PCB part recognition for material recycling

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 6](#_Toc404092897)

[List of tables 8](#_Toc404092898)

[Abbreviations 9](#_Toc404092899)

[1. Introduction 10](#_Toc404092900)

[1.1 Recycling potential of electronic waste 10](#_Toc404092901)

[1.1 Object recognition from 2D Images 10](#_Toc404092902)

[2. Recognition of electronic components 10](#_Toc404092903)

[2.1 Image preprocessing 10](#_Toc404092904)

[2.1.1 Image rotation correction 10](#_Toc404092905)

[2.1.2 Scaling determination based on scaling symbol 13](#_Toc404092906)

[2.2 Electronic component detection 18](#_Toc404092907)

[2.2.1 Electronic component detection based on color based background detection 18](#_Toc404092908)

[2.2.2 Electronic component detection based on 3D range image 18](#_Toc404092909)

[2.2.3 Electronic component detection based on normalized correlation 18](#_Toc404092910)

[3. Data fusion model 18](#_Toc404092911)

[3.1 Multiclass data fusion model 18](#_Toc404092912)

[3.2 One-vs.-rest data fusion model 19](#_Toc404092913)

[3.2.1 Classifier level fusion 19](#_Toc404092914)

[3.2.2 Feature level fusion 20](#_Toc404092915)

[4. Classification 21](#_Toc404092916)

[4.1 Feature extraction algorithms for electronic components 21](#_Toc404092917)

[4.1.1 Image resolution for feature extraction 21](#_Toc404092918)

[4.1.2 Fourier coefficients based feature extraction 23](#_Toc404092919)

[4.1.3 Histogram based feature extraction 25](#_Toc404092920)

[4.1.4 Segment based feature extraction 26](#_Toc404092921)

[4.1.5 PCA reconstruction error based feature extraction 28](#_Toc404092922)

[4.2 Feature selection and feature fusion techniques for classification 32](#_Toc404092923)

[4.2.1 Introduction to feature selection 32](#_Toc404092924)

[4.2.2 Fisher score 32](#_Toc404092925)

[4.2.3 Random forest feature selection 33](#_Toc404092926)

[4.2.4 Fisher score + Random forest feature selection 34](#_Toc404092927)

[4.2.5 Survey of the most important features 34](#_Toc404092928)

[4.3 Random forest classifier 35](#_Toc404092929)

[4.3.1 Introduction to Ensemble classifiers 35](#_Toc404092930)

[4.3.2 Introduction to Random forest ensemble classifier 35](#_Toc404092931)

[4.3.3 Random forest training 35](#_Toc404092932)

[4.3.4 Random forest prediction 36](#_Toc404092933)

[4.3.5 Out-of-bag (oob) estimation 36](#_Toc404092934)

[4.4 Support vector machine classifier 37](#_Toc404092935)

[4.4.1 Linear Support vector machine 37](#_Toc404092936)

[4.4.2 RBF Support vector machine 37](#_Toc404092937)

[5. Optical character recognition of electronic component marking 37](#_Toc404092938)

[5.1 Optical character recognition difficulties and limits 38](#_Toc404092939)

[5.2 Optical character recognition flow chart 39](#_Toc404092940)

[5.3 Optical character recognition with Tesseract and Cognex Vision Pro software 45](#_Toc404092941)

[5.3.1 Tesseract OCR engine 45](#_Toc404092942)

[5.3.2 Cognex VisionPro® OCRmax engine 46](#_Toc404092943)

[5.4 Electronic part label verification based on Octopart database 46](#_Toc404092944)

[6. Experimental results 47](#_Toc404092945)

[6.1 Implementation 47](#_Toc404092946)

[6.2 Dataset creation 47](#_Toc404092947)

[6.2.1 Image acquisition 50](#_Toc404092948)

[6.2.2 Dataset composition 51](#_Toc404092949)

[6.3 Feature selection results 52](#_Toc404092950)

[6.4 Classification results 54](#_Toc404092951)

[6.4.1 Ones-vs.-rest classification result 54](#_Toc404092952)

[6.4.2 Multiclass classification result 56](#_Toc404092953)

[6.5 Optical character recognition results 56](#_Toc404092954)

[6.5.1 Optical character recognitio0n on Word level 56](#_Toc404092955)

[6.5.2 Optical character recognition results on label level 56](#_Toc404092956)

[6.5.3 Octopart based part name assignment 56](#_Toc404092957)

[7. Life-cycle inventory analyses of printed circuit boards 56](#_Toc404092958)

[7.1 Categorization of WEEE and recycling potential of PCB waste 56](#_Toc404092959)

[7.1.1 Recycling potential of electronic parts from PCB waste 57](#_Toc404092960)

[7.1.2 Reuse potential of electronic parts from PCB waste 57](#_Toc404092961)

[7.2 Printed circuit board region classification based on electronic part recognition results 57](#_Toc404092962)

[7.2.1 PCB support material (epoxy) 57](#_Toc404092963)

[7.2.2 Detected and not correctly classified electronic parts 57](#_Toc404092964)

[7.2.3 Detected and correctly classified electronic parts 57](#_Toc404092965)

[7.2.4 Detected, correctly classified and label recognized electronic parts 57](#_Toc404092966)

[7.3 GaBi-Software and LCI data availability of electronic components 57](#_Toc404092967)

[7.4 Increasing of precious metal concentration by selective dismantling 58](#_Toc404092968)

[7.4.1 Increasing of tantalum concentration by selective dismantling 58](#_Toc404092969)

[7.5 International Reference Life cycle Data System (ILCD) format for LCI-automatic generation of LCI-models 58](#_Toc404092970)

[7.6 Arduino Due board LCI-model 58](#_Toc404092971)

[8. Conclusion and prospects 58](#_Toc404092972)

[8.1 Real time PCB board inspection 58](#_Toc404092973)

[8.2 Feature extraction based on Wavelet basis functions 58](#_Toc404092974)

[Appendix A 58](#_Toc404092975)

# List of figures

[Figure 1: Image rotation correction process 11](#_Toc404092976)

[Figure 2: Image rotated by 3.0 degree 12](#_Toc404092977)

[Figure 3: Canny edge image of the rotated image 12](#_Toc404092978)

[Figure 4: Shifted DFT of the rotated image (logarithmic representation) 12](#_Toc404092979)

[Figure 5: Summed amplitude over angle (invariants by 90 degree) 13](#_Toc404092980)

[Figure 6: Scale symbol 14](#_Toc404092981)

[Figure 7: Scale symbol placed on the board 14](#_Toc404092982)

[Figure 8: Scaling determination process 15](#_Toc404092983)

[Figure 9: Value channel (brightness) of HSV color image 17](#_Toc404092984)

[Figure 10: Cosine transform filtered image 17](#_Toc404092985)

[Figure 11: Otsu thresholding 17](#_Toc404092986)

[Figure 12: Blobs of the scaling symbol 17](#_Toc404092987)

[Figure 13: Data fusion model 19](#_Toc404092988)

[Figure 14: Image resolution 23](#_Toc404092989)

[Figure 15: DIP14 package with equidistant solder joints 24](#_Toc404092990)

[Figure 16: Tantalum capacitor in RGB color model (left) and HSV color model (right) 26](#_Toc404092991)

[Figure 17: Normalized histogram of hue channel (tantalum capacitor) 26](#_Toc404092992)

[Figure 18: Normalized histogram of saturation channel (tantalum capacitor) 26](#_Toc404092993)

[Figure 19: Normalized histogram of value channel (tantalum capacitor) 26](#_Toc404092994)

[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) 30](#_Toc404092995)

[Figure 21: PCA feature construction process 31](#_Toc404092996)

[Figure 22: Difficulties of IC marking recognition 39](#_Toc404092997)

[Figure 23: Label composition from words 42](#_Toc404092998)

[Figure 24: IC marking recognition flow chart 44](#_Toc404092999)

[Figure 25: Component border definition 49](#_Toc404093000)

[Figure 26: Database section 50](#_Toc404093001)

[Figure 27: Image acquisition system 51](#_Toc404093002)

[Figure 28: A comparison of different feature selection approaches 53](#_Toc404093003)

# List of tables

[Table 1: Feature extraction algorithm based resolution parameter 22](#_Toc404093004)

[Table 2: Component properties 48](#_Toc404093005)

[Table 3: Dataset composition 52](#_Toc404093006)

[Table 5: Dataset approaches for non-part images 54](#_Toc404093007)

[Table 6: Random forest classification results 55](#_Toc404093008)

[Table 7: Components in database 58](#_Toc404093009)

[Table 4: Most important selected features 61](#_Toc404093010)

[Table 8: Random forest Classification results (comprehensive) 62](#_Toc404093011)

# Abbreviations

DFT

Discrete fourier transform, 11

FFT

Fast fourier transform, 11

LoG

Laplacian of Gaussion, 48

# Introduction

ewt

## Recycling potential of electronic waste

Dg

* Derzeitiger Stand des Recyclings + Am Ende der Arbeit wieder aufgreifen und Verbesserungen beschreiben

## Object recognition from 2D Images

# Recognition of electronic components

Erdt

## Image preprocessing

S

Asf

### 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 : 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.

|  |  |
| --- | --- |
| C:\Users\WIN\Masterthesis\Masterthesis\Masterarbeit_daten\2.2.1\rot1.png  Figure : Image rotated by 3.0 degree | C:\Users\WIN\Masterthesis\Masterthesis\Masterarbeit_daten\2.2.1\rot3.png  Figure : Canny edge image of the rotated image |
| C:\Users\WIN\Masterthesis\Masterthesis\Masterarbeit_daten\2.2.1\rot2.png  Figure : Shifted DFT of the rotated image (logarithmic representation) |  |



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

* Linien werdan auf punkte abgebildet

### 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 : Scale symbol



Figure : 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 : 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).

|  |  |
| --- | --- |
|  | () |
|  | () |

|  |  |
| --- | --- |
|  | () |

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.

|  |  |
| --- | --- |
| } | () |

* 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.

|  |  |
| --- | --- |
|  | () |

|  |  |
| --- | --- |
| C:\Users\WIN\Masterthesis\Masterthesis\Masterarbeit_daten\2.2.2\scale5.png  Figure : Value channel (brightness) of HSV color image | Figure : Cosine transform filtered image |
| Figure : Otsu thresholding | C:\Users\WIN\Masterthesis\Masterthesis\Masterarbeit_daten\2.2.2\scale6.png  Figure : Blobs of the scaling symbol |

## Electronic component detection

Asdsad

### Electronic component detection based on color based background detection

### Electronic component detection based on normalized correlation

### Electronic component detection based on 3D range image

Sasd

# Data fusion model

Asdas

## Multiclass data fusion model

Saf

dsfas

## One-vs.-rest data fusion model

Dsf



Figure : Data fusion model

### 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.

### 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 which represents the probability that the sample is from class . represents the probability of classifier to be component . represents the probability of classifier to be component based on the true positive rate of the test set.

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  | () | |  |
| |  |  | | --- | --- | |  | () | |  |

### Decision level fusion

Decision-level fusion consists of merging information at higher level of abstraction. The fusion sstep combines multiple algorithms to yield a final fused decision. In this approach the outputs of the classifier fusion models at the classifier fusion level are soft outputs between 0 and 1. The output of the classifier fusion model for component i can be interpreted as the probability that the detected part is from component i. The outputs from all classifier fusion outputs are combined to make a final decision to which component the examined part belongs (Dong, 2009).

There exists variety of Decision fusion techniques based on Bayesian decision theory, Dempster- shafer fusion methods, artificial neural networks or Principal component analyses.

* Explane what are they and how are they used

In this approach the Bayesian decision theory is used to combine the output of the classifier fusion level to make a final decision. The Bayes decision theory is a fundamental statistical approach to the Problem of Pattern classification. It is based on quantifying the tradeoffs between various classification decisions and the cost that accompany such decision (Duda, 2001). In this approach the soft-outputs of the Classifier fusion model for each component i are measurements that are continuous random variables between 0 and 1. Under the assumption that n components exist in the recognition database with their associated n classification outputs on classifier fusion level a feature vector can be defined in a -dimensional Euclidian spacecalled the feature space. Let be the finite set of c states. In this approach every component corresponds to one state whereas one state for components that are not in the recognition database is defined that the number of states is. Let be the state-conditional probability density function for, with the probability density function for conditioned on being the true state of nature.is the prior probability, and describes that the nature is in state . The posterior probability can be computed from by Bayes formula:

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  | () | |  |

with the evidence

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  | () | |  |

Suppose that we observe a particular and that we contemplate taking action . It the true state of nature is, by definition we will incur the loss . Because is the probability that the true state of nature is, the expected loss associated with taking action is merely (Duda, 2001)

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  | () | |  |

The action in this approach is the decision to assign the component to the part if and assign the unknown component if.

For recycling of precious metals the loss function can be based on the concentration of a specific precious metal in a component whereas the loss function for components with a height amount of precious metal is generally smaller than the loss functions for components with a small amount of precious metals. That would have the consequence that components…

In this approach the zero-one loss function is used:

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  | () | |  |

The decision process is based on discriminant functions. For the general case with risk, we can let because the maximum discriminant function will correspond to the minimum conditional risk. On decision level the vector is assigned to class if

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  | () | |  |

The discriminant function can be simplified by the fact that we can always multiply all the discriminant functions by the same positive constant or shift them by the same additive constant without influencing the decision. More generally, if we replace every by the , where is a monotonically increasing function, the resulting classification is unchanged (Duda, 2001).

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  | (13) | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  | (14) | |  |
|  |  |
| |  |  | | --- | --- | |  | (15) | |  |

The discriminant function can be simplified to

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  | (15) | |  |

The prior probability describes the probability that a randomly drawn part from the entire set of components is from component for and the part is an unknown component for. To estimate these values the distribution of components over all PCBs has to be determined. In this approach the prior probabilities were equally distributed over all component classes with probability.

* Prior problem

is the state-conditional probability density function for conditioned on being the true state of nature. Under the assumption that the soft-outputs from the classifier level fusion are statistically independent the joint probability density function is equal the products of the state-conditional probability density function of one feature from the feature vector.

* Explain independence

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  | (17) | |  |
| |  |  | | --- | --- | |  | (18) | |  |

This density functions are approximated by gamma distributions based on the test data outputs from the classification level.

The gamma distribution models sums of exponentially distributed random variables. The gamma distribution is defined as

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  | (16) | |  |

where is the gamma function (Mathworks-Gamma, 2014).

* Why gamma?

The gamma function is an extension of the factorial function with its arguments shifted down by 1, to real and complex numbers. The gamma function for all complex numbers is defined via a convergent improper integral:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | (17) |   (Wikipedia-Gamma, 2014).  The state-conditional probability density functions for the DIP14, DIP16 and Resistor network 1206 conditioned on the DIP14 component based on the test data are shown in…   |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  | (21) | |  |  |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  | (22) | |  |  |  |  |  |  | | --- | --- | --- | --- | | |  |  | | --- | --- | |  | (23) | |  | |  |

# Classification

Dsfdsf

## Feature extraction algorithms for electronic components

Sad

### 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.

|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  | () | |  |
| |  |  | | --- | --- | |  | () | |  |

The algorithm dependent resolution parameters are defined in Table 1.

Table : 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 : Image resolution

* Abbildung korrigieren

### 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 : DIP14 package with equidistant solder joints

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

|  |  |
| --- | --- |
|  | () |

and where 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.

|  |  |
| --- | --- |
|  | () |

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

|  |  |
| --- | --- |
|  | () |

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 8.2.

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

### 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.

|  |  |
| --- | --- |
| C:\Users\WIN\Masterthesis\Masterthesis\Masterarbeit_daten\2.4.2\tantal.png  Figure : Tantalum capacitor in RGB color model (left) and HSV color model (right) | Figure : Normalized histogram of hue channel (tantalum capacitor) |
| Figure : Normalized histogram of saturation channel (tantalum capacitor) | Figure : Normalized histogram of value channel (tantalum capacitor) |

### 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).

|  |  |
| --- | --- |
|  | () |
|  | () |

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

### 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.

#### Image reconstruction with PCA

A set of m part images each of size r x c is reshaped to a vectors of size r\*c x 1. First the mean vector **μ** and the covariance matrix **C** are computed for all vectors according to (15) and (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 with the k largest eigenvalues of the covariance matrix are used to construct the projection matrix **P** of size r\*c x k. The projection of an image vector into the eigenspace is given by

|  |  |
| --- | --- |
|  | () |

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 is the reconstructed image of the image .

|  |  |
| --- | --- |
|  | () |

The reconstruction error is defined by the euclidean distance between the image and its reconstructed image .

|  |  |
| --- | --- |
|  | () |

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).

#### 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 : 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 and the mean are computed from a subset of component images and the projection matrix and mean 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).

|  |  |
| --- | --- |
|  | () |

|  |  |
| --- | --- |
|  | () |

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).

|  |  |
| --- | --- |
|  | () |

The process is shown in Figure 21.

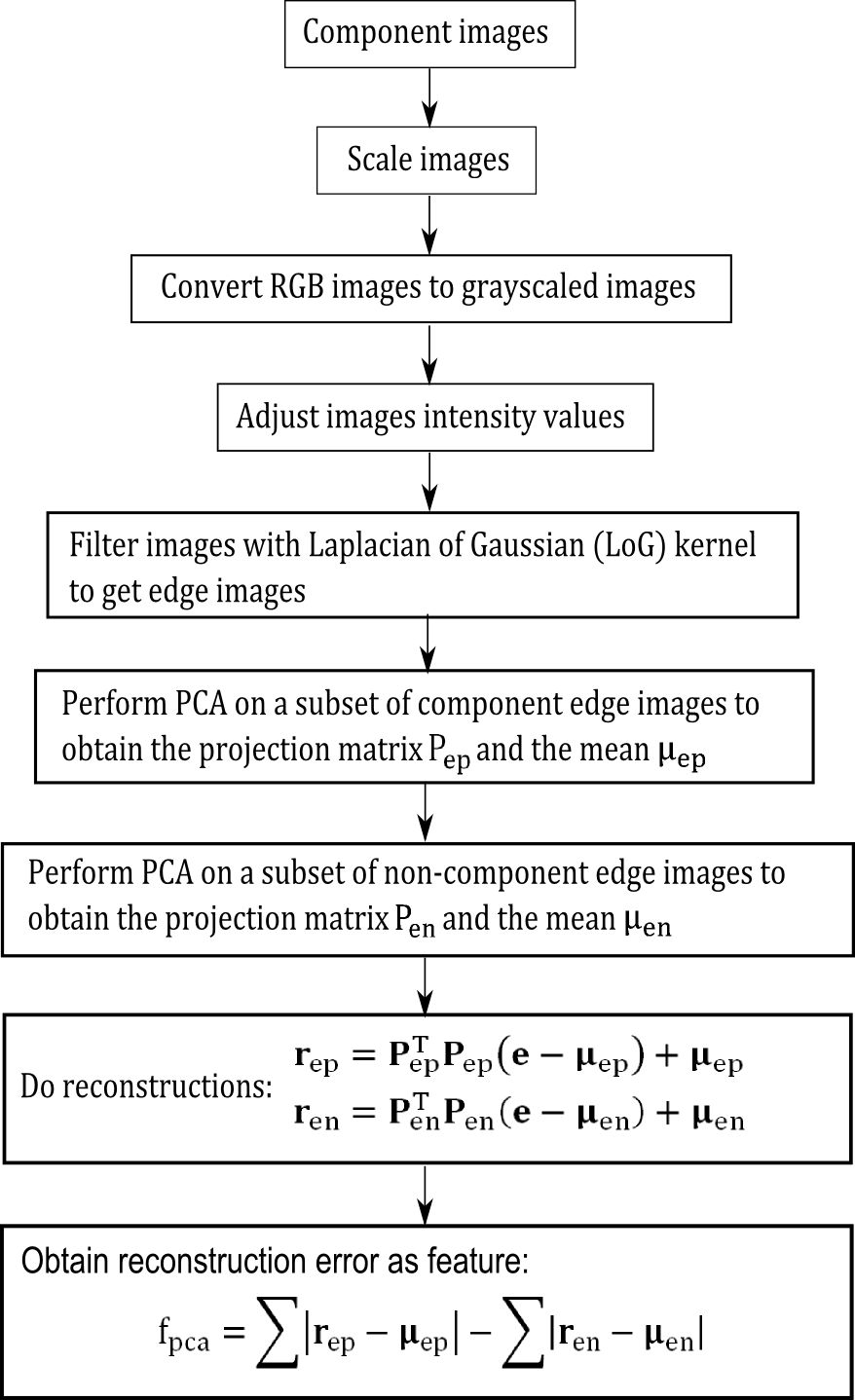


Figure : PCA feature construction process

## Feature selection and feature fusion techniques for classification

Asd

### Introduction to feature selection

Asd

* Wrapper, embedded
* Guyon

### 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 , is the number of samples in the subset labeled and c is the number of classes. The Fisher score of the -th feature is computed in (23).

|  |  |
| --- | --- |
|  | () |

Where is the standard deviation and the mean of the whole data set corresponding to the -th feature and is the -th feature of the sample .

|  |  |
| --- | --- |
|  | () |
|  | () |
|  | () |

|  |  |
| --- | --- |
|  | () |

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

### 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 -th tree (oob-samples). Each sample that was left out to construct the tree is predicted by the -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 -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

### 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 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.

– change plot

### Survey of the most important features

dfg

Asdf

## Random forest classifier

### 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 inducer. It is well known that ensemble methods can improve the prediction performance (Rokach, 2010).

### Introduction to Random forest ensemble classifier

The random forest is an ensemble classifier where the individual classifiers are unpruned tree predictors. The training algorithm of random forest applies bagging (bootstrap aggregating) for tree learning.

* Cart
* breiman

### Random forest training

Given a training set with response , bagging repeatedly selects bootstrap samples of the training set and fits trees to the samples. For each tree in the random forest classifier selects k random training samples (bootstrap samples) from the training set and trains the bagging trees on and . The optimal number of trees in the random forest depends on the size and structure of the data. In general a few hundred to several thousand trees are used whereas the generalization error for forests converges to a limit as the number of trees becomes large (Breiman, Random Forests, 2001). In random forests at each candidate split a random subset of features is selected. Typically for a dataset with p features features are used in each split (Random forest, 2014).

### Random forest prediction

The random forest prediction of a sample is done by predicting each trained tree in the random forest and averaging the prediction results over all trees. The output of the random forest can be normalized by the number of trees and interpreted as a soft-output probability. The prediction output is shown in (28) whereas is the number of trees in the forest and the trained tree (Random forest, 2014).

|  |  |
| --- | --- |
|  | () |
|  |  |

### Out-of-bag (oob) estimation

To train a -th tree a random subset of training samples is used to construct the tree whereas each tree uses different bootstrap samples. The samples that are not used to construct the -th tree are predicted by the -th tree to get a classification. The estimation is calld out-of-bag estimation. In this way, a test set classification is obtained for each case. At the end of the run, take j to be the class that got most of the votes every time case n was oob. The proportion of times that j is not equal to the true class of n averaged over all classes is the oob error estimate (Breiman, www.stat.berkeley.edu).

* overestimate

## Support vector machine classifier

zhh

### Linear Support vector machine

ijj

### RBF Support vector machine

Asdf

Asf

# Optical character recognition of electronic component marking

The optical character recognition (OCR) of printed text is widely studied and used in numerous applications like book scanning for digitalization, data entry for business documents, check and passport or license plate recognition. The automatic inspection of IC markings is a field which mainly focuses on inspection and quality control of PCB assembly processes. Inappropriate placement of chips and surface mounted devices (SMDs) can automatically be detected and corrected (B. Luo, Y.Gao, Z.Sun, 2013). This approach is focusing on the inspection of IC markings whereas makings of other component like capacitors or coils are out of focus because of their complexity.

## Optical character recognition difficulties and limits

The difference between the inspection of IC markings of PCB assembly line lies in the quality of the ICs and there markings. Newly printed IC markings have much better quality than markings from ICs which can be found in electronic scrap. The following difficulties of the optical character recognition of IC markings are caused by the fact that the ICs are from PCB scrap but also universal for similar OCR tasks.

* Company logos or symbols in character lines
* Symbols for part orientation confuse OCR software
* Dirty disturb segmentation process
* Scratches disturb segmentation process
* Broken characters of IC markings
* Overwritten characters
* Skew IC markings
* Scraped IC markings
* Different character fonts and character size
* Uneven illumination based on shadows from height components beside the examined component

Some difficulties about IC marking recognition from electronic scrap are shown in Figure 22.



Figure : Difficulties of IC marking recognition

To refine the investigation of Optical character recognition of IC markings the following restriction limits were taken.

* The components which are used to investigate the optical character recognition of IC markings have a black (dark) surface and the markings are white (bright).
* Marking characters have a minimum height of 1.0mm
* Makings made by laser engraving are out of focus
* The IC markings have to be readable by humans

Parts that are out of that restriction are not used in the OCR dataset for IC marking inspection.

* Why preprocessing for tessseract and vision pro?

## Optical character recognition flow chart

* Assumption made for the algorithm (character size, baselines,…)

Each IC component has different requirements for the IC marking recognition algorithm what does it matter that the markings have different properties. Properties which have to be known for the algorithm and are stored in the component database are the region of interest (ROI) for the IC marking and the subset of characters making up the marking. For the SMD resistor 1206 component for example the character subset could be {“0”, “1”, “2”, “3”, “4”, “5”, “6”, “7”, “8”, “9”, “R”} because smaller character subsets increases the recognition rate. The marking recognition flow chart is shown in Figure 24.

The input of the process is the already recognized part image. At first the OCR-ROI is selected from the part image to reduce the character search space and cut component solder joints and component boundaries. The RGB-image is converted in grayscale image caused by the fact that the characters are white (bright) and the character background is black (dark). Median filtering is applied to reduce noise mainly salt and pepper noise.

To emphasize the characters of the markings a Laplacian of Gaussian (LoG) filter is applied. The LoG kernel is a rotationally symmetric filter which is mainly used for edge detection. The filter is composed of the second derivative (Laplace operator) of a Gaussian filter shown in equations (30) and (29). The approximated discretized kernel mask is of size h x h where h is in pixel. In this approach the kernel size is changing linear with the image scale so that the kernel mask size is constant 1mm and in practice lies between 50 and 120 pixels. The standard deviation of the Gaussian is constant.

* Explane LoG operator

|  |  |
| --- | --- |
|  | () |
|  | () |

The next step is the blob segmentation which is done by Otsu’s segmentation method (OTSU, 1978). Otsu method is a segmentation process based on a global segmentation threshold which is computed by minimizing the intra-class variance (variance within classes). After segmentation step a morphologic closing operator is applied to reduce holes in the character blobs. The size of the rectangular closing kernel (h x h) changes linear with the image scale .

After the segmentation process blobs that do not correspond to a character still exist. Therefore the areas of the blobs are estimated and blobs with an area smaller and blobs with an area greater are rejected. The next step is the rough determination of possible lower character baselines. The y coordinate of the lower right corner of the blobs bounding box is used as samples to find upper baselines. This is done by computing the probability density estimate which is done with the matlab function ksdensity. The function returns a probability density estimate for samples based on a normal kernel function and is evaluated at equally spaced points that cover the range of the data (ksdensity, mathworks, 2014). In this approach 1000 equally spaced points from zero to one are used whereas the samples are normalized by the height of the image, the smoothing parameter is set to 0.025. All local maximums in the probability density function are potential lower character baselines.

After demining potential character baselines the blobs are assigned to the baselines based on the distance threshold. All characters that distance from baseline is shorter than are assigned to the baseline as potential characters of the baseline. To remove manufacturer symbols or dirt that are segmented as potential characters, baselines with a number of assigned blobs less than or equal two are removed together with their assigned blobs. This assumption is based on the condition that part names usually consist of more than two characters.

To remove blobs that correspond to a baseline but are no characters the RANSAC outlier detection approach is used to estimate baseline models and select all characters that fit the baseline model with a distance error from the baseline smaller . This is done with the lower and upper baseline of the character lines.

* RANSAC

Once again baselines with a number of assigned blobs less than or equal two are removed together with their assigned blobs.

In the next step the characters which are assigned to baselines are segmented in character lines (words). These words are transferred as an image to the character recognition software Tesseract or Cognex Vision Pro. The output of these software are the recognized characters of the word image. A comparison of the two OCR programs Tesseract and Cognec Vision Pro is done in . The settings and difficulties of the software is mentioned in 5.3. The recognized words are composed to labels whereas each label is a potential part name.



Figure : Label composition from words

Potential part names are requested by the Octopart API ([www.octopart.com](http://www.octopart.com)) by sending the composed labels. After making a label request, the Octopart API sends back a list of potential part names located in their database which could correspond to the requested label. The distance between the potential part names and the requested label is determined. The distance measure is the levenshtein distance which assigns a distance to two words based on their similarity. This is done with all labels of the marking and the potential part name with the smallest distance to requested label is assigned as part name to the part.

* levenshtein distance



Figure : IC marking recognition flow chart

asd

## Optical character recognition with Tesseract and Cognex Vision Pro software

The most important step of this OCR approach is the character recognition step where the binarized image of characters is mapped to the recognized ASCCI characters. Therefore the two OCR programs “Tesseract” and “Cognex Vision Pro” were used and compared based on the Electronic component marking recognition problem. The software Tesseract was already used in mobile IC Package Recognition (Patrick Blaes, Chris Young). For OCR engines without a-priori knowledge about the OCR task it is pretty difficult to identify electronic marking. To get a suitable recognition result the preprocessing steps in the flow chart in Figure 24 were carried out.

### Tesseract OCR engine

Tesseract is an open-source OCR engine that was developed by HP between 1984 and 1994. The program is written in C and C++ and can be used on varies platforms. Since 2006 Tesseract development was sponsored by Google and provides support for various languages. A comparison between Tesseract 3.0.1 and FineReader10 Corporation Edition from ABBYY shows that there is no significant difference in accuracy between both software engines. The differences in accuracy depend on quality and font of the characters whereas each engine has its advantages and disadvantages (Helinski, 2012).

For character recognition with Tesseract, the markings were decomposed in lines referring to the flow chart in Figure 24. The segmented binarized character line images where transferred to tesseract engine by the command-line interface in matlab and the recognized results were stored in a text file. Tesseract is an already trained OCR engine and therefore can directly be used without training. The following settings were made to improve the accuracy rate.

* Character limitation subset was set to “0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ/”
* Tesseract pagesegmode: 7 = Treat the image as single text line

### Cognex VisionPro® OCRmax engine

is a font-trainable OCR and OCV (Optical character recognition and Optical character verification) tool from the Cognex VisionPro® software suite for image processing (VisionPro, 2014). In this approach the OCR engine was just like Tesseract used to recognize characters from segmented binarized character line images. The main difference between the engine and Tesseract is the fact that OCRMax has to be trained before it can be used. Therefore a training data set was composed consisting of electronic part markings. The Software was trained with 1500 characters from 84 IC markings. The following settings were made to improve the accuracy rate.

* Character limitation subset was set to “0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ/”
* Einstellungen beschreiben

## Electronic part label verification based on Octopart database

Octopart is a company that offers an electronic part database with structured data for more than 30 million parts. The octopart tools facilitate to search parts across thousands of suppliers. An easy way to access the database is the Octopart API which provides information about up-to-date pricing and availability information, datasheets, compliance documents and technical specs for electronic components from distributors and manufacturers. Octopart allows access to information from more than 100 distributors including Digi-Key, Mouser, Newark, Premier farnell, Arrow, RS Component, Future electronics, grainger and many others (Octopart, 2014).

This tool was used for part name verification in which the recognized labels from OCR engines (Tesseract, OCRMax) were requested to the Octopart API. The response of the API is a list of equal or similar written part names provided from different suppliers. To assign a part name from the obtained list to the recognized label, the Levenshtein distance between the part names and the requested label is computed. The part name with the smallest distance less than or equal the distance threshold is assigned to the part. The requests were made with the data transfer tool curl in matlab.

# Experimental results

## Implementation

## 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 part border, border pixels are also selected from the printed circuit board images as can be seen in Figure 25. Additional important information and properties of the component are listed in Table 2.

Table : 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 : Component border definition

A section of the component database is shown in Figure 26.



Figure : Database section

### 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 27.



Figure : Image acquisition system

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

### Dataset composition

The dataset used in the experiments consist of 2000 parts from 12 component classes. The dataset composition is shown in Table 3.

Table : Dataset composition

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Number of**  **component images** | **Number of training data** | **Number of test data** |
| **Tantalum capacitor** |  |  |  |
| **SMD Aluminum electrolytic capacitor** |  |  |  |
| **QFP100** |  |  |  |
| **SMD Resistor Network array 1206, 4 Resistors** |  |  |  |
| **SMD Transistor SOT23-3** |  |  |  |
| **DIP14** |  |  |  |
| **DIP16** |  |  |  |
| **SMD Resistor 1206** |  |  |  |
| **SOIC-8** |  |  |  |
| **Ceramic capacitor 1210** |  |  |  |
| **SOT223-3** |  |  |  |
| **SMD Resistor 0806** |  |  |  |

## Feature selection results

The out-of-bag error depends on the number of random forest trees. The oob-error depending on the number of trees for 3136 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), the blue one the out-of-bag error from the random forest feature selection (RF) and the green one the out-of-bag error from fisher score (FS) feature selection with 235 selected features. The graphs show that the error rate of the FS+RF feature selection approach decreases faster and becomes smaller compared to the others whereas the oob-error does not show a big difference between the algorithm what indicates that the samples tend to be well linearly separable.



Figure : A comparison of different feature selection approaches

In this approach the feature selection algorithm based on Fisher score and Random forest described in 4.2.4 was used to select a subset of important features for classification. The most important features depend on the component therefore feature selection was applied to each component dataset. A list of the most important features from all examined components is shown in Appendix A, (Table 7). The numbers in the table correspond to the feature numbers of the associated feature set. The assignment of the features numbers to the features is explained in .

### Survey of most important features

Several selected important features are examined in detail to understand and confirm their importance for specific components.

#### Fourier features

The second most important feature of the SMD Resistor Network array 1206 is the second Frequency feature. The feature is the real part of the frequency coefficient with period of image high. It is the amplitude of the cosine transform in vertical direction. The mainly black region in the middle of the resistor is clearly visible. Toward the vertical image border the intensity becomes brighter caused by the reflective solder joints. This intensity gradient is typical for the resistor network and the curve correspond to the cosine curve of the second frequency feature. The elementary image of the frequency is shown in Figure 29.



Figure : Resistor network 1206 and the most significant real part elementary image

The values have been linearly scaled to vary between 0 (black and 255 (white).

#### Color features

The most important feature of the tantalum capacitor is a color feature which makes sense under the knowledge that the tantalum capacitor is a yellow-orange colored component and very different from the colors of other components or image regions in the PCB image.

* Bild + erläuterung

#### Segment features

The second most important feature of the Ceramic capacitor 1206 is the seventh segment feature. The seventh segment feature is the vertical component of the center of gravity from the segment which was produced by the region growing approach with the seed point at the seed position y=1.7mm, x=0.26mm. The brown/orange segment in the middle of the capacitor is significant for the component. Compared to other components the probability that a seed point located near the image border produces a segment with the center of gravity in the middle of the image is much smaller. The red marker shows the seed point of the segment which was produced by the region growing approach. The blue marker is the center of gravity from the segment. The vertical component of the center of gravity is the second most important feature for the ceramic capacitor.



Figure : Most important segment and seed point from ceramic capacitor

#### PCA reconstruction feature

The most important feature of the SMD Aluminum electrolytic capacitor is the PCA-reconstruction feature. That can be explained by looking at the circular border of the cylindrical part. The rounded border reflects the light almost independent from the beam angle of the illumination. That forms a bright shiny circle that is striking in the Laplacian of Gaussian (LoG) filtered images and can be efficiently be compressed into the part image PCs. A LoG filtered edge image of the SMD Aluminum electrolytic capacitor and the unit matrix projection into the PCs is shown in Figure 29.



Figure : SMD Electrolyte capacitor (top, left), SMD Electrolyte capacitor edge image (top, right), unit matrix projection into component PCs (bottom, left), unit matrix projection into non-component PCs (bottom, right)

## Classification results

gsdfg

### Ones-vs.-rest classification result

The One-vs.-rest classification strategy is based on the approach that for each component a classifier is trained and tested. The training set and test set consist of part images and non-part images.

There are two approaches to select the non-part images whereas the first approach is based on the idea that the part detection algorithm works in a way that the algorithm detects almost all parts in the PCB image and that most of the parts are in the database. Under these requirements the non-part images consist of images from parts of different components. The second approach is based on the idea that the non-part images should represent a plurality of possible images and therefore the non-part images are arbitrary selected image sections from the PCB images. An example of both approaches for the DIP14 component is shown in Table 4.

Table : Dataset approaches for non-part images

|  |  |  |
| --- | --- | --- |
| Part images for DIP14 | Non-part images for DIP14 (images from different parts) | Non-part images for DIP14  (images from arbitrary image section) |
| C:\Users\WIN\Masterthesis\Masterthesis\Masterarbeit_daten\6.3.1\part2.png | C:\Users\WIN\Masterthesis\Masterthesis\Masterarbeit_daten\6.3.1\part1.png | C:\Users\WIN\Masterthesis\Masterthesis\Masterarbeit_daten\6.3.1\part4.png |
| C:\Users\WIN\Masterthesis\Masterthesis\Masterarbeit_daten\6.3.1\part6.png | C:\Users\WIN\Masterthesis\Masterthesis\Masterarbeit_daten\6.3.1\part3.png | C:\Users\WIN\Masterthesis\Masterthesis\Masterarbeit_daten\6.3.1\part5.png |

Both approaches use the same number of part images and non-part images and have advantages and disadvantages with respect to the representativeness of the data. If the non-part images consist of only images from different components than the variance of the non-part image set is smaller and the accuracy should be better greater. On the other hand for the classifier it is more difficult to handle non-part images from components that were not in the training set or images on which no part is seen.

* bbox
* continue

#### Ones-vs.-rest classification results with Random forest classifier

Five random forest classifiers were trained whereas the first to fourth are based on the four selected feature sets from Frequency features, Color features, Segment features and PCA-reconstruction features which were extracted from the four feature extraction algorithms described to chapter 4.1. The accuracy for the random forest classifier from the mean accuracies of all twelve components is shown in Table 5. A detailed breakdown can be found in Appendix A.

The used random forest parameters are as followed:

* Number of random forest trees: 100

Table : Random forest classification results

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Frequency features** | **Color features** | **Segment features** | **PCA reconstruction features** | **Features selection from all feature sets** |
| **Average recognition accuracy of all Components** | True  positive |  |  |  |  |  |
| False  positive |  |  |  |  |  |

The results for the One-vs.-rest classification strategy is shown in .

Sdf

### Multiclass classification result

dfg

## Optical character recognition results

To evaluate the optical character recognition results different recognition levels are defined. The lowest one is the character level where

### Optical character recognition result on character level

To evaluate the optical character recognition results,

### Optical character recognition on word level

### Optical character recognition results on label level

sdf

### Octopart based part name assignment

Asd

# Life-cycle inventory analyses of printed circuit boards

Asdf

## Categorization of WEEE and recycling potential of PCB waste

### Recycling potential of electronic parts from PCB waste

### Reuse potential of electronic parts from PCB waste

Safd

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

dftg

### PCB support material (epoxy)

Dsf

### Detected and not correctly classified electronic parts

Fdg

### Detected and correctly classified electronic parts

Dfg

### Detected, correctly classified and label recognized electronic parts

Jhg

## Printed circuit board LCI model

Life cycle inventory (LCI) is a process of quantifying energy and raw material requirements, atmospheric emissions, waterborne emissions, solid wastes, and other releases for the entire life cycle of a product, process, or activity (Curran, 2006). An LCI is the basis of an Life cycle impact assessment (LCA) to evaluate comparative environmental impacts or potential improvements. With respect of reuse and recycling an LCI can assist organizations in comparing products or processes and considering environmental factors in material recycling. The “Guidelines for Assessing the Quality of Life Cycle Inventory Analysis” (Lynda Wynn, Eugene Lee, 1995) provides a framework for performing an inventory analysis. Four steps are defined for making a life cycle inventory:

1. Develop a flow diagram of the process being evaluated
2. Develop a data collection plan
3. Collect data
4. Evaluate and report results

### PCB flow diagram

The LCI-model in this approach is a generalize model for Printed circuit boards. It is developed to handle PCBs from scrap automatically. The quality of the LCI results depends on the composition of the specific PCB and is described in more detail in . In this work two different models are created. The first model represents the LCI model of the PCB and uses full aggregated data to quantify energy and raw material requirements, emissions, solid waste and other releases.

* Erklärung keine richtiges LCI-model
* Assembly line modellierem

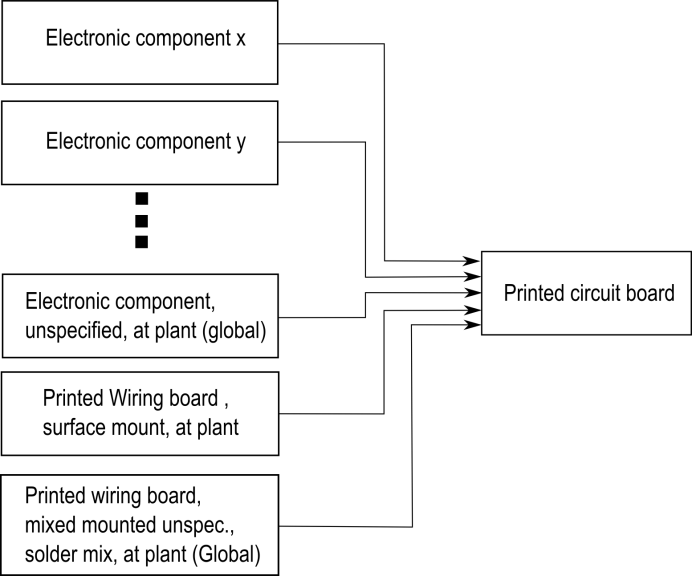


Figure : PCB flow diagram for LCI-model

The flow diagram for a PCB is shown in Figure 32. The PCB is consists of four different PCB regions:

* Gabi check Bestückung modellieren (system boundyries)

1. Detected part, recognized packages: Modeled by the Electronic component x, Electronic component y and so on.
2. Detected part, unrecognized package: Modeled by the Electronic component unspecified, at plant (global)
3. Recognized printed wiring board surface: Modeled by Printed wiring board surface mount, at plant
4. Unknown PCB regions: Modeled by Printed wiring board mixed mounted unspec., solder mix, at plant (global)

The second model represents the material composition of the PCB. This model is of interest for recycling organizations to analyze the content of precious metals or other valuable resources. Moreover the amount of hazard materials in the specific PCB can be analyzed and specially treated. The flow diagram of the PCB composition model is shown in Figure 33. The flows in the figure between the PCB components and the Materials are symbolic and depend on the content of the composition of the components.

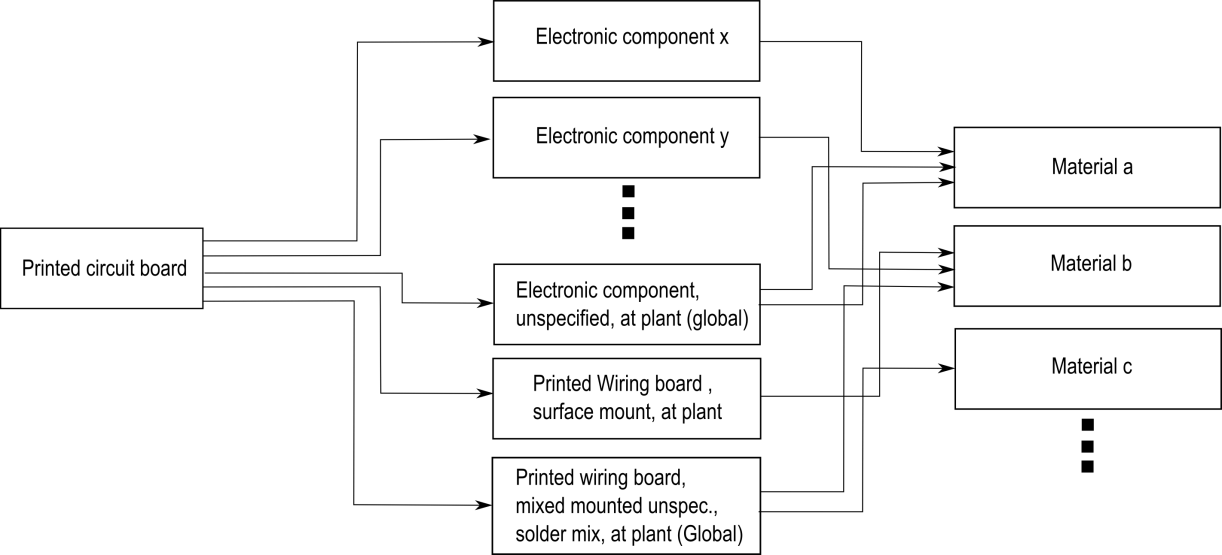


Figure : PCB flow diagram for composition model

### Data collection plan and data collection

The data collection is mainly based on the GaBi Extension database XI: Electronics from PE INTERNATIONAL. For each electronic component in the recognition database an ILCD package model is assigned.

* flowproperties

Detected and correctly classified electronic parts are modeled by ILCD component models. If the ILCD component model exists in the GaBi database it is used in the recognition database. If a component is not modeled in the GaBi database but a similar model which differs mainly in size, the amount of the component is scaled by mass and assigned to the component in the recognition database see formula (31).

|  |  |
| --- | --- |
|  | () |

Electronic parts which are detected but the package could not be classified correctly are modeled by the “Electronic component, unspecified, at plant (global)” woher? and assigned to the component in the recognition database.

PCB regions where PCB support material (epoxy) was detected are modeled by “Printed wiring board surface mount, at plant”. The amount unit is mass and is calculated by the region area recognized in the image and the basis weight. The basis weight is based on the information on <http://www.leiton.de> (Leiton: leiton-tools-gewichtsberechnung, 2014).

|  |  |
| --- | --- |
|  | () |
|  |  |

PCB regions where an assignment of a component or PCB support material could not be made are modeled by “Printed wiring board mixed mounted unspecific, solder mix, at plant (global)”. The amount is calculated based on the basis weight which was determined by the average value of 25 PCBs which is listen in detail in Appendix A.

|  |  |
| --- | --- |
|  | () |
|  |  |

### Evaluation and results

The results of the two models are different in a way that the composition model quantifies the materials which make up the PCB. Parts with a high amount of precious metals or other valuable materials for recycling can be determined and detached. The separate treatment can increase the concentration of valuable materials in the separated electronic scrap and therefor is an important factor for an economic recycling process. The increase of concentration for some valuable materials is discussed in detail in chapter 7.5.

The LCI-model quantifies energy and raw material requirements, atmospheric emissions, waterborne emissions, solid wastes, and other releases. It can be used to discover PCB boards or electronic parts containing hazard materials that can be specially treaded.

The material composition model and the LCI-model for the Arduino Due board are drawn up in chapter 7.7.

## GaBi-Software and LCI data availability of electronic components

Saf

## Increasing of precious metal concentration by selective dismantling

gh

### Increasing of tantalum concentration by selective dismantling

hg

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

Sdf

## Arduino Due board LCI-model

# Conclusion and prospects

## Real time PCB board inspection

## Feature extraction based on Wavelet basis functions

dsf

# Appendix A

Table : Components in database

|  |  |
| --- | --- |
| Component name and description | Component image |
| **Tantalum capacitor**  - Package: EIA Code 2412  - Color: yellow/orange  - Tantalum and aluminum electrolytic capacitor with solid electrolyte polarity markings | C:\Users\WIN\Masterthesis\Masterthesis\Masterarbeit_daten\6.1\Bib1.png |
| **SMD Aluminum electrolytic capacitor**   * Diameter: 6.5mm | C:\Users\WIN\Dropbox\Masterthesis\6.1\Bib2.png |
| **QFP100**   * Package: QFP100 * 23.4mm x 17.4mm | C:\Users\WIN\Dropbox\Masterthesis\6.1\Bib5.png |
| **SMD Resistor Network array 1206, 4 Resistors**   * Long Side Terminals * Four resistors | C:\Users\WIN\Dropbox\Masterthesis\6.1\Bib6.png |
| **SMD Transistor SOT23-3**   * Package: SOT23-3 * 3.0mm x 2.6mm | C:\Users\WIN\Dropbox\Masterthesis\6.1\Bib7.png |
| **DIP14**   * Package: DIP14 * 19.5mm x 7.6mm | C:\Users\WIN\Dropbox\Masterthesis\6.1\Bib8.png |
| **DIP16**   * Package: DIP14 * 19.5mm x 7.6mm | C:\Users\WIN\Dropbox\Masterthesis\6.1\Bib9.png |
| **SMD Resistor 1206**   * Imperial code: 1206 (3216 metric) * 3.2mm x 1.6mm | C:\Users\WIN\Dropbox\Masterthesis\6.1\Bib11.png |
| **SOIC-8**   * Package: SOIC8 * 5.0mm x 6.2mm | C:\Users\WIN\Dropbox\Masterthesis\6.1\Bib14.png |
| **Ceramic capacitor 1210**   * Imperial code: 1210 (3225 metric) * 3.2mm x 2.5mm * Color: brown/orange | C:\Users\WIN\Dropbox\Masterthesis\6.1\C1.png |
| **SOT223-3**   * Package: SOT223-3 * 6.5mm x 7.0mm | C:\Users\WIN\Dropbox\Masterthesis\6.1\T1.png |
| **Quartz** |  |
| **PCI** |  |

Table : Most important selected features

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Most important Frequency features** | **Most important Color features** | **Most important Segment features** | **Most important PCA reconstruction features** | **Most important Features selection from all feature sets** |
| **Tantalum capacitor** | 34, 45, |  |  |  |  |
| **SMD Aluminum electrolytic capacitor** |  |  |  |  |  |
| **QFP100** |  |  |  |  |  |
| **SMD Resistor Network array 1206, 4 Resistors** |  |  |  |  |  |
| **SMD Transistor SOT23-3** |  |  |  |  |  |
| **DIP14** |  |  |  |  |  |
| **DIP16** |  |  |  |  |  |
| **SMD Resistor 1206** |  |  |  |  |  |
| **SOIC-8** |  |  |  |  |  |
| **Ceramic capacitor 1210** |  |  |  |  |  |
| **SOT223-3** |  |  |  |  |  |
| **SMD Resistor 0806** |  |  |  |  |  |

Table : Random forest Classification results (comprehensive)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Frequency features** | **Color features** | **Segment features** | **PCA reconstruction feature** | **Features selection from all features sets** |
| **Tantalum capacitor** | True  positive | 114/116 (98.3%) |  |  |  | 112/111 (98.3%) |
| False  positive | 108/112 (98.3%) |  |  |  | 108/112 (98.3%) |
| **SMD Aluminum electrolytic capacitor** | True  positive | 114/116 (98.3%) |  |  |  | 112/111 (98.3%) |
| False  positive |  |  |  |  |  |
| **QFP100** | True  positive |  |  |  |  |  |
| False  positive |  |  |  |  |  |
| **SMD Resistor Network array 1206, 4 Resistors** | True  positive |  |  |  |  |  |
| False  positive |  |  |  |  |  |
| **SMD Transistor SOT23-3** | True  positive |  |  |  |  |  |
| False  positive |  |  |  |  |  |
| **DIP14** | True  positive |  |  |  |  |  |
| False  positive |  |  |  |  |  |
| **DIP16** | True  positive |  |  |  |  |  |
| False  positive |  |  |  |  |  |
| **SMD Resistor 1206** | True  positive |  |  |  |  |  |
| False  positive |  |  |  |  |  |
| **SOIC-8** | True  positive |  |  |  |  |  |
| False  positive |  |  |  |  |  |
| **Ceramic capacitor 1210** | True  positive |  |  |  |  |  |
| False  positive |  |  |  |  |  |
| **SOT223-3** | True  positive |  |  |  |  |  |
| False  positive |  |  |  |  |  |
| **SMD Resistor 0806** | True  positive |  |  |  |  |  |
| False  positive |  |  |  |  |  |

Table : Basis weight determination (PCB mounted)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Length** | **Width** | **Weight** | **Area** | **Basis weight** |
| 26 | 23 | 450 | 598 | 0.752508361 |
| 17 | 5.5 | 110 | 93.5 | 1.176470588 |
| 31 | 24 | 670 | 744 | 0.900537634 |
| 14 | 19 | 110 | 266 | 0.413533835 |
| 23 | 10 | 160 | 230 | 0.695652174 |
| 19 | 14 | 110 | 266 | 0.413533835 |
| 11 | 25 | 170 | 275 | 0.618181818 |
| 31 | 24 | 620 | 744 | 0.833333333 |
| 24 | 24 | 400 | 576 | 0.694444444 |
| 24 | 16 | 250 | 384 | 0.651041667 |
| 20 | 14 | 145 | 280 | 0.517857143 |
| 24 | 19 | 440 | 456 | 0.964912281 |
| 19 | 14 | 200 | 266 | 0.751879699 |
| 27 | 15 | 275 | 405 | 0.679012346 |
| 17 | 8.5 | 120 | 144.5 | 0.830449827 |
| 13 | 10 | 90 | 130 | 0.692307692 |
| 30.5 | 22 | 600 | 671 | 0.894187779 |
| 16 | 16 | 150 | 256 | 0.5859375 |
| 8.5 | 5.5 | 35 | 46.75 | 0.748663102 |
| 14 | 5.5 | 70 | 77 | 0.909090909 |
| 12 | 7 | 70 | 84 | 0.833333333 |
| 19 | 14 | 105 | 266 | 0.394736842 |
| 18 | 10 | 150 | 180 | 0.833333333 |
| 17 | 10 | 200 | 170 | 1.176470588 |
|  |  | **5700** | **7608.75** | **0.749137506** |