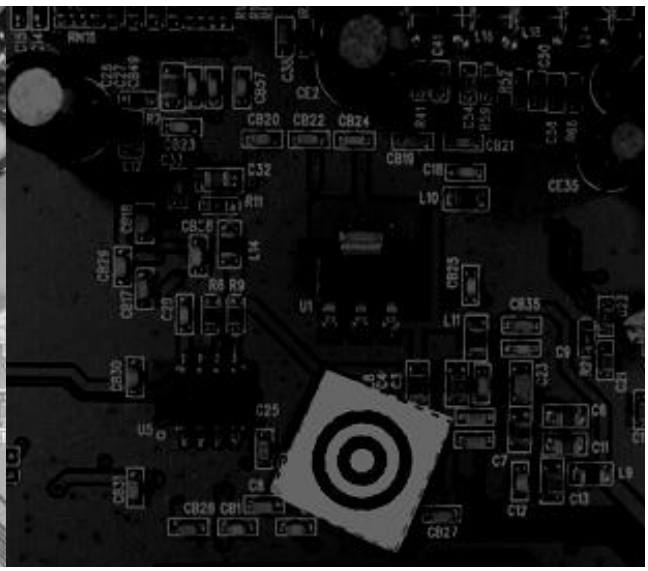


Figure 10: Cosine transform filtered image

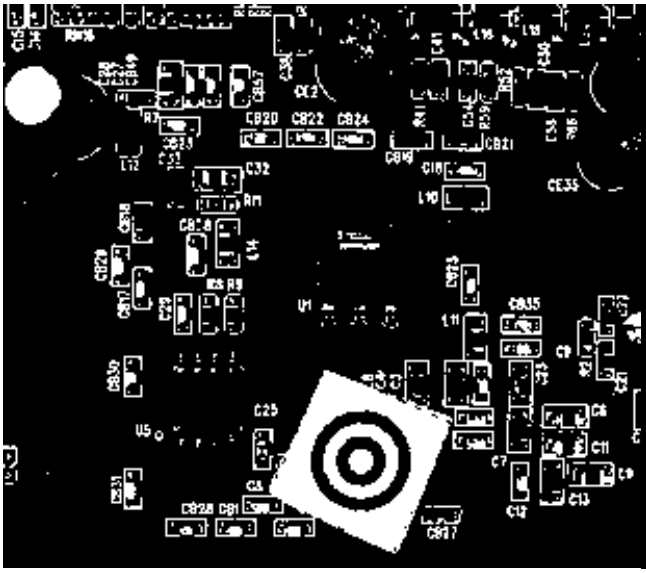


Figure 11: Otsu thresholding

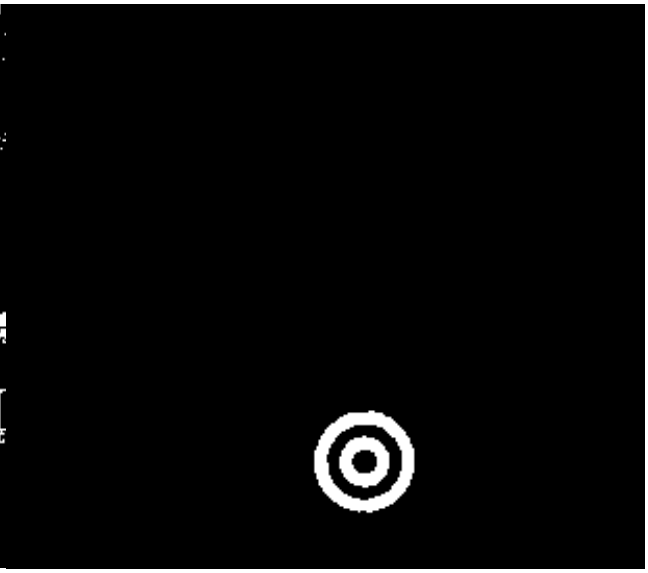


Figure 12: Blobs of the scaling symbol

2.2 Electronic component detection

Asdsad

2.2.1 Electronic component detection based on color based background detection

Sdf

2.2.2 Electronic component detection based on 3D range image

Sd

2.2.3 Electronic component detection based on normalized correlation

Sasd

3. Data fusion model

Asdas

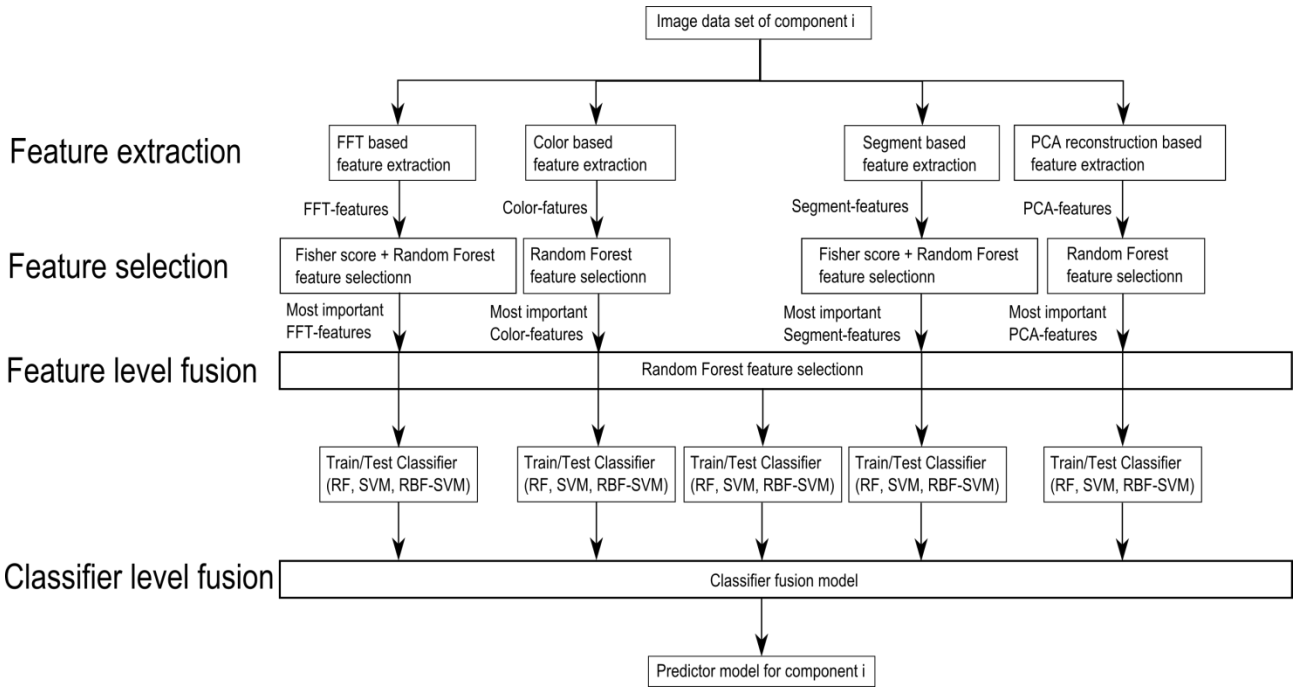


Figure 13: Data fusion model

Saf

dsfas

3.1 One-vs.-rest data fusion model

dsf

3.1.1 Classifier level fusion

The data fusion on classifier level (classifier level fusion) is performed to make the performance more robust against the difficulties that each individual classifier may have. Combining classifiers is one of the most widely explored methods in pattern recognition and it has been shown that these techniques can reduce error rate in classification tasks (Moreno-Seco). I this

approach each classifier is responsible for a specific feature subset. The first classifier rates the sample data based on the most important FFT-features, the second on the most important color features, the third on the most important segment features and the fourth on the most important PCA features. The fifth classifier rates the sample data based on the most important features of all important features of all feature extraction algorithms. The largest groups of classifier fusion methods operate on classifiers which produce so-called soft outputs. The outputs are real values in range [0, 1] (D. Ruta, B. Gabrys, 2000). The random forest classifier outputs the number of votes for a class based on the number of trees. The number of votes can be normalized by the number of trees to get a soft output.

In this approach the simple weighted vote scheme (SWV) is used to combine the five classifiers (Moreno-Seco). The soft outputs of all five classifiers are weighted by their estimation accuracy of the test samples. The output of the classifier fusion process is the soft-output P_i which represents the probability that the sample is from class i. $P_{i,k}$ represents the probability of classifier k to be component i. $P_{i,k,test}$ represents the probability of classifier k to be component i based on the true positive rate of the test set.

$$P_i = \sum_{k=1}^{5} w_{i,k} * P_{i,k} \tag{6}$$

$$w_{i,k} = \frac{\sum_{j=1}^{5} P_{i,j,test}}{P_{i,k,test}} \tag{7}$$

3.1.2 Feature level fusion

The feature level fusion is based on the feature selection approach whereas all the most important features of the feature selection algorithms are used as input features for a classifier in the classifier fusion step. This approach is based on the idea that a combination of features from different feature ranges can improve the estimation accuracy of a classifier.

h

4. Classification

Dsfdsf

4.1 Feature extraction algorithms for electronic components

Sad

4.1.1 Image resolution for feature extraction

The resulting features quality of feature extraction algorithms depend on the resolutions of the images. In general higher image resolutions improve the feature precision but also increase the run time and require more memory. Therefore a trade off between a high image resolution on one hand and memory usage and runtime on the other side must be found. In this approach the image resolution depends on the size of the component. Smaller components require a higher resolution than larger ones because there images contain more details.

- entropy

In this approach the resolution depends on the components area and the feature extraction algorithm.

$$area_{component}\ [mm^2] = width_{component}[mm] * hight_{component}[mm] \qquad (8)$$

$$PPMM(area_{component}) = a * \exp(-b\left(area_{component}\ [mm^2]\right) - c)\ [ppmm] \quad (9)$$

The algorithm dependent resolution parameters are defined in Table 1.

Table 1: Feature extraction algorithm based resolution parameter

	a	b	c
Fourier coefficients based feature extraction	5	0.003	15
Histogram based feature extraction	10	0.003	10
Segment based feature extraction	19	0.005	1
PCA reconstruction based feature extraction	18	0.005	2

The area and algorithm dependent resolution is plotted in Figure 14.

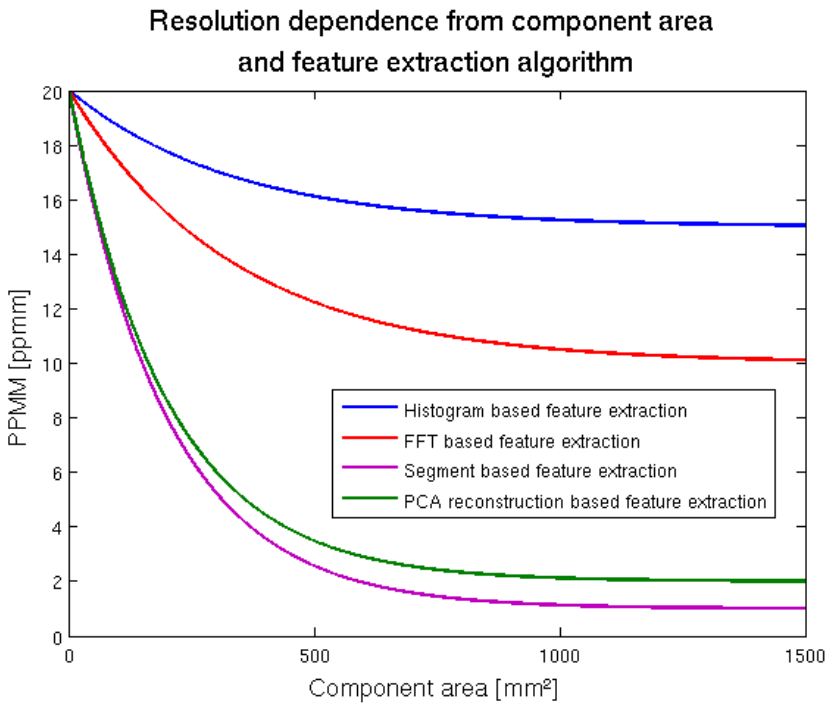


Figure 14: Image resolution

- Abbildung korrigieren

4.1.2 Fourier coefficients based feature extraction

Every periodical infinite signal can be decomposed in ….

Fourier descriptors as features were used in already used in applications for face recognition and object recognition (Campos, 2000).

The idea to use Fourier coefficients as features comes from the representation of solder joints by most electronic component images. Many computer vision systems for solder join detection, localization and segmentation have been develop. Specular reflections of solder joint depending on small changes in viewing direction and different shape and size of the solder

joints make it difficult to create a stable recognition system (Tianshoul, 2012). Many electronic components consist of several equidistant arranged solder joints. An example is the widely used DIP14 package seen in Figure 15. Since the solder joints appear in the grayscaled image as bright equidistant spots they should be representative frequencies in the 2D Fourier spectrum with the period around the solder joint distance.

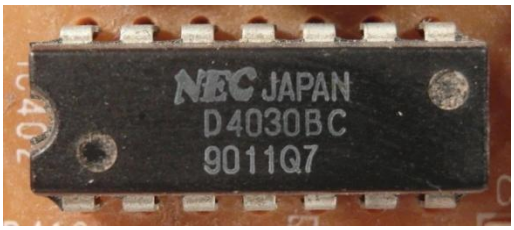


Figure 15: DIP14 package with equidistant solder joints

The 2D discrete Fourier transform for an M x N image is defined as

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\, e^{-j2\pi(\frac{ux}{M}+\frac{vy}{N})} \tag{10}$$

$u = 0, 1, 2, \dots, M-1$ and $v = 0, 1, 2, \dots, N-1$ where $f(x,y)$ is the image of size M x N (Rafael C.Gonzalez, 2008). The Fourier coefficients are in general complex numbers consisting of real and imaginary part. The real part represents the cosine and the imaginary the sinus proportion of the signal. The M x N image consists of M x N Fourier coefficients which produces 2 x M x N frequency features which is a large number of features that can be used. To increase execution time of the classifier and decrease recognition rate, a subset of low frequency features is extracted. Further research shows that spatial frequencies with lower frequency represent global information about the shape such as general orientation and proportion. The visual information is represente

Since the solder joints are the main focus for frequency feature, the solder joint distance of electronic components is used as a measure of minimal frequency period. In our feature extraction all Fourier coefficient (real and imaginary part) with a frequency under the cutoff frequency are used as features.

$$f_{cutoff} = \frac{1}{T_{cutoff}} = \frac{1}{0.50\ mm} = 2\ mm^{-1} \tag{11}$$

The numbers of features depend on the size of the component image.

$$\#frequency\ features = \left\lceil \frac{length\ [mm]}{T_{cutoff}[mm]} + 1 \right\rceil * \left\lceil \frac{width\ [mm]}{T_{cutoff}[mm]} + 1 \right\rceil \tag{12}$$

Abtastthorem (+1)

Another interesting feature extraction based on wavelets could analyze frequencies and there temporal occurrence which could improve the classification results. A view on that topic was done in the prospective section 9.2.

- Energie in niedrigen frequnzen -> hohe information (paper)

4.1.3 Histogram based feature extraction

Color image segmentation algorithms for automated optical inspection in electronics have already been investigated (Tarnawski, 2003). Electronic components varying in color, such as several tantalum capacitors, ICs or SMD electrolyte capacitors. To find representative features the color model has to be defined. In this system, the HSV (hue-saturation-value) color model was used because the channels are not that strong correlated such as in the RGB color model and relative stable against illumination changes or shadows (H. Cheng, H. Jiang, Y. Sun, Jingli Wang, 2000), (Noor. A. Ibraheem, Mokhtar M. Hasan, Refiqul Z. Khan, Pramod K. Mishra, 2012). Histogram based features are features which depend on the probability distribution of the pixels over the color values. In the histogram based feature extraction 10 equidistant bins are defined in each color channel (hue-saturation-value) and the pixel distributions are determined and normalized by the number of pixels. The values correspond to the probability density function of the gray value. All ten bin values are use as features that results in 30 color features. The histogram of a tantalum capacitor is seen in Figure 16, Figure 17, Figure 18 and Figure 19.

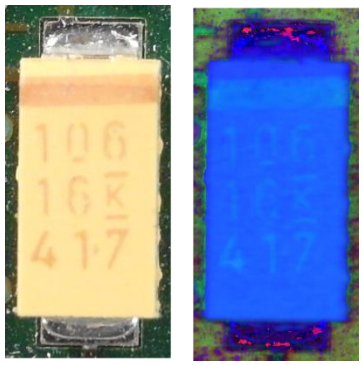


Figure 16: Tantalum capacitor in RGB color model (left) and HSV color model (right)

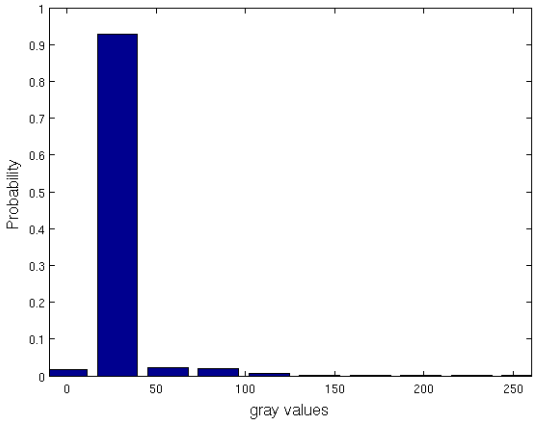


Figure 17: Normalized histogram of hue channel (tantalum capacitor)

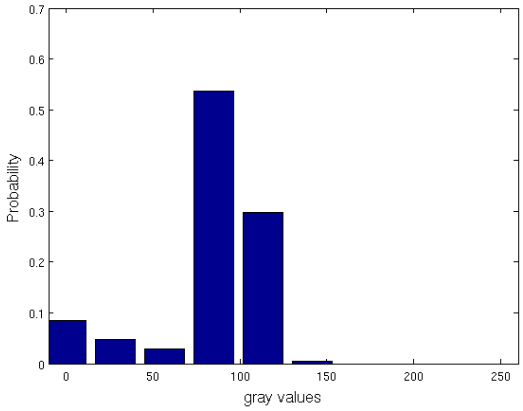


Figure 18: Normalized histogram of saturation channel (tantalum capacitor)

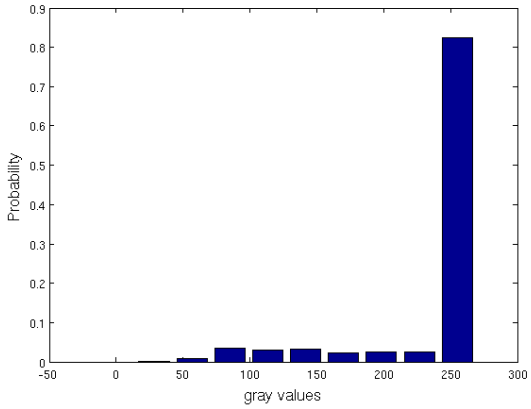


Figure 19: Normalized histogram of value channel (tantalum capacitor)

4.1.4 Segment based feature extraction

The segment based feature extraction is based on the idea that electronic components can be identifies by striking color regions. One approach to extract information about spatial proximity

of pixels is the region growing algorithm. The region growing starts with seed points which pixel position is the most important drawback.

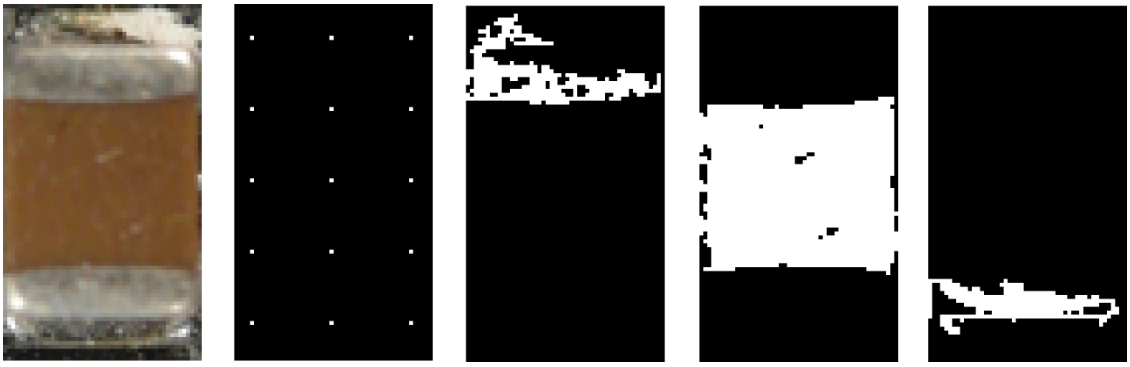
In this feature extraction algorithm, the seed points are uniformly distributed over the part image. The region growing and feature extraction of the segments is done in HSV color space. The distance between the seed points depends on the size of the component which is based on the assumption that smaller components consist of smaller color regions than big components. The equation for the distances is specified in (13).

$$A_{component} = length_{component} * \ width_{component} [mm^2] \quad (13)$$

$$\Delta_{seed} = 0.02 * A_{component} + 0.5 \ [mm] \quad (14)$$

In the region growing segmentation approach the neighboring pixel of the seed pixel are added to the segment if the distance between the color of the seed point and the neighboring pixel is smaller than a certain value. Further the neighboring pixels of the new segment are added to the segment if their distance to the color mean of the segment is smaller than a certain value. This process is iterated until no more pixels are added to the new segment (Maria Petrou, Costas Petrou, 2010).

One example is the Multi-layer ceramic capacitor (MLCC) shown in .



- Abbildung korrigieren

Seven Features are extracted for every segmented region which are the x-coordinate of center of gravity, y-coordinate of center of gravity, bounding box height, bounding box width and the arithmetic mean color value in all three color channels.

- Border to seeds in picture
- Formeln region growing

4.1.5 PCA reconstruction error based feature extraction

Object detection based on image reconstruction with Principal Component Analyses was already applied for pedestrian recognition. (L. Malagón-Borja, Olac Fuentes, 2007). A similar approach was used to extract a PCA reconstruction feature. In that system the PCA reconstruction is based on edge images of the parts. At first a subset of the training images of parts are used to find principal components which can compress optimally only the kind of images that were used to compute the principal components.

4.1.5.1 Image reconstruction with PCA

A set of m part images I_i each of size r x c is reshaped to a vectors $\boldsymbol{v_i}$ of size r*c x 1. First the mean vector $\boldsymbol{\mu}$ and the covariance matrix **C** are computed for all vectors according to (15) and (16).

$$\boldsymbol{\mu} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{v_i} \tag{15}$$

$$\mathbf{C} = \sum_{i=1}^{m} (\mathbf{v_i} - \boldsymbol{\mu})(\mathbf{v_i} - \boldsymbol{\mu})^T \tag{16}$$

Next the eigenvectors and eigenvalues are computed and sorted according to decreasing eigenvalues. This computation can be done in several ways in which Matlab implementation based on the QZ algorithm was used in this approach. The eigenvectors $\boldsymbol{e_i}$ with the k largest

eigenvalues λ_i of the covariance matrix are used to construct the projection matrix **P** of size r*c x k. The projection of an image vector v_i into the eigenspace is given by

$$\boldsymbol{p} = \boldsymbol{P}(\boldsymbol{v_i} - \boldsymbol{\mu}) \tag{17}$$

The reconstruction of an image projects the image into the PCs and from this projection, try to recover the original image by applying the invers projection matrix. The projection and recover step is shown in whereas $\boldsymbol{v_i}'$ is the reconstructed image of the image $\boldsymbol{v_i}$.

$$\boldsymbol{v_i'} = \boldsymbol{P^T p} + \boldsymbol{\mu} = \boldsymbol{P^T P}(\boldsymbol{u} - \boldsymbol{\mu}) + \boldsymbol{\mu} \tag{18}$$

The reconstruction error is defined by the euclidean distance between the image $\boldsymbol{v_i}$ and its reconstructed image $\boldsymbol{v_i}'$.

$$d = |\boldsymbol{v_i} - \boldsymbol{v_i}'| = \sqrt{\sum (\boldsymbol{v_i} - \boldsymbol{v_i}')^2} \tag{19}$$

Often there will be just a view large eigenvalues whose eigenvectors contain the most information while the rest of the dimensions generally contain noise (Duda, 2001).

4.1.5.2 PCA feature construction

A set of PCs from a set of images from one component reconstruct the images of the same component better than other types of images. The fact can be observed in the images in Figure 20 and can be used to create a feature which represents the difference between the reconstruction error of the projection into the component PCs and the non-component PCs.

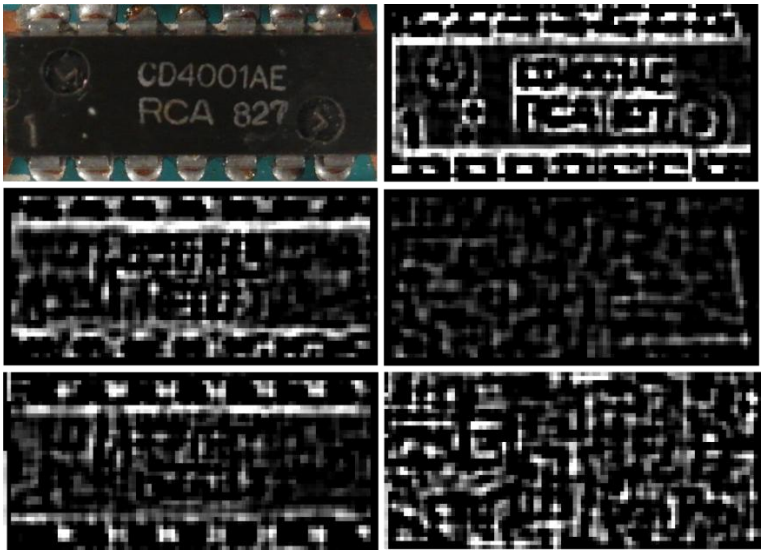


Figure 20: DIP14 (top, left), DIP14 edge image (top, right), DIP14 reconstruction with component PCs (middle, left), DIP14 reconstruction with non-component PCs (middle, right), unit matrix projection into component PCs (bottom, left), unit matrix projection into non-component PCs (bottom, right)

In this approach the component images and non-component images are scaled according to the size of the component. After the RGB images are converted to grayscaled images and the image intensity values are adjusted for contrast improvement. To obtain a feature that contains information about the edges the edge image was created by applying a Laplacian of Gaussian (LoG) filter. The projection matrix $\boldsymbol{P}_{ep}$ and the mean $\boldsymbol{\mu}_{ep}$ are computed from a subset of component images and the projection matrix $\boldsymbol{P}_{en}$ and mean $\boldsymbol{\mu}_{en}$ for the non-component images are computed. The reconstruction based on the component PC projection is computed by (20) and the reconstruction based on the non-component PC projection is computed by (21).

$$\boldsymbol{r}_{ep} = \boldsymbol{P}_{ep}^{T}\boldsymbol{P}_{ep}\left(\boldsymbol{e} - \boldsymbol{\mu}_{ep}\right) + \boldsymbol{\mu}_{ep} \tag{20}$$

$$\boldsymbol{r}_{en} = \boldsymbol{P}_{en}^{T}\boldsymbol{P}_{en}(\boldsymbol{e} - \boldsymbol{\mu}_{en}) + \boldsymbol{\mu}_{en} \tag{21}$$

The reconstruction error of component images projected by component PCs should be smaller for component images than non-component images. The features is the difference between the

reconstruction error projected in the component PCs and the error projected in the non-component PCs shown in (22).

$$\boldsymbol{f}_{pca} = \sum |\boldsymbol{r}_{ep} - \boldsymbol{\mu}_{ep}| - \sum |\boldsymbol{r}_{en} - \boldsymbol{\mu}_{en}| \tag{22}$$

The process is shown in Figure 21.

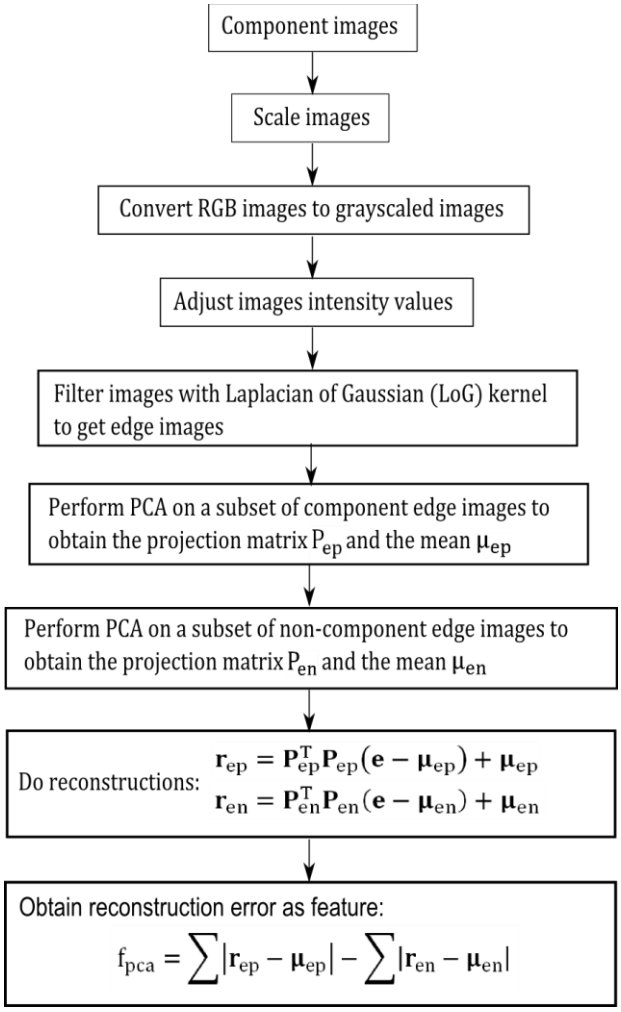


Figure 21: PCA feature construction process

4.2 Feature selection and feature fusion techniques for classification

Asd

4.2.1 Introduction to feature selection

Asd

- Wrapper, embedded
- Guyon
-

4.2.2 Fisher score

Fisher score is a variable ranking method that rates the efficient for discriminations for each feature. It can be applied in two-class problems as well as in multi-class problems. The score evaluates each feature by the ration of the between class variance to the within-class variance (Guyen, 2003). Suppose we have a set of d-dimensional samples $x_1, \dots, x_n$, n_k is the number of samples in the subset D_k labeled ω_k and c is the number of classes. The Fisher score of the j-th feature is computed in (23).

$$F(x^j) = \frac{\sum_{k=1}^{c} n_k (\mu_k^j - \mu^j)^2}{(\sigma^j)^2}) \tag{23}$$

Where σ^j is the standard deviation and μ^j the mean of the whole data set corresponding to the j-th feature and x_i^j is the j-th feature of the sample $\boldsymbol{x}_i$.

$$(\sigma^j)^2 = \sum_{k=1}^{c} n_k (\sigma_k^j)^2 \tag{24}$$

$$\sigma_k^j = \sum_{\boldsymbol{x}_i \in D_k} x_i^j - \tilde{\mu}_k^j \tag{25}$$

$$\tilde{\mu}_k^j = \frac{1}{n_i} \sum_{\boldsymbol{x}_i \in D_k} x_i^j \tag{26}$$

$$\mu^j = \frac{1}{n}\sum_{k=1}^{c} n_k \tilde{\mu}_k^j \tag{27}$$

After computing the fisher score for each feature, it selects the top-m features as the subset of features. The number of features m can be fixed or depend on a score threshold. The score of each feature is computed independently of all other features. Therefore the feature subset can be suboptimal because features with low individual scores but a very high score when they are combined are discarded furthermore redundant features are not discarded (Q. Gu, Z. Li, J. Han, 2012). In this approach the fisher score is only used in the two stage feature selection and not applied alone for feature selection (see chapter 4.2.5).

- is filter method

4.2.3 Random forest feature selection

The Random forest feature selection is based on the out-of-bag (oob) error estimation. Each tree is constructed using different bootstrap samples from the data. A subset of samples is left out and not used to construct the k-th tree (oob-samples). Each sample that was left out to construct the tree is predicted by the k-th tree and compared to the true class of the sample. This is done with all trees of the random forest and the error over all trees and out-of-bag-samples are summed and divided by the number of out-of-bag-samples (Breiman, www.stat.berkeley.edu).

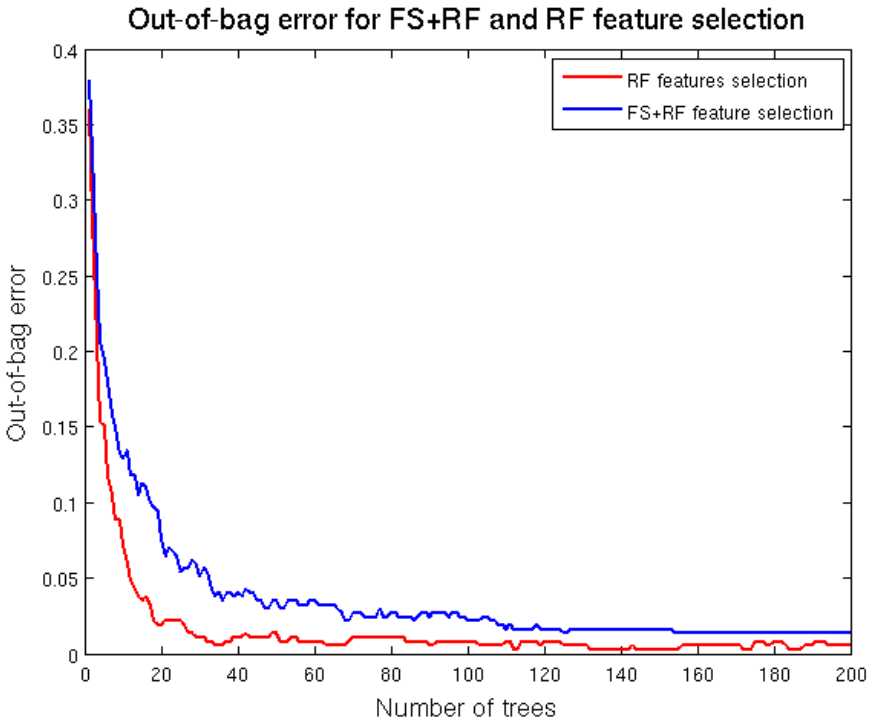
In the Random forest feature selection approach the oob-error is estimated. Now the values of the m-th feature of the oob-samples are randomly permuted and the the new oob-error is estimated. Subtract the number of oob-errors made by the variable-m-permuted oob-samples from the number of oob-errors made by the untouched oob-samples. The average of this number over all trees in the forest is the raw importance score for variable m. This raw importance score is divided by the standard deviation to get the z-score which is used as the variable importance score (Breiman, Random Forests, 2001).

- Redundante features
- Plot tantalum importance

4.2.4 Fisher score + Random forest feature selection

In practice random forest cannot handle a lot of features because it requires a lot of time to estimate the trees of the random forest and the accuracy decreases with a large number of features (Y. Chen, C.Lin, 2003). This approach does feature selection in two steps. First the Fisher score is used to select a subset of features from the feature set with a large number of features. The features are selected by a fisher score threshold of 0.01. All features with a fisher score larger than the threshold value and maximum number of 200 features are selected for the second step. In the second step the random forest based feature selection from 4.2.3 is applied to select the most important features from step one.

The out-of-bag error depending on the number of random forest trees and 2112 used FFT features extracted from the Resistor network 1206 component was computed. The red graph shows the out-of-bag error from the two step feature selection (FS+FR) and the blue one the out-of-bag error from the random forest feature selection (RF). The graphs show that the error rate of the FS+RF feature selection approach decreases faster and becomes smaller.



– change plot

4.2.5 Survey of the most important features

dfg

Asdf

4.3 Random forest classifier

4.3.1 Introduction to Ensemble classifiers

In supervised learning a supervisor (teacher) provides a category label for each pattern in a training set which also are referred to classes or labels. The classification of pattern is based on classification models (classifiers) which are learning the reclassified patterns of the training set. An algorithm which constructs the model is called inducer and an instance of an inducer for a specific training set is called a classifier. The idea behind an ensemble classifier is to weight several individual weak classifiers and combine them to form a strong classifier. It is well known that ensemble methods can improve the prediction performance (Rokach, 2010).

4.3.2 Random forest ensemble classifier

The random forest is an ensemble classifier where the individual classifiers are unpruned tree predictors. The generalization error for forests converges to a limit as the number of trees becomes large (Breiman, Random Forests, 2001).

4.4 Support vector machine classifier

Asdf

5. Decision fusion for component recognition

Asf

6. Optical character recognition of electronic component marking

Dsf

6.1 Introduction

Dsf

6.2 Character segmentation

Dsf

6.3 Optical character recognition with Tesseract and Cognex Vision Pro software

Asdf

6.4 Electronic part label verification based on Octopart database

Dsf

7. Experimental results

7.1 Implementation

Asfd

7.2 Dataset creation

The dataset consist of 12 electronic components which were analyzed. The components are listed in Appendix A. The component selection depends on the occurring frequency on the available printed circuit boards. It was taken care that also similar looking components were selected. Therefor the DIP14 component and DIP16 component which differ almost only by number and position of solder joints were selected. In addition the tantalum capacitors of different size but similar appearance were selected. For electronic component recognition, a machine learning application was used whereas multiple representation of the component must be created to analyze representative features. The component representations are taken from different parts of a component and different printed circuit boards to create a representative dataset. The available printed circuit boards are seen in .

To detect the edges of the parts, border pixels are also selected from the printed circuit board images. Additional important information and properties of the component are listed in.

Table 2: Component properties

Component properties	Description
Package properties	
Component length	
Component width	
Component border size	
Package DOF	
OCR properties	
ROI for optical character recognition	
Subset of characters for optical character recognition	
Maximum and minimum number of OCR lines	

Frequency features generation properties	
Image scale for frequency feature generation	
Number of Fourier coefficient features	
Border cut information	
Color histogram features	
Image scale for histogram feature generation	
Segment features	
Image scale for histogram feature generation	
Number of initial seed points for region growing approach	
PCA reconstruction features	
Image scale for histogram feature generation	
Kernel size for LoG (Laplacian of Gaussian) edge detection	
Number of PCs	
LCI properties	
ILCD-model full aggregated model	
ILCD-model composition model	

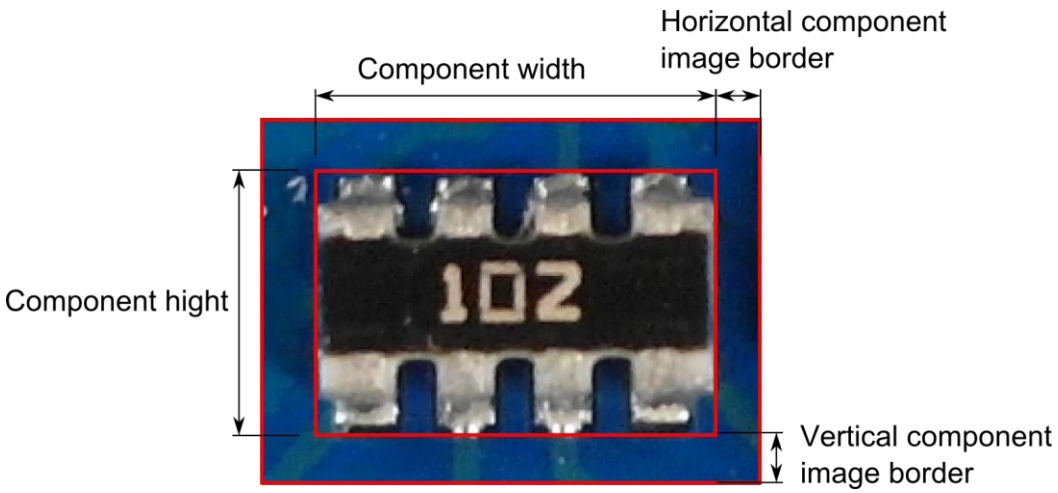


Figure 22: Component border definition

A section of the component database is shown in Figure 23.

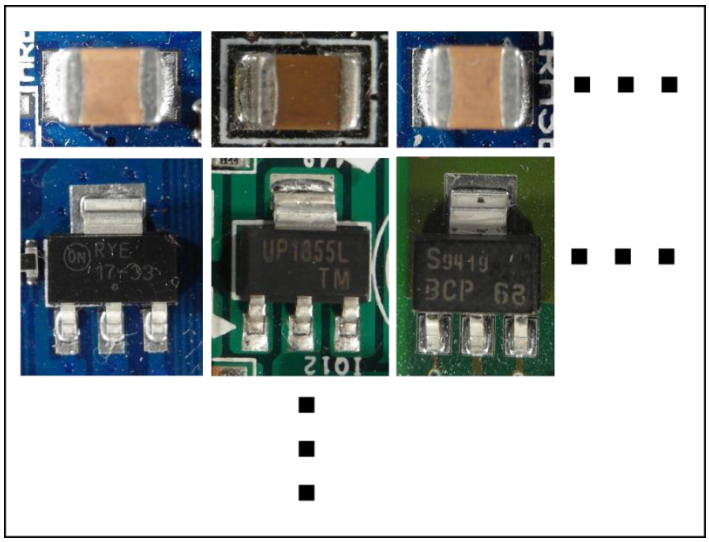


Figure 23: Database section

7.2.1 Image acquisition

The image acquisition was done with a Samsung EX2F camera and a working distance in a range from 20 mm to 120 mm through the Object. Autofocusing was used to get sharp images. The working distance was adapted to the size of the component in which the distance was decreased for smaller components and increased for bigger components. For illumination a bright-field incident illumination was selected because it generates a uniformly bright, well-contrasted image (Imaging, 2012). The lighting sources consist of four DSL-1110 table lamps with diffusion film to generate a uniformly bright and diffuse illumination. The image acquisition system is seen in Figure 24.

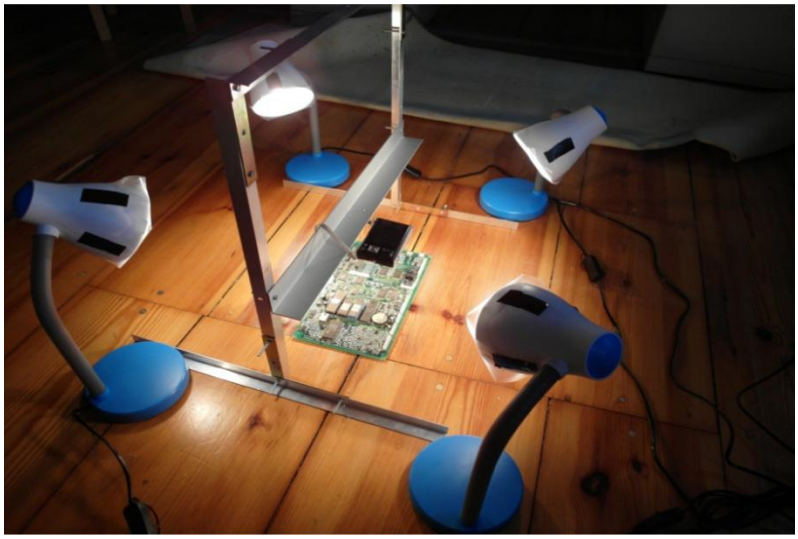


Figure 24: Image acquisition system

- Verzerrungen in bildern -> ausschnitt
- Schatten
- Winkel
- Kamera bewerten -> ausblick besseres kamerasystem
- Tiefenschärfe
-
-

7.3 Classification results

Ef

7.4 Optical character recognition results

Asd

8. Life-cycle inventory analyses of printed circuit boards

Asdf

8.1 Categorization of WEEE and recycling potential of PCB waste

8.1.1 Recycling potential of electronic parts from PCB waste

8.1.2 Reuse potential of electronic parts from PCB waste

Safd

8.2 Printed circuit board region classification based on electronic part recognition results

dftg

8.2.1 PCB support material (epoxy)

Dsf

8.2.2 Detected but not correctly classified electronic parts

Fdg

8.2.3 Detected and correctly classified electronic parts

Dfg

8.2.4 Detected, correctly classified and label recognized electronic parts

Jhg

8.3 GaBi-Software and LCI data availability of electronic components

Saf

8.4 Increasing of precious metal concentration by selective dismantling

gh

8.4.1 Increasing of tantalum concentration by selective dismantling

hg

8.5 International Reference Life cycle Data System (ILCD) format for LCI-automatic generation of LCI-models

Sdf

8.6 Arduino Due board LCI-model

9. Conclusion and prospects

9.1 Real time PCB board inspection

9.2 Feature extraction based on Wavelet basis functions

dsf

Appendix A

Table 3: Components in database

Component name and description	Component image
	
	
	

	 201
	 KN2 34
	 TOSHIBA 9014H 74HC4094AP JAPAN
	 NEC JAPAN D4030BC 9011Q7
	 103
	 953B 82F