

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

27.02.2015

Danksagung

An dieser Stelle möchte ich mich herzlich bei allen bedanken, die mich bei der Anfertigung dieser Arbeit unterstützt haben.

Zuerst möchte ich mich bei den Betreuern meiner Masterarbeit Dipl.-Ing. Hendrik Grosser (Fraunhofer IPK) und Dr.-Ing. Perrine Chancerell (TU Berlin) für die hilfreichen Anregungen und konstruktiven Kritiken bei der Erstellung dieser Arbeit bedanken.

Mein Dank gilt Prof. Dr.-Ing. Rainer Stark und allen Mitarbeitern der Abteilung Virtuelle Produktentstehung am Fraunhofer Instituts für Produktionsanlagen und Konstruktionstechnik.

Außerdem einen herzlichen Dank an alle Mitarbeiter der Abteilung Environmental and Reliability Engineering des Fraunhofer Instituts für Zuverlässigkeit und Mikrointegration für die Bereitstellung von Arbeitsplätzen und entsprechender Software. Besonderer Dank gilt Antonia Ritschel für die durchgeführten Messungen zu Bauteilzusammensetzungen.

Bedanken möchte ich mich bei Herrn Norbert Storch, den Betriebsleiter der BRAL Reststoff-Bearbeitungs GmbH für das Experteninterview.

Großer Dank gilt Johanna, Christian, Marc, Annika, Felix und Julia für die inhaltliche und sprachliche Korrektur der Arbeit.

Ein besonderer Dank gilt meinen Eltern und meinen Geschwistern die mich bei meiner Arbeit unterstützt haben und auf die ich mich immer verlassen kann.

Abstract

Abstract

The recovery and recycling of precious metals, rare earth materials and other critical materials from electronic PCB waste is of ecological and economical interest. Electronic PCBs contain a high concentration of critical materials which are processed in today's recycling chains by shredding or smelting. A high amount of critical material is not recycled because of economical, thermo dynamical, physical and chemical reasons. A higher recycling rate can be achieved by selective material composition estimation and dismantling process which is not feasible with today's recycling procedures. Such a process would also support the reuse and upgrade of electronic components.

In this thesis a software demonstrator, for the automatic evaluation of 2D images from PCBs with their components and for determination of material compositions is developed. Therefore a data fusion model for electronic component detection and classification was created. The data fusion model consists of algorithms for feature extraction from different feature domains with the goal to extract significant feature for electronic component package classification. The feature domains are based on package features such as package color, package color segments, package form and frequency spectrum of the package image. Important features are selected by a package specific feature selection. The classification evaluation is based on a regenerated database with 2D images of package references.

In the next step the exact electronic component name is determined by evaluating the electronic component markings. Therefore an OCR algorithm is developed which verifies the component names based on an electronic component name database.

To evaluate the content of critical materials of a PCB and its components, a Life-cycle-inventory model of the PCB is automatically generated based on the recognized electronic components with the data fusion model. The ILCD-Format (International Reference Life Cycle Data System) is used to model the LCI-model data and material composition data for each component and merge them to a PCB model which can be imported in common LCA software (Life-cycle assessment software) like GaBi or OpenLCA.

Table of Contents

Abstract.....	ii
Table of Contents.....	iii
List of Figures	viii
List of Tables	xii
List of Abbreviations	xiv
1. Introduction	1
1.1 Problem formulation.....	1
1.2 Purpose.....	4
2. Background Theories and related works	5
2.1 Feature extraction algorithms.....	6
2.1.1 Single seed region growing approach for color images.....	6
2.1.2 k-means clustering	8
2.1.3 Normalized cross correlation for 2D pattern matching.....	9
2.1.4 Image reconstruction with PCA	10
2.2 Feature selection.....	11
2.2.1 Fisher score	12
2.2.2 Random forest feature selection	13
2.3 Object Classification	14
2.3.1 Random forest classifier	14
2.3.2 Support vector machine classifier	16
2.4 Data fusion model	19

Table of Contents

Data fusion with Dempster-Shafer theory.....	20
2.5 Optical character recognition of IC markings from electronic PCB scrap.....	22
2.5.1 Levenshtein distance	22
2.5.2 RANSAC algorithm	23
2.5.3 Octopart database for component-name verification	24
2.6 Life Cycle Inventory (LCI) analysis	25
2.6.1 Categorization of WEEE and PCB waste	25
2.6.2 Recycling and reuse potential of electronic PCB waste.....	27
2.6.3 International Reference Life cycle Data System (ILCD) format	28
3. Methods for electronic component recognition	29
3.1 Image preprocessing	29
3.1.1 Image rotation correction.....	30
3.1.2 Scaling determination based on scaling symbol.....	32
3.1.3 Image resolution for feature extraction	36
3.2 Electronic component detection.....	37
3.2.1 PCB board segmentation	38
3.2.2 Color based PCB surface detection.....	39
3.2.3 Electronic component detection based on normalized 2D cross-correlation.....	43
3.3 Feature extraction.....	45
3.3.1 A priori knowledge generation	46
3.3.2 Fourier coefficients based feature extraction	47
3.3.3 Histogram based feature extraction.....	49
3.3.4 Segment based feature extraction	50
3.3.5 PCA reconstruction error based feature extraction	51

Table of Contents

3.4	Feature selection based on Fisher score and Random forest.....	53
3.5	Classification.....	54
3.5.1	Random forest classifier	55
3.5.2	Support vector machines.....	56
3.6	Data fusion model	56
3.6.1	Feature level fusion	57
3.6.2	Classifier level fusion.....	58
3.6.3	Decision level fusion with Dempster-Shafer theory	59
3.7	Optical character recognition of electronic component marking	64
3.7.1	Optical character recognition difficulties	64
3.7.2	Optical character recognition flow chart.....	66
3.7.3	Optical character recognition evaluation scheme.....	71
4.	Life-cycle inventory model analyses of printed circuit boards	76
4.1	Printed circuit board region classification based on electronic component recognition results.....	76
4.2	Estimated PCB-LCI model and PCB-composition model	77
4.3	Data collection plan and data collection.....	79
5.	Implementation and experiments	82
5.1	Dataset creation	82
5.1.1	Image acquisition	85
5.1.2	Dataset composition	86
5.2	PCB surface detection results	87
5.3	Feature selection results	89
5.3.1	Fourier features	90

Table of Contents

5.3.2	Color features	90
5.3.3	Segment features.....	91
5.3.4	PCA reconstruction feature	92
5.4	Classification results.....	92
5.4.1	Random forest classifier results.....	92
5.4.2	Support vector machine classifier results.....	94
5.5	Decision level fusion results with Dempster-Shafer theory	96
5.6	Optical character recognition results.....	97
5.6.1	Optical character recognition dataset and limits	97
5.6.2	Optical character recognition accuracy results on character level, word level, label level and part level.....	98
5.6.3	Octopart based component name assignment	99
5.6.4	Octopart based part price assignment	103
5.7	Life-cycle inventory analyses evaluation and results.....	104
5.7.1	GaBi-Software and LCI data availability of electronic components	104
5.7.2	Tantalum as an example for concentration increasing by selective dismantling	105
5.7.3	Arduino Due board LCI-model	106
6.	Discussion and future work	112
6.1	Electronic component detection.....	112
6.1.1	Electronic component detection based on 3D model	112
6.1.2	Electronic component detection based on Height map from laser triangulation	113
7.	Conclusion.....	115
7.1	Data fusion model for electronic component recognition	115
7.2	PCB material composition model estimation	117

Table of Contents

7.3	Electronic part name assignment for electronic part reuse	118
7.4	Application inclusion in the PCB recycling process chain	119
	Bibliography	123
Appendix A	Recognition database.....	CXXX
Appendix B	Random forest classification results	cxxxiii
Appendix C	Linear-SVM classification results.....	cxxxv
Appendix D	RBF-SVM classification results	cxxxvii
Appendix E	Decision level fusion results.....	cxxxix
Appendix F	Basis weight determination (PCB mounted).....	cxi
Appendix G	Arduino Due component replacement model	cxli
Appendix H	Arduino Due estimated part prices	cxliii
Appendix I	Material prices	cxlv
Appendix J	Improved WEEE recycling chain.....	cxlviii

List of Figures

Figure 1: Simplified recycling chain for WEEE	2
Figure 2: Mass balance of the preprocessing of 1,000 kg of input WEEE (Chancerel, et al., 2009).....	3
Figure 3: Connection between belief, disbelief, plausibility and doubt (Rakowsky, 2007)	21
Figure 4: RANSAC example (http://www.codeproject.com/KB/recipes/automatic_panoramas/ransac.png)	24
Figure 5: Transformation from lines in the image to points in the frequency domain (www.svi.nl/FourierTransform)	30
Figure 6: Image rotation correction process	31
Figure 7: Image rotated by 3.0 degree	32
Figure 8: Canny edge image of the rotated image	32
Figure 9: Shifted DFT of the rotated image (logarithmic representation)	32
Figure 10: Summed amplitude over angle (invariants by 90 degree)	32
Figure 11: Scale symbol.....	33
Figure 12: Scale symbol placed on the board	33
Figure 13: Scaling determination process.....	33
Figure 14: Value channel (brightness) of HSV color image.....	35
Figure 15: Cosine transform filtered image	35
Figure 16: Otsu thresholding	35
Figure 17: Blobs of the scaling symbol	35
Figure 18: Resolution dependency from component area and feature extraction algorithm....	37
Figure 19: PCB board segmentation process flow	38
Figure 20: Acquired PCB image.....	39
Figure 21: Otsu segmentation	39
Figure 22: Morphological eroded image with 10x10 kernel	39
Figure 23: Segmented PCB board image	39
Figure 24: PCB surface segmentation process flow.....	40
Figure 25: Original image	41
Figure 26: First 200 image segments based on region growing approach.....	41

List of Figures

Figure 27: PCB surface cluster pyramid	42
Figure 28: Image template for DIP14 component (RGB color space)	44
Figure 29: Spatial image resolution for 2D-cross correlation.....	44
Figure 30: SOT223 transistor	45
Figure 31: Determined potential component positions for SOT223 transistor	45
Figure 32: Seed point grid (30 seed points)	47
Figure 33: DIP14 package with equidistant solder joints	48
Figure 34: Tantalum capacitor in RGB color model (left) and HSV color model (right)	49
Figure 35: Normalized histogram of hue channel (tantalum capacitor)	49
Figure 36: Normalized histogram of saturation channel (tantalum capacitor).....	50
Figure 37: Normalized histogram of value channel (tantalum capacitor).....	50
Figure 38: Three important seed points from the priori knowledge generation.....	51
Figure 39: 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)	52
Figure 40: PCA feature construction process	53
Figure 41: Feature selection process chain	54
Figure 42: Data fusion model.....	57
Figure 43: Normal distribution of Resistor network classifier (positive test data)	60
Figure 44: Difficulties of IC marking recognition	65
Figure 45: OCR of a QFP144 from top left to top right: grayscaled image, LoG filtered image, binarized image, blobs filtered image. From bottom left to bottom right: four character lines (words)	68
Figure 46: IC marking recognition flow chart	69
Figure 47: Dependency of Tesseract character recognition accuracy from number of characters	71
Figure 48: OCR evaluation on character level.....	72
Figure 49: OCR evaluation on word level without Octopart	72

List of Figures

Figure 50: OCR evaluation on word level with Octopart	73
Figure 51: OCR evaluation on label level with Octopart.....	74
Figure 52: OCR evaluation on part level with Octopart.....	75
Figure 53: PCB model regions	77
Figure 54: PCB flow diagram for LCI-model	78
Figure 55: PCB flow diagram for composition model	79
Figure 56: Component border definition.....	84
Figure 57: Database section.....	84
Figure 58: Image acquisition system.....	85
Figure 59: Component dataset splitting	87
Figure 60: original PCB image	88
Figure 61: Sum of RBF-kernel SVM scores wx, y (grayvalues are scaled between -20 and 20) ..	88
Figure 62: A comparison of different feature selection approaches.....	89
Figure 63: Resistor network 1206 and the most significant real part elementary image.....	90
Figure 64: Tantalum capacitor and the most important histogram color features (HSV color space)	91
Figure 65: Most important segment and seed point from ceramic capacitor	91
Figure 66: 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).....	92
Figure 67: Dependence of the misclassification rate based on the number of trees (Resistor network, most important features from all feature domains).....	93
Figure 68: Dependency between the true positive and false positive rate from the number of features for the DIP14 component classifier and random forest classifier.....	94
Figure 69: Dependency of the error rate from the regularization constant C (Resistor 0806)....	95
Figure 70: Dependency of the error rate from the regularization constant C and kernel constant γ (Resistor 0806).....	96
Figure 71: Arduino Due board	106
Figure 72: Estimated material composition of Arduino Due components.....	108

List of Tables

Figure 73: Estimated material prices of Arduino Due components	109
Figure 74: Estimated Gold distribution over Arduino Due parts	110
Figure 75: Estimated Palladium distribution over Arduino Due components.....	110
Figure 76: Estimated Arduino Due component prices.....	111
Figure 77: 3D model based component detection	113
Figure 78: Principle of laser triangulation (Torsten Koch, 2013)	114
Figure 79: Improved pre-processing step in PCB recycling process chain	121
Figure 80: Improved recover and disposal step in PCB recycling process chain	122

List of Tables

Table 1: Feature extraction algorithm based resolution parameter	36
Table 2: Dataset approaches for non-component images	55
Table 3: Normal distribution outputs from outputs from classifier fusion level.....	62
Table 4: Basic probability assignments	63
Table 5: Belief and plausibility of component classes	64
Table 6: Component properties.....	83
Table 7: Dataset composition	86
Table 8: Confusion matrix of the predicted PCB surface training data	88
Table 9: Confusion matrix of the predicted PCB surface test data.....	88
Table 10: Random forest classification results	93
Table 11: Linear-SVM classification results	95
Table 12: RBF-SVM classification results	96
Table 13: OCR accuracy results.....	98
Table 14: Confusion matrix of the manual labeled words (word-level) verified with Octopart database.....	99
Table 15: Confusion matrix of the manual labeled labels (label-level) verified with Octopart database.....	100
Table 16: Accuracy rate of component assignment with manual labeled components on word level verified with Octopart database (part-level)	100
Table 17: Accuracy rate of part assignment with manual labeled parts on label level verified with Octopart database (part-level)	100
Table 18: Confusion matrix of the Tesseract recognized words (word-level) verified with Octopart database	101
Table 19: Confusion matrix of the Tesseract recognized labels (label-level) verified with Octopart database	101
Table 20: Accuracy rate of part assignment with Tesseract OCR engine on word level verified with Octopart database (part-level)	101

List of Tables

Table 21: Accuracy rate of part assignment with Tesseract OCR engine on label level verified with Octopart database (part-level)	102
Table 22: Confusion matrix of the OCRMax recognized words (word-level) verified with Octopart database	102
Table 23: Confusion matrix of the OCRMax recognized labels (label-level) verified with Octopart database.....	102
Table 24: Accuracy rate of part assignment with OCRMax OCR engine on word level verified with Octopart database (part-level)	102
Table 25: Accuracy rate of part assignment with OCRMax OCR engine on label level verified with Octopart database (part-level)	103
Table 26: Arduino Due parts of the LCI model	107

List of Abbreviations

AOI	
	Automatic optical inspection, 1, 4, 105, 120
API	
	Application programming interface, 24, 25, 85, 99, 119
B2B	
	Business-to-Business, 121
B2C	
	Business-to-Consumer, 121
DAI-DAO	
	data in-data out, 19
DAI-FEO	
	data in-feature out, 19
DFT	
	Discrete fourier transform, 30
DoD	
	Department of Defense, 19
DS	
	Dempster-Shafer, 20, 62
EEE	
	electric and electronic equipment, 1, 27
FEI-FEO	
	feature-in feature out, 19
FFT	
	Fast fourier transform, 31, 57, 58, 89
FN	
	False negative rate, 73
FP	
	False positive rate, 73, 74
FPA	
	False part assignment rate, 74, 100, 102, 103
FS	
	Feature selection, 89
IC	
	Integrated circuit, 22, 64, 65, 66, 69, 70, 97, 106, 107, 109, 110, 119, cxli, cxlii
ILCD	
	International Life Cycle Data System, ii, 28, 77, 78, 79, 80, 82, 84, 85, 104, 105, 106, 117
LCA	
	Life-cycle assessment, ii, 25, 28, 77, 78
LCI	
	Life cycle inventory, ii, 25, 28, 76, 77, 78, 84, 104, 105, 106, 107, 120
LCIA	
	Life cycle impact assessment, 28
LoG	
	Laplacian of Gaussian, 52, 66, 92
OCR	
	Optical character recognition, ii, 5, 22, 25, 29, 64, 65, 66, 68, 69, 70, 71, 72, 73, 74, 75, 83, 85, 97, 98, 99, 101, 102, 103, 118, 119
OCV	
	Optical character verification, 22, 70
OOB	
	Out-of-bag error, 12, 13, 14, 16, 89
PC	
	Principal Component, 52
PCA	
	Principal component analysis, 10, 36, 47, 51, 53, 57, 58, 84, 92, 93, 95, 96, cxxxiii, cxxxv, cxxxvii
PCBs	
	Printed circuit boards, ii, 1, 3, 4, 5, 26, 27, 28, 77, 80, 105, 111, 113, 119, 120
RBF	
	Radial basis function, 18, 19, 29, 42, 43, 56, 87, 88, 92, 95, 96, 116, 117, cxxxvii
RF	
	Random forest, 89
SMD	
	Surface-mounted device, 5, 49, 66, 86, 90, 92, 104, 107, cxxx, cxxxii, cxxxiii, cxxxiv, cxxxv, cxxxvi, cxxxvii, cxxxviii, cxli
SVM	
	Support vector machine, 14, 16, 18, 19, 29, 42, 43, 87, 88, 94, 95, 96, 116, cxxxv, cxxxvii
TN	

True negative rate, 73
TP
True positive rate, 73
TPA
True part assignment rate, 74, 100, 102,
103, 118
WEEE
waste electric and electronic waste, 1, 2,
3, 25, 26, 27, 119, 120, cxlviii, , ,

1. Introduction

Efficient recycling and reuse of waste electric and electronic waste (WEEE) requires detailed information about the material content and the electronic composition of PCBs (Printed circuit boards). An Automatic optical inspection system (AOI-System) of PCBs provides a good opportunity for estimating composition models of PCBs.

1.1 Problem formulation

The production of electric and electronic equipment (EEE) is increasing worldwide. At the end of the life the equipment ends up as waste electric and electronic waste (WEEE). This development requires an End-of-life management system which serves the following goals:

- Reduction of materials going to landfill, and minimization of landfill-volumes
- Recycling of materials in order to keep the maximum economic and environmental value and to avoid new material extraction
- Reduction of emissions of environmentally relevant substances, for example through leaching from landfill sites, incineration slags and off-gasses from combustion processes

(Huisman, 2004)

Recycling of WEEE is an important subject not only from the point of waste treatment but also from the recovery of valuable materials and the reuse of electronic components. WEEE is diverse and complex in terms of materials and components. Electronic products, in particular IT and communication equipment contains a lot of precious metals (gold, silver, palladium) and special metals (indium, selenium, tellurium, tantalum, bismuth, antimony). The precious metals are mainly found in printed circuit boards (PCBs). The concentration of precious metals in PCBs is usually much higher than the concentration in ores, especially for gold and palladium. Moreover the extraction of precious metals through mining is associated with negative environmental impacts through significant emissions of greenhouse gases and energy, water and land usage. Moreover the high economic value of precious metals on the world market as well as the limited available reserves of precious metals requires an improvement of recovering precious metals from WEEE. The proportion of PCBs in WEEE over different equipment type is

Introduction

around 9% (Chancerel, et al., 2009). The concentration of precious metals in unshredded printed circuit boards is around 669 g/t of silver, 135 g/t of gold and 50 g/t of Palladium. Other metals like tantalum are very rare recycled in today's recycling chains.

The reuse of electronic components is the best ecological way of treating electronic waste. Unfortunately the determination of economical valuable electronic components which can be reused is not done in today recycling chains.

A simplified recycling chain for WEEE is shown in Figure 1. The recycling chain consists of three steps. The first step is the Collection of WEEE which is out of focus for the improvement of the recycling chain in this thesis. The pre-processing step consists of manual sorting and dismantling as well of shredding and automated sorting. The improvement of the pre-processing stage is the main focus of this thesis. An improved pre-processing also enables improved or new recovery and disposal steps.

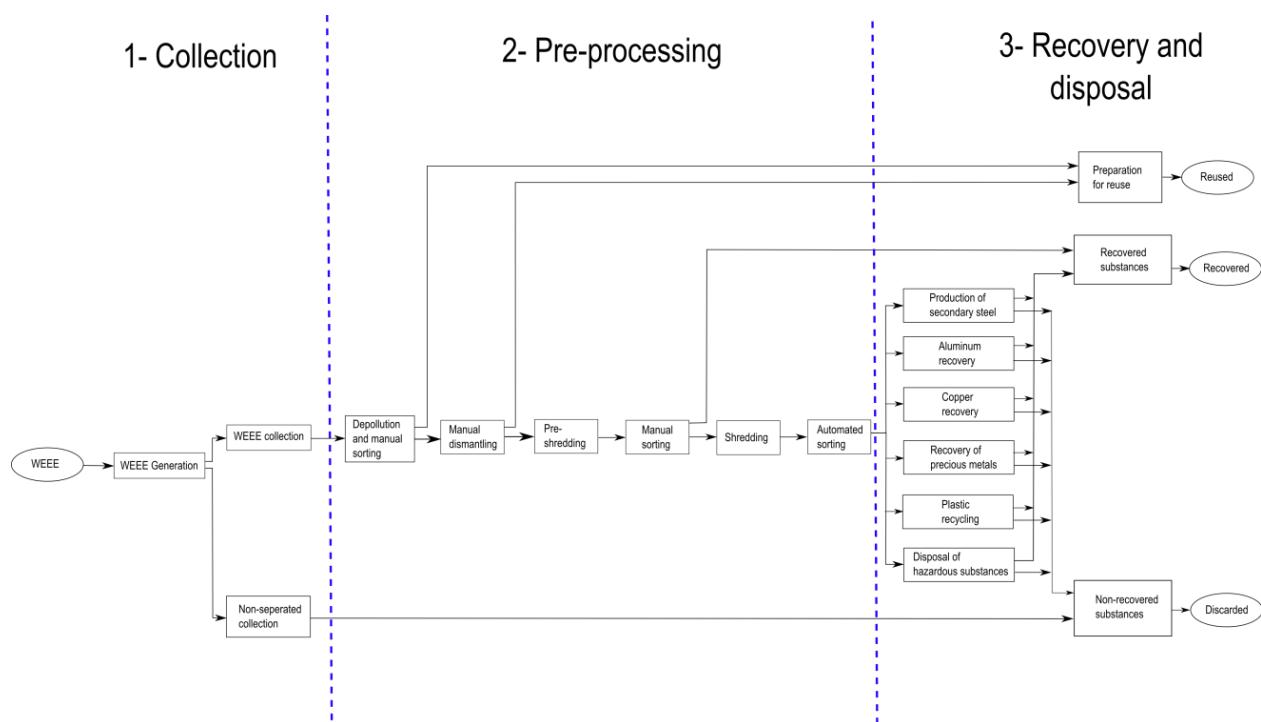


Figure 1: Simplified recycling chain for WEEE

Introduction

The mass balance of the preprocessing step is shown in Figure 2.

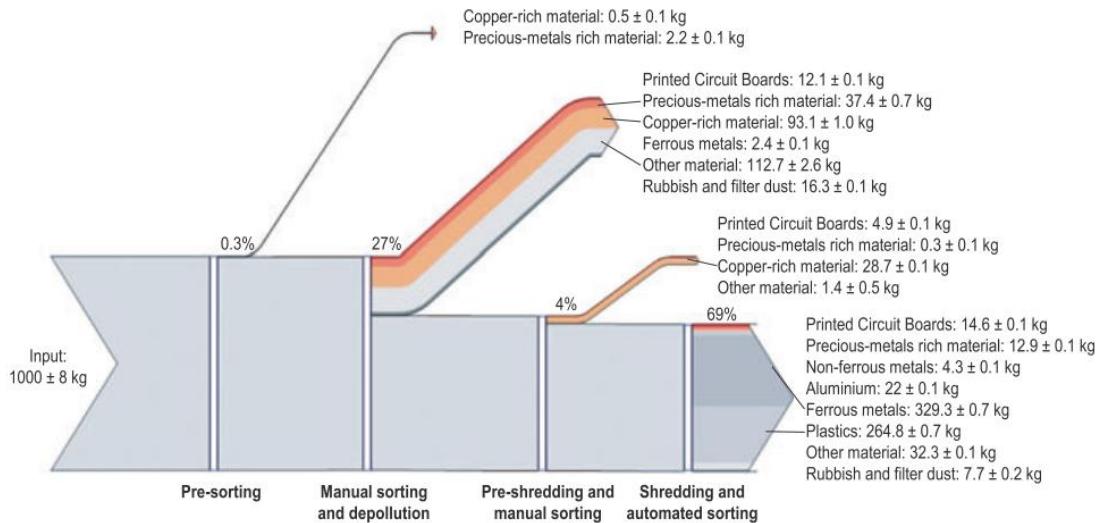


Figure 2: Mass balance of the preprocessing of 1,000 kg of input WEEE (Chancerel, et al., 2009)

A comparison of the input concentration and the output concentration of precious metals shows that only about a quarter of the gold and palladium and about one tenth of silver are sent to the output fraction from which precious metals will be directly recovered. Most of the precious metals go to the most mass relevant fractions. Per ton of input WEEE the company operating the facility does not get any revenues for around 16.5 g gold and 5.3g palladium. At a price of \$900 per ounce of gold and \$370 per ounce of palladium (average price for 2008 [UGS 2009]), this means that a metal value of \$524 for gold and almost \$70 for palladium per ton of treated WEEE is lost. More shredding results in a decrease of concentration of precious metals in PCBs. To reduce the losses of precious metals in preprocessing, in particular during shredding and subsequent sorting, the first and most straight-forward approach is to reduce the quantity of precious metals entering in the shredder. This implied adjusting the manual sorting step at the beginning of the process to remove most precious metal-rich materials. This requires knowledge about the location of precious metals in WEEE, which is currently partially missing (Chancerel, et al., 2009). Characterization of the waste stream is of paramount importance for developing a cost-effective and environmentally friendly recycling system (Cui, et al., 2003).

Introduction

1.2 Purpose

The purpose of this work is to improve the preprocessing step of the recycling chain by an improved automatic characterization of the PCB waste stream which is done on component level. The unshredded or pre-shredded PCBs are inspected by an automatic optical inspection system (AOI-System) based on an electronic component recognition database which contains information about component recognition features and component composition.

Information about the content of valuable materials (gold, silver, palladium, ...) or hazards materials (heavy metals, brominated flame, ...) are used to automatically estimate PCB composition models which contain the location and quantity of specific materials depending on the electronic components of the PCB. This model can help for automatic or manual selective disassembly of precious metal rich components or hazard material rich components.

Information about the economic value of reusable electronic components helps to locate reusable components from an economic point of view. The increase of the reuse rate decreases the negative environmental impacts caused by the production of new electronic components and increases the revenue of recycling companies. An improved recycling chain model with the approach examined in this work is specified in chapter 7.4.

2. Background Theories and related works

Numerous papers where published and research projects are performed in the field of electronic component recognition for PCB recycling.

The goal of the INPIKO project (“Integrierte Prozesskette für die Instandhaltung elektronischer Komponenten”) is to create PCB circuit diagrams from PCBs for the inspection of obsolete electronic components which can be used for repairing or reengineering. The process chain contains the acquisition of 2D-images, 3D-models and CT-data which are combined and analyzed to form an electronic net list (IPK, 2013).

Erik van Dop studied a sensor fusion approach with a range image acquisition module, color image module, and a high-resolution image module. It shows that the fusion of multiple sensor data can increase the recognition rate of electronic components compared to individual sensors.

The AutDem project (Automated disassembly of PWBs) was conducted for automatic disassembly of electronic component for reuse. The project was focusing on the automatic inspection of electronic components for reuse without estimating the material composition of electronic components (Griese, et al., 2002).

The Institute of Imaging and Computer Vision of the RWTH Aachen University examine the generation of height maps with laser triangulation (Koch, et al., 2013) and segmentation of SMD components for automated PCB recycling (Li, et al., 2013). Other approaches deal with the localization of PCBs based on color distribution of solder joints (Article, 2011).

The optical character recognition of electronic components where studied by (Luo, 2014) and an application for mobile package recognition based on the OCR engine Tesseract was developed by (Blaes, et al.).

The approach in this thesis is based on a data fusion model which estimates the component class based on specific component features from 2D-images. The features are extracted from different feature domains to find specific features for each electronic component package.

2.1 Feature extraction algorithms

The traditional goal of feature extraction is to characterize an object to be recognized by measurements whose values are similar for objects in the same class and very different for objects in different classes. This leads to the idea of seeking distinguishing features that are invariant to irrelevant transformations of the input data. In the case of image processing the invariance of features against translation, rotation and scaling is from particular importance. Feature invariance requirements can be skipped if the input data are adjusted (Duda, et al., 2012). Important techniques and algorithms used for feature extraction are summarized in the following chapters.

2.1.1 Single seed region growing approach for color images

For background segmentation and feature extraction from electronic component color segments, a region growing approach is used for region segmentation. The region growing approach is a pixel based image segmentation method since it involves the selection of initial seed pixel. The region growing algorithm examines neighboring pixel of a region or the initial seed pixel and determines if the neighboring pixel should be added to the region (Wikipedia, 2014). The first step is the selection of seed point (x, y) . The seed point selection is depending on the segmentation goal and based on user criterion. The seed point selection is specified for the specific methods in 3.3.4 for Segment based feature extraction and 3.2.2 Color based PCB surface detection. The seed pixel is the first region, from which neighboring pixel are added to grow the region iterative depending on a region membership criterion. In this approach the region growing segmentation is used to segment color images. The criterion to add adjacent pixel $f(x, y)$ to the region pixel PG is the Euclidian distance $DIST$ between the color of the adjacent pixel and the mean color value of the region PG_{mean} . Before segmentation, the image was converted from RGB color space to HSV color space and the gray scaled values in the three channels where linear scaled between 0 and 1.

$$DIST = \sqrt{D_H + D_S + D_V} \quad (1)$$

$$D_H = (f(x, y, 1) - PG_{mean}(1))^2 \quad (2)$$

$$D_S = (f(x, y, 2) - PG_{mean}(2))^2 \quad (3)$$

Background Theories and related works

$$D_V = (f(x, y, 3) - PG_{mean}(3))^2 \quad (4)$$

$$PG_{mean}(1) = \frac{1}{\#PG} \sum_i f(PG(i), 1) \quad (5)$$

$$PG_{mean}(2) = \frac{1}{\#PG} \sum_i f(PG(i), 2) \quad (6)$$

$$PG_{mean}(3) = \frac{1}{\#PG} \sum_i f(PG(i), 3) \quad (7)$$

If the distance smaller than a determined threshold $THR = 0.02$, the pixel is added to the region. If the distance exceeds the threshold, the pixel is not added to the region. If the distance from all neighboring pixel to the region exceed the threshold, the region growing stops and the segmented region is determined as a segment of the image (Verma, et al., 2011). The pseudo code of the single seed region growing approach is shown in [Code 1](#).

Background Theories and related works

SEED: position of seed (x,y)
 RCOUNT: Counter of keep track of current region being grown
 PG – stack to store pixel to grow
 BP – stack to store boundary pixels of grown region
 REGION: matrix with same size if image I, storing the labels of growing region
 CP(j): 4-neighbours of CP, j=1,2,3,4

PSEUDOCODE:

```

Region_Growing(HSV image I)
  THR=0.02
  SEED=(x,y)
  RCOUNT=1
  i=1
  j=1
  PG(i)=SEED
  While PG not empty
    CP=PG(i)
    i=i-1
    For(4-nb of CP, k=1:4)
      If(REGION (CP(k) not labeled)
          Calculate: DIST(SEED,CP(k))
          If(DIST<THR)
            REGION(CP(k))=1;
            i=i+1
            PG(i)=CP(k)
          Else
            j=j+1
            BP(j)=CP(k)
          End if
        End if
      End for
    End for
  End

```

Code 1: Single seed region growing pseudo code

2.1.2 k-means clustering

In the color based PCB surface recognition algorithm in chapter 3.2.2, the k-means clustering algorithm is used to find clusters of PCB surface segments. The algorithm is an unsupervised procedure with the goal to find k mean vectors $\mu_1, \mu_2, \dots, \mu_k$ which represents the center of the k clusters. The k-means clustering is an iterative method where k is the number of clusters. The determination of the number of clusters is described in detail in the belonging chapter. In this approach the initial means $\mu_1, \mu_2, \dots, \mu_k$ where selected randomly from the sample space. The squared Euclidian distance $\|x_k - \hat{\mu}_i\|^2$ is computed for each sample and the nearest mean $\hat{\mu}_m$ is selected to approximate $\hat{P}(w_i|x_k, \hat{\theta})$ as:

Background Theories and related works

$$\widehat{\mathbf{P}}(w_i|x_k, \widehat{\theta}) \simeq \begin{cases} 1 & \text{if } i = m \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

After approximating $\widehat{\mathbf{P}}(w_i|x_k, \widehat{\theta})$ the means $\widehat{\mu}_1, \widehat{\mu}_2, \dots, \widehat{\mu}_k$ are recomputed by:

$$\widehat{\mu}_i = \frac{\sum_{k=1}^n \widehat{\mathbf{P}}(w_i|x_k, \widehat{\theta}) x_k}{\sum_{k=1}^n \widehat{\mathbf{P}}(w_i|x_k, \widehat{\theta})} \quad (9)$$

The approximations of $\widehat{\mathbf{P}}(w_i|x_k, \widehat{\theta})$ and the recomputations of the means are repeated until the approximations do not change compared to the previous iteration step (Duda, et al., 2012).

The pseudo code is shown in Code 2.

PSEUDOCODE:

```
k_Means_Clustering(samples)
begin initialize n, k, μ₁, μ₂, …, μₖ
    do classify n samples according to nearest μᵢ
        recompute μᵢ
    until no change in μᵢ
    return μ₁, μ₂, …, μₖ
end
```

[Code 2_ k-means clustering pseudo code \(Duda, et al., 2012\)](#)

2.1.3 Normalized cross correlation for 2D pattern matching

Template matching is a technique for finding regions in an image that matches a smaller image template. One approach of determining the position of a pattern in an image is based on the 2D normalized cross correlation. Let $f(x, y)$ be the intensity value of an image at the point (x, y) where $x \in \{0, \dots, M_x - 1\}$, $y \in \{0, \dots, M_y - 1\}$ and $M_x \times M_y$ is the image size of the image. The pattern is represented by a given template t of size $N_x \times N_y$. At each position (u, v) in the image f , the normalized cross correlation value γ is calculated between the image f and the template t whereas the template t is shifted by u steps in the x direction and v steps in the y direction. The normalized cross correlation value γ is calculated as follow:

Background Theories and related works

$$\gamma(u, v) = \frac{\sum_{x,y} (f(x, y) - \bar{f}_{u,v})(t(x - u, y - v) - \bar{t})}{\sqrt{\sum_{x,y} (f(x, y) - \bar{f}_{u,v})^2 \sum_{x,y} (t(x - u, y - v) - \bar{t})}} \quad (10)$$

The value $\bar{f}_{u,v}$ is the mean value of $f(x, y)$ within the area of the template t shifted to (u, v) which is calculated by

$$\bar{f}_{u,v} = \frac{1}{N_x N_y} \sum_{x=u}^{u+N_x} \sum_{y=v}^{v+N_y} f(x, y) \quad (11)$$

The value \bar{t} is the mean value of the template t.

$$\bar{t} = \frac{1}{N_x N_y} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} t(x, y) \quad (12)$$

The dominator is the variance of the zero mean image function $f(x, y) - \bar{f}_{u,v}$ and the shifted zero mean template $t(x - u, y - v) - \bar{t}$. With this normalization the value $\gamma(u, v)$ is independent to changes in brightness or contrast of the image. The cross correlation matrix $\gamma(u, v)$ gives a value about similarity between the template and the Image region (Lewis, 1995).

The computation of the normalized cross correlation between a color image and a color template is done by estimating the normalized cross correlation between the image and the template in all three color spaces and estimating the mean value of all three cross correlation matrices $\gamma(u, v, c)$, $c \in \{1, 2, 3\}$.

$$\gamma(u, v) = \frac{1}{3} (\gamma(u, v, 1) + \gamma(u, v, 2) + \gamma(u, v, 3)) \quad (13)$$

The larger the value $\gamma(u, v)$ is, the more likely the template matches the image region.

2.1.4 Image reconstruction with PCA

A set of m component images I_i each of size $r \times c$ is reshaped to a vectors \mathbf{v}_i of size $r * c \times 1$. First the mean vector μ and the covariance matrix \mathbf{C} are computed for all vectors according to (14) and (15).

Background Theories and related works

$$\boldsymbol{\mu} = \frac{1}{m} \sum_{i=1}^m \boldsymbol{v}_i \quad (14)$$

$$\mathbf{C} = \sum_{i=1}^m (\mathbf{v}_i - \boldsymbol{\mu})(\mathbf{v}_i - \boldsymbol{\mu})^T \quad (15)$$

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 \mathbf{e}_i with the k largest eigenvalues λ_i of the covariance matrix are used to construct the projection matrix \mathbf{P} of size $r * c \times k$. The projection of an image vector \boldsymbol{v}_i into the eigenspace is given by

$$\mathbf{p} = \mathbf{P}(\boldsymbol{v}_i - \boldsymbol{\mu}) \quad (16)$$

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 (17) whereas \boldsymbol{v}_i' is the reconstructed image of the image \boldsymbol{v}_i .

$$\boldsymbol{v}_i' = \mathbf{P}^T \mathbf{p} + \boldsymbol{\mu} = \mathbf{P}^T \mathbf{P}(\boldsymbol{v}_i - \boldsymbol{\mu}) + \boldsymbol{\mu} \quad (17)$$

The reconstruction error is defined by the Euclidean distance between the image \boldsymbol{v}_i and its reconstructed image \boldsymbol{v}_i' .

$$d = \|\boldsymbol{v}_i - \boldsymbol{v}_i'\| = \sqrt{\sum (\boldsymbol{v}_i - \boldsymbol{v}_i')^2} \quad (18)$$

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, et al., 2012).

2.2 Feature selection

Variable and feature selection have become the focus of much research in areas of applications for datasets with hundreds or thousands of features variables are available. The goal is to select a subset of features from a feature set which can be useful to improve the prediction performance. Many techniques where published to address the problem of elimination of irrelevant and redundant features in a feature set. Other methods deal with linear combinations of features to form a set of new more useful features. There are three reasons why feature selection is used in applications of classification:

Background Theories and related works

- Improving the prediction performance
- Providing faster and more cost-effective predictors
- Providing a better understanding of processing the data

There are several feature selection algorithms which can be classified in the three categories called wrapper methods, filter methods, and embedded methods.

Wrapper methods are based on a learning machine which is treated as a black box model to score subsets of variables according to their predictive power. In most wrapper algorithms the prediction performance of a given learning machine is used to evaluate subset of features. Important wrapper strategies are the Greedy search strategies of forward selection and backward elimination. The forward selection starts with an empty features set and adds useful features in each step. The backward elimination starts with a set of all variables and progressively eliminates the most useless features.

Filters select subsets of variables as a pre-processing step, independently of the chosen predictor. A distinguished filter method is the Fisher score which is a variable selection method that rates all features and selects the subset of features with the highest score. The Fisher score features selection method is specified in chapter 2.2.1.

Embedded methods perform variable selection in the process of training and are usually specific to given learning machines. The random forest feature selection is an embedded method which uses the out-of-bag (OOB) error to evaluate subsets of features. The random forest feature selection algorithm based on the OOB error is specified in chapter 2.2.2. When the number of variables is very small compared to the number of features one may need to resort the selecting variables with filter methods to avoid over fitting (Guyon, 2003).

2.2.1 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 ratio of the between class variance to the within-class variance (Guyon, 2003). Suppose we have a set of n d-dimensional samples x_1, \dots, x_n , n_k is the number

Background Theories and related works

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 (19).

$$F(x^j) = \frac{\sum_{k=1}^c n_k (\mu_k^j - \mu^j)^2}{(\sigma^j)^2} \quad (19)$$

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 x_i .

$$(\sigma^j)^2 = \sum_{k=1}^c n_k (\sigma_k^j)^2 \quad (20)$$

$$\sigma_k^j = \sum_{x_i \in D_k} x_i^j - \tilde{\mu}_k^j \quad (21)$$

$$\tilde{\mu}_k^j = \frac{1}{n_i} \sum_{x_i \in D_k} x_i^j \quad (22)$$

$$\mu^j = \frac{1}{n} \sum_{k=1}^c n_k \tilde{\mu}_k^j \quad (23)$$

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 (Gu, et al.). In this approach the fisher score is only used in the two stage feature selection and not applied as individual feature selection method.

2.2.2 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 is 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 of all out-of-bag-samples are summed and divided by the number of out-of-bag-samples (Breiman, 2014).

Background Theories and related works

In the Random forest feature selection approach the OOB-error is estimated. The values of the m -th feature of the OOB-samples are randomly permuted and the new OOB-error is estimated. The number of OOB-errors which were made by the variable- m -permuted is subtracted 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 (Cutler, 2014).

2.3 Object Classification

Object recognition in image processing is the act of finding and identifying objects in an image or video sequence. Object classification is a special case of object recognition where the task is to detect objects and classify the objects in object categories. The task is still challenging for computer vision systems and many approaches have been implemented over multiple decades. The object recognition can be classified in three categories: Approach based on CAD-like object models, appearance-based methods and feature-based methods. Feature based methods are often combined with classifiers which classify the objects based on the features according to their object category. There exist many classification algorithms which can be divided in supervised and unsupervised classification methods.

A classifier which is used in many applications is the Support vector machine (SVM) which is based on the idea to classify data based on the largest margin between data cluster. Another popular ensemble classifier which is based on decision trees classifiers is the random forest. Both classifiers are specified in the chapter 2.3.1 and 2.3.2 (Wikipedia, 2015).

2.3.1 Random forest classifier

Random forests are ensemble classifiers which are constructed of a multitude of decision trees. The algorithm was introduced by Leo Breiman and Adele Cutler and is used for classification and regression in many applications.

Introduction to ensemble classifier

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

Background Theories and related works

classification models (classifiers) which are learning the classified 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, 2009).

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.

Random forest training

Given a training set $\mathbf{X} = \mathbf{x}_1, \dots, \mathbf{x}_n$ with response $\mathbf{Y} = \mathbf{y}_1, \dots, \mathbf{y}_n$, bagging repeatedly selects bootstrap samples of the training set and fits trees to the samples. For each tree in the random forest classifier, training subsets $\mathbf{X}_b, \mathbf{Y}_b$ (bootstrap samples) from the training set are randomly selected and train the bagging trees f_b on \mathbf{X}_b and \mathbf{Y}_b . 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 (Cutler, 2014). In random forests at each candidate split a random subset of features is selected. Typically for a dataset with p features \sqrt{p} features are used in each split (Breiman, 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 (24) whereas B is the number of trees in the forest and \hat{f}_b is the trained tree (Breiman, 2014).

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B \hat{f}_b(\mathbf{x}) \quad (24)$$

Out-of-bag (OOB) estimation

To train a k -th tree, a random subset of training samples X_b, Y_b is used to construct the tree, whereas each tree uses different bootstrap samples. The samples that are not used to construct the k -th tree are predicted by the k -th tree to get a classification. The estimation is called 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, 2014).

2.3.2 Support vector machine classifier

Support vector machine (SVM) is a learning algorithm that analyzes data and recognizes patterns used for classification and regression analyses. Given a set of training samples, each marked with one of two classification categories, an SVM model can be trained to assign new samples into one category or the other. In addition to performing linear classification, an SVM can efficiently perform a non-linear classification by using the so called kernel-trick. The kernel-trick is a mapping of the input data to a high-dimensional feature space (Wikipedia-SVM, 2015). The SVM classifier constructs a hyperplane or set of hyperplanes in a high- or infinite dimensional space. A good separation is achieved if the hyperplane has a large distance to the nearest training data points of any class (functional margin), since in general the larger the margin the lower the generalization error of the classifier.

Linear Support vector machine

The linear support vector machine (Linear-SVM) is the simplest case of SVMs and can be used to classify linear separable data by constructing a separating hyperplane. Suppose there are labeled training data

$$\{\mathbf{x}_i, y_i\}, i = 1, \dots, l, y_i \in \{-1, 1\}, \mathbf{x}_i \in \mathbf{R}^d \quad (25)$$

and a hyperplane which separates the positive and negative data. The points \mathbf{x} which lies on the hyperplane satisfy $\mathbf{w} \cdot \mathbf{x} + b = 0$, where \mathbf{w} is the normal of the hyperplane and $|\mathbf{b}|/||\mathbf{w}||$ is the perpendicular distance from the hyperplane to the origin, and $||\mathbf{w}||$ is the Euclidian norm

Background Theories and related works

of \mathbf{w} . For the linear separable case, the goal of the algorithm is to find the separating hyperplane with the largest margin. This can be formulated as follows:

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +1 \text{ for } y_i = +1 \quad (26)$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \leq -1 \text{ for } y_i = -1 \quad (27)$$

These can be combined into one set of inequalities:

$$y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \quad \forall i \quad (28)$$

The points for which the equality (27) holds are placed on the hyperplane $H_1: \mathbf{x}_i \cdot \mathbf{w} + b = 1$ and the point for which the equality (28) holds are placed on the hyperplane $H_2: \mathbf{x}_i \cdot \mathbf{w} + b = -1$, they are called support vectors. The distance of the hyperplane H_1 and H_2 from the separation hyperplane is $d_+ = d_- = 1/\|\mathbf{w}\|$ and the margin is $2/\|\mathbf{w}\|$. To maximize the margin, $\|\mathbf{w}\|$ has to be minimized subject to the constraints (28). This problem can be reformulated by introducing Lagrange multipliers α to the Lagrangian:

$$L_p = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^l \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^l \alpha_i \quad (29)$$

The Lagrangian L_p has to be minimized with respect to \mathbf{w} and simultaneously require that the derivatives of L_p with respect to all the α_i vanish, subject to the constraints $\alpha_i \geq 0$. Now it is a quadratic programming problem which can be solved by standard quadratic programming techniques and programs. The solution can be read in (Burges, 1998). The vector \mathbf{w} can be expressed as a linear combination of the training vectors:

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \quad (30)$$

The problem can be reformulate in the “dual” problem which maximizes L_p subject to the constrain that the gradient of L_p with respect to \mathbf{w} and b vanish, and the subject also to the constrain that the $\alpha_i \geq 0$. Requiring that the gradient of L_p with respect to \mathbf{w} and \mathbf{b} vanish give the condition:

Background Theories and related works

$$\sum_i \alpha_i y_i = 0, 0 \leq \alpha_i \leq C \quad (31)$$

This can be substituted in (29) to give

$$L_D = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (32)$$

(Burges, 1998)

RBF Support vector machine

The linear-SVM algorithm can be extended by using non-linear functions as hyperplane. This is done with the so called kernel-trick. The dot product $\mathbf{x}_i \cdot \mathbf{x}_j$ is replaced by a nonlinear kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$. The hyperplane can now separate the positive and negative samples in a higher feature space. A common used nonlinear kernel is the Gaussian radial basis function (RBF) kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right), \text{ for } \gamma > 0 \quad (33)$$

$$\gamma = \frac{1}{2\sigma^2} \quad (34)$$

An RBF-kernel is used because of the complexity of the RBK kernel which is lower than for example polynomial kernels (Hsu, et al., 2010).

Grid search method for parameter selection

One of the most important steps of support vector machines (SVM) modeling is the parameter selection. In this approach the grid search method is used to estimate the optimal parameter which maximizes the classification accuracy. For the linear Support vector machine only the regularization constant C has to be determined. The regularization constant is adjusting the confidence interval range of the learning machine. By selecting a RBF kernel function, the regularization constant C and the kernel hyperparameter γ have to be determined. For the linear SVM the grid search method is taking m values in C to form a one dimensional grid. The values are used to estimate the performance of trained SVMs in a three-fold-cross-validation model. The optimal parameter is chosen depending on the maximum performance.

Background Theories and related works

The grid search method for the nonlinear RBF kernel SVM is taking m values in C and n values in γ to form a $m \times n$ grid. The values are used to estimate the performance of trained SVMs in a three-fold-cross-validation model. The optimal parameter combination is chosen depending on the maximum performance (Qubo, et al.).

2.4 Data fusion model

The integration of data and knowledge from several sources is known as data fusion. It is a combination of multiple data sources to obtain information with higher quality or more relevant information. In this approach a data fusion model is used for object recognition. The data fusion techniques can be classified in three nonexclusive categories: (i) data association, (ii) state estimation, (iii) decision fusion (Castanedo, 2013). Some common classification schemes are bases on the relation between the data sources. The Dasarathy's Classification is a data fusion classification schema which classifies the data fusion in five categories: data in-data out (DAI-DAO), data in-feature out (DAI-FEO), feature-in feature out (FEI-FEO), feature in-decision out (FEI-DEO) and Decision In-Decision Out (DEI-DEO). The JDL data fusion classification is concept propose be the JDL and the American Department of Defense (DoD). It classifies the data fusion on five processing levels: Sub-Object Data Assessment, Object Assessment, Situation Assessment, Impact Assessment, Process Refinement (Steinberg, et al.).

The Data fusion classification model which is manly used in image processing and used in this approach is based on the following abstraction levels:

- signal level: directly addresses the signals that are acquired from the sensors
- pixel level: operates at the image level and could be used to improve image processing tasks
- characteristic: employs features that are extracted from the images or signals
- symbols: at this level, information is represented as symbols, this level is also known as the decision level

The data fusion on characteristics level (feature level) and the data fusion on symbol level (decision level) are used in this approach to improve the recognition process of electronic

Background Theories and related works

components (Castanedo, 2013). A multi-sensor object recognition system for electronic components was already investigated by Erik Roeland van Dop in “Multi-sensor object recognition: The case of electronics recycling” (van Dop, et al., 2001). In this work the image data from a range image module, a color image module and a high-resolution image module are combined to improve the information for object classification. In the experiments he used 448 modeled objects (electronic components) and reached a correctly classified rate of the combined sensor module from 82% (369/448) (van Dop, et al., 2001).

Data fusion with Dempster-Shafer theory

Decision-level fusion consists of merging information at higher level of abstraction. The fusion step combines multiple algorithms to yield a final fused decision.

The Dempster-Shafer (DS) theory of evidence, also known as theory of belief functions, is a tool for representing and combining evidence. The DS-theory is a generalization of the Bayesian reasoning but does not require probabilities for each question of interest. The Dempster-Shafer theory starts by assuming a universe of discourse consisting of a finite set of mutual exclusive atomic hypotheses $h = \{h_1, \dots, h_n\}$. Let 2^h denote the power set of all subsets of h . The function $m: 2^h \rightarrow [0,1]$ is called a basic probability assignment (masses) if it satisfies:

$$m(\emptyset) = 0 \quad \text{and} \quad \sum_{A \subseteq h} m(A) = 1 \tag{35}$$

The belief can not only be assigned to an atomic hypothesis, but some set $A = \{a_1, \dots, a_n\} \subset h$. The belief in $m(A)$ represents our ignorance, which can be subdivided among the subsets of A . Each element B with $m(B) \neq 0$ is called a focal element. The belief function is defined as:

$$bel(B) = \sum_{A \subseteq B} m(A) \tag{36}$$

It represents the minimal trust we can have in B because of the supporting subset A . The complement of belief is doubt.

$$doubt(B) = 1 - bel(B) \tag{37}$$

Background Theories and related works

The plausibility $pl(A)$ is the sum of all masses of the subset of the set of interest.

$$pl(B) = \sum_{A \cap B \neq \emptyset} m(A) \quad (38)$$

The plausibility $pl(A)$ can be derived from the belief in the following way:

$$pl(B) = 1 - bel(\bar{B}) \quad (39)$$

The complement of plausibility is disbelief.

$$disbelief(B) = 1 - pl(B) \quad (40)$$

The connection between belief, disbelief, plausibility, and doubt is shown in Figure 3 (Kay, 2007).

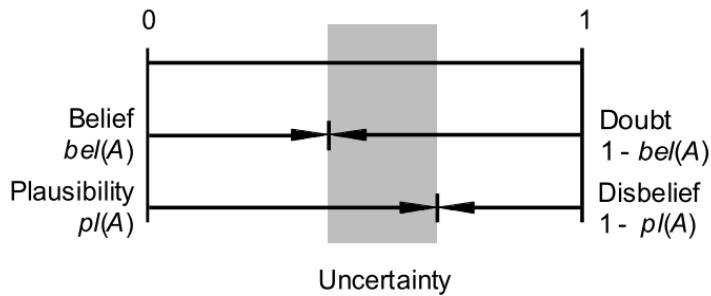


Figure 3: Connection between belief, disbelief, plausibility and doubt (Rakowsky, 2007)

Dempster combination rule

The Dempster combination rule is the possibility to combine masses m_1, \dots, m_s on h with the orthogonal sum $m_{1,\dots,s} = m_1 \oplus \dots \oplus m_s$ which is defined as:

$$m_{1,\dots,s}(C) = K \sum_{A_1 \cap \dots \cap A_s = C} m_1(A_1) \cdot \dots \cdot m_s(A_s) \quad (41)$$

where

$$K^{-1} = \sum_{A_1 \cap \dots \cap A_s = \emptyset} m_1(A_1) \cdot \dots \cdot m_s(A_s) \quad (42)$$

The factor K is measuring the conflict between m_1, \dots, m_s .

Background Theories and related works

After performing the combination, the decision associated to the most probable element in h has to be quantified. The most common decision rule is the maximum of belief, where the element in h is quantified which corresponds to the element with the maximum belief. In applications for safety and reliability modeling different decision rules are used.

2.5 Optical character recognition of IC markings from electronic PCB scrap

Optical character recognition (OCR) is the conversion from images of typewritten or printed text into machine-encoded text. OCR is widely used in many applications for document digitalization, analyses of passports, bank statements, license plate character recognition or other documents. One kind of object-oriented OCR is the recognition of electronic component markings. The PCB production increases worldwide and quality control becomes more and more important. Therefore many OCR engines were developed to recognize character strings on ICs or other electronic components. Most of the IC-marking recognition engines are developed for the inspection of chips and electronic components for assembly (Luo, 2014). Many applications use Optical character verification approaches (OCV) due to the fact that the expected IC marking is well known. The quality of the string characters for assembly or quality control of the component production is sufficient for good character recognition results.

Another area of IC marking recognition is the recognition of IC markings from electronic components from electronic PCB scrap. The quality of IC markings of used electronic components is much worse compared to new IC components. Dirt, scratches or faded markings decrease the recognition rate dramatically. Unknown character positions, font or size make it more difficult to recognize characters. Just a few publications deal with the task of IC marking recognition from electronic PCB waste (Li, et al., 2014). An important measurement in OCR is the Levenshtein distance, which is a distance measure between sequences of characters and used to compare recognition results.

2.5.1 Levenshtein distance

The Levenshtein distance is a string metric for measuring the difference between two sequences. The distance is the number of deletions, insertions, or substitutions required to transform a string $string1$ into another string $string2$. The greater the Levenshtein distance, the more

Background Theories and related works

different the strings are (Wikipedia-Levenshtein, 2015). The Levenshtein distance between two strings a and b is given by $lev_{a,b}(|a| < |b|)$ where

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0 \\ \min \left\{ \begin{array}{l} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{array} \right\} & \text{otherwise} \end{cases}$$

where $1_{(a_i \neq b_j)}$ is the indicator function equal to 0 when $a_i = b_j$ and equal to 1 otherwise (Wikipedia-Levenshtein, 2015).

2.5.2 RANSAC algorithm

The RANSAC algorithm is an iterative method to estimate parameters of a mathematical model from a set of observed data which contain outliers. The algorithm categorizes all data samples between “inliers” and “outliers” whereas inliers fit the model with a certain error and outliers do not fit the model.

1. The algorithm selects randomly a number of samples from the whole set to fit a model. The number of selected samples is the minimum number of data items which are necessary to estimate the model parameter.
2. A model is fitted by the selected data samples.
3. The model is evaluated by the data samples which were not used to fit the model. The algorithm checks if the data samples are consistent with the model, therefore an error threshold is determined. If the error between the model and a data sample is greater than the error threshold, the sample is classified as outlier. If the error between the data sample and the model is within the error threshold the data sample is classified as inlier.
4. The goodness of the model is estimated according to the number of outliers and inliers of the model.

This procedure is repeated a fixed number of times and the most refined model parameters with the minimum number of outliers are selected as parameters for the mathematical model

Background Theories and related works

(Wikipedia-RANSAC, 2015). An example of a linear model selected by the RANSAC algorithm compared with the fitted least square error model is shown in Figure 4.

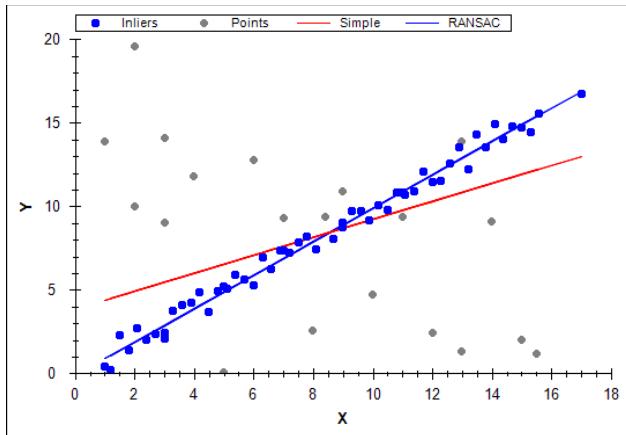


Figure 4: RANSAC example (http://www.codeproject.com/KB/recipes/automatic_panoramas/ransac.png)

2.5.3 Octopart database for component-name verification

Potential component names are requested by the Octopart API (www.Octopart.com) by sending the composed component labels. After making a label request, the Octopart API sends back a list of potential component names located in their database which could correspond to the requested label. The distance between the potential component 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 component name with the smallest distance to requested label is assigned as component name.

Octopart is a company that offers an electronic component database with structured data for more than 30 million electronic components. The Octopart tools facilitate to search component 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

Background Theories and related works

including Digi-Key, Mouser, Newark, Premier farnell, Arrow, RS Component, Future electronics, Grainger and many others (octopart, 2014).

This tool was used for component 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 component names provided from different suppliers. To assign a component name from the obtained list to the recognized label, the Levenshtein distance between the component names and the requested label is computed. The component name with the smallest distance less than or equal the distance threshold $distance_{label, oct, thresh} = 2$ is assigned to the component. The requests were made with the data transfer tool curl in MATLAB.

2.6 Life Cycle Inventory (LCI) analysis

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 ((SAIC), et al., 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” (Bakst, et al., 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

2.6.1 Categorization of WEEE and PCB waste

Waste electrical and electronic equipment (WEEE) describes discarded electrical or electronic devices. The WEEE directive sets targets for collection, recycling and recovery for WEEE and

Background Theories and related works

became a European law in 2003 (Directive 2002/96/EC, 2002) WEEE directive sets a total of 10 categories of WEEE:

- Large household appliances
- Small household appliances
- Consumer equipment
- Lighting equipment
- Electrical and electronic tools
- Toys, leisure and sports equipment
- Medical devices
- Monitoring and control instruments
- Automatic dispensers

In this work the focus is set on the recycling and reuse of electronic components from PCBs. Therefore the WEEE categories which contain a height amount of PCBs like IT and telecommunication equipment, consumer equipment, medical devices, monitoring and control instruments and automatic dispensers are of particular importance (Chancerel, et al., 2009).

PCB waste from WEEE can be categorization in the following categories:

- PCBs class 1 A: old PCBs with golden contacts, height chip density
- PCBs class 1 B: PCBS from computers, industry equipment, many gildings and precious metal reach chips
- PCBs class 1 C: colored motherboards, graphic cards, sound cards
- PCBs class 2 A: PCBs from industry equipment without golden contacts, small precious metal reach chips
- PCBs class 2 B: PCBs from industry equipment without golden contacts, without precious metal reach chips, contain small heat sinks or transformers
- PCBs class 3: PCBs with big capacitors, heat sinks or transformers (PCBs from old monitors or power supply controllers)

Background Theories and related works

Valuable PCB component can be categorized as gold connectors, mobile phone PCBs, CPU ceramic gold caps, CPU ceramic Intel AMD, plastic CPU processors, CPU slot processors, RAM devices, chips (chips, ICs, Eproms), hard drives, CD-/DVD-drives, transformers, cables, relays and precious metal reach components (quartz, transistors, capacitors, resistors,...)

(Scheideanstalt, 2015). The categorization became in more detail in the last years, what is associated with the increasing interest in recycling of electronic waste. Several recycling companies recycle tantalum capacitors due to the fact that tantalum recycling became more profitable in the last years (Tantalumrecycling, 2015).

2.6.2 Recycling and reuse potential of electronic PCB waste

The use of electronic equipment has increased worldwide in the past few years. Precious metals are an important raw material for EEE manufacturers and the demand is growing fast. After use phase the EEE becomes waste (WEEE). The concentration of precious metals in WEEE is small, but the economic and ecological value of precious metals like gold, silver or palladium and special metals like tantalum or neodymium make recycling economically and ecologically relevant. Recycling of raw materials from end-of-life electronics is the most effective solution for solving the growing of e-waste problem. Recycling also prevents for landfill of hazardous materials from PCBs. The highest concentration of precious metals in WEEE is located in the PCBs. One ton of PCB waste contains around 135 g gold, 669 g silver and 50 g palladium which can be recycled. In several recycling chains only about a quarter of the gold and palladium and a tenth of silver are sent to the output fraction from which precious metals will be directly recovered (Chancerel, et al., 2009).

Reuse of electronic components can help to prevent health problems, create jobs and reduce greenhouse-gas emissions. Unfortunately today's market for reused electronic component is very small. Testing of unsoldered electronic components is very difficult caused by the height diversity and complexity of electronic components. Also low prices of electronic components in consumer electronics is challenging for a growing reuse market. A system which determined the price of specific electronic components to estimate the revenue is necessary to increase the potential of reusing electronic components.

2.6.3 International Reference Life cycle Data System (ILCD) format

The International Life Cycle Data System (ILCD) has been developed by the Joint Research Centre - Institute for Environment and Sustainability (JRC-IES) of the European Commission to provide guidance for consistent and quality assured Life Cycle Assessment data and studies (Commission, et al., 2012). The ILCD Data Format was developed for storing and structuring data set information within a data stream or file to enhance the availability of consistent and quality assured Life Cycle Inventory (LCI) data sets. It was designed to serve as reference format and for data exchange between varieties of Life Cycle impact assessment (LCA) software. The ILCD data format has been released in 2009 and has already seen some adoption among tools like GaBi or OpenLCA and databases in the meantime. The ILCD format is based on an Internet-aware, linked data approach. The ILCD format provides currently seven data set types which identify different semantic concepts in LCA modeling that are linked together via typed links called global references (Wolf, et al., 2011). These types of data set concepts are:

- Process: Modeling unit and aggregated processes and result sets. Input and Output flows are modeled by global references to other datasets of type flow.
- Flow: Describes an elementary, product or waste flow. It reference one or more Flow properties.
- Flow Property: Describes physical or other properties of a flow that can be used to quantify it, for example mass. Each instance references one Unit Group data set.
- Unit Group: Describes a group of convertible units and the conversion factors to its reference unit
- LCIA Method: Describes an LCIA method and its characteristic factors
- Source: Represents an external source of information, such as literature or a database or data format. It can reference a contact it is related to.
- Contact: describes a person or organization.

The ILCD format is used in this work to transfer LCI models of PCBs which are automatically created in MATLAB. They can be imported in LCA software like GaBi or OpenLCA to analyze ecological impacts.

3. Methods for electronic component recognition

The methods for object recognition used in this work are based on the data fusion model specified in chapter 3.6. At first the acquired images are preprocessed which consist of rotation correction and scaling determination. After preprocessing, the detection of electronic components is studied which includes the determination of the component bounding boxes in the image. A detailed measurement of detection investigation is not performed. This work is focusing on component classification based on the component detection step. The classification step is based on feature extraction and the following feature selection (feature-fusion-level) of the most important features. The classification of the components is examined with the random forest classifier and support vector machines (Linear-SVM and RBF-SVM) (classifier-fusion-level). The component class which can be one of the components in the database or an unknown component is determined on decision-fusion-level. To assign a component to a component in the Octopart database, an OCR approach is applied to identify the component name. The Octopart database is used to verify the electronic component name and receive additional information about availability and prices of the electronic component.

3.1 Image preprocessing

The image preprocessing is the first step after image acquisition. In this work the preprocessing consists of two steps, the image rotation correction and the determination of the image scale. The object recognition is based on features which are extracted from the images. In many object recognition tasks, based on 2D image data, the object can be rotated or appear in different scales. Features which are invariant in scaling and rotation have to be found for object classification. The advantage of this work is the fact that invariance against scaling and rotation of the object is determined in the preprocessing step. The rotation correction is applied on the whole PCB image which is specified in chapter 3.1.1. The scaling is also applied on the whole image whereas the dimensions of the electronic component are fixed and the scale of the image is determined based on a scaling symbol. The scaling estimation process is specified in chapter 3.1.2.

3.1.1 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 component are horizontal or vertical aligned. The whole process is based on the assumption that conductor tracks and electronic components 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 an angle Θ is the DFT of the unrotated image, rotated by the same angle Θ . The rotation property of a DFT is derived in (Petrou, et al., 1999) and therefore omitted here. The approach is based on the property that lines (edges) in the image are transformed to points in the frequency domain. Horizontal lines in the image are transformed to points on the centered vertical line in the frequency domain and vertical lines in the image are transformed to points on the horizontal centered line in the frequency domain. An example is shown in Figure 5.

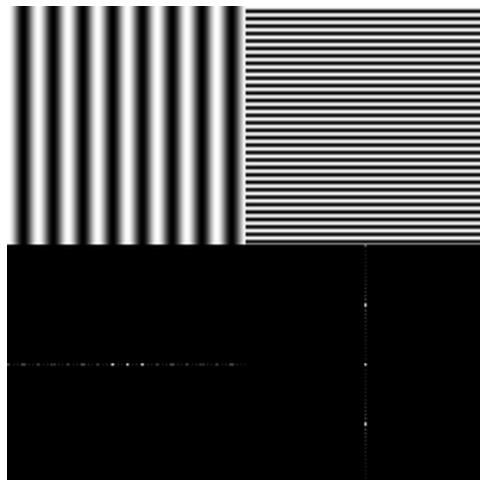


Figure 5: Transformation from lines in the image to points in the frequency domain (www.svi.nl/FourierTransform)

The image rotation correction process is shown in Figure 6.

Methods for electronic component recognition

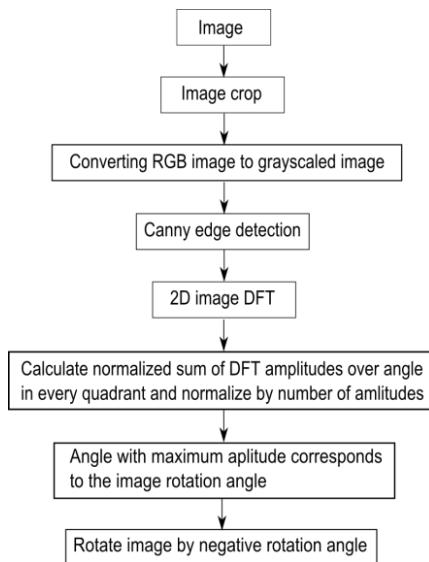


Figure 6: Image rotation correction process

At first the Image is cropped to a squared image (2000 pixel x 2000 pixel) to reduce process runtime. The RGB image is converted to a grayscaled image and canny edge detection is applied. Afterward a 2D-DFT (discrete Fourier transform) 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 3.0 degree, the edge image and amplitude of the DFT-image and summed up amplitude over the angle is shown in Figure 7, Figure 8, Figure 9 and Figure 10. The precision of angle estimation was not investigated in detail but inaccuracies could not be determined by eye.

Methods for electronic component recognition

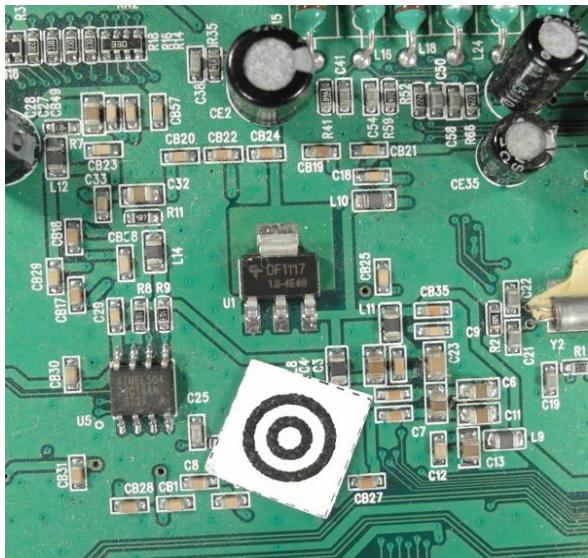


Figure 7: Image rotated by 3.0 degree

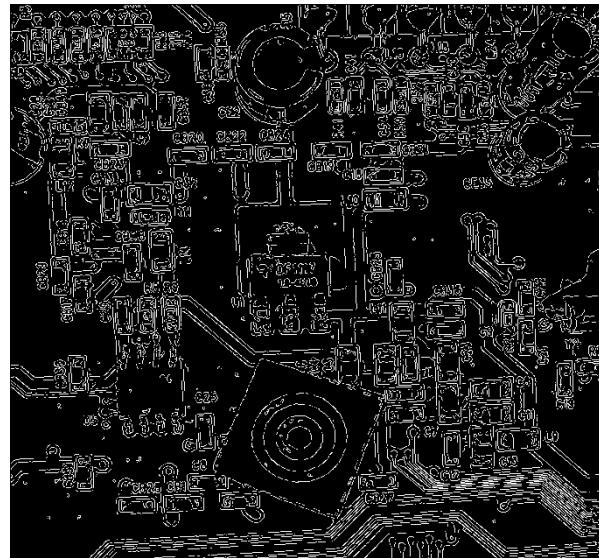


Figure 8: Canny edge image of the rotated image

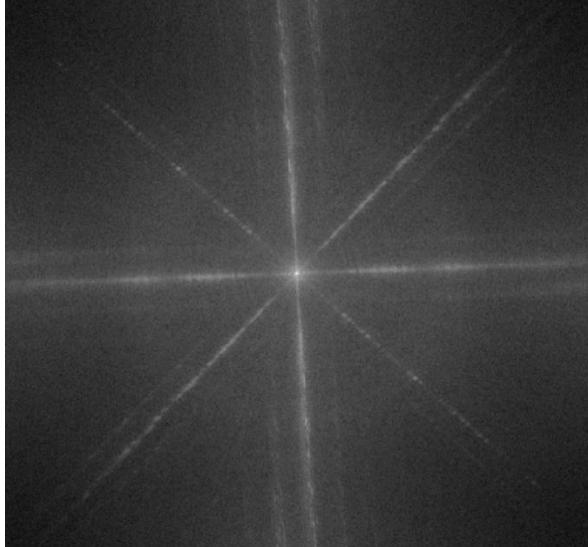


Figure 9: Shifted DFT of the rotated image (logarithmic representation)

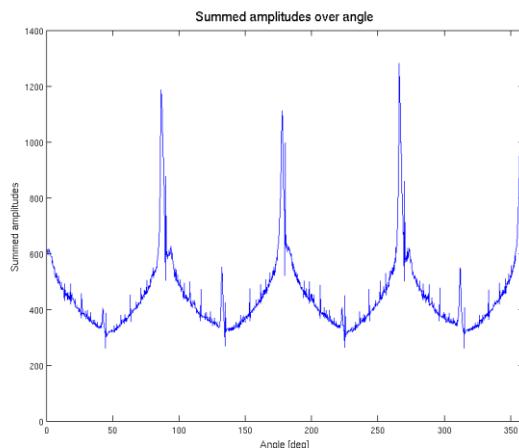


Figure 10: Summed amplitude over angle (invariants by 90 degree)

3.1.2 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.

Methods for electronic component recognition

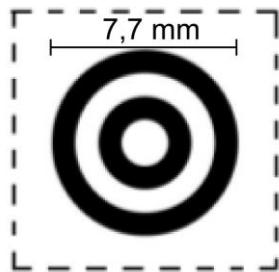


Figure 11: Scale symbol



Figure 12: Scale symbol placed on the board

The scaling symbol is shown in Figure 11 and Figure 12. The whole scaling determination process is shown in Figure 13.

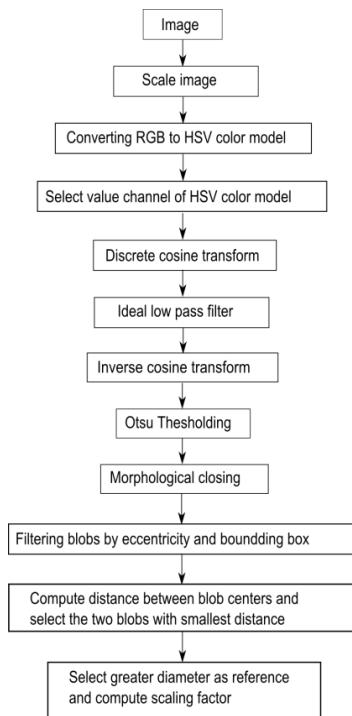


Figure 13: Scaling determination process

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

Methods for electronic component recognition

kernels. The discrete cosine transform is shown in Equation (43) and (44) (Gonzalez, et al., 2006).

$$T(u, v) = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} g(x, y) \alpha(u) \alpha(v) \cos \left[\frac{(2x+1)u\pi}{2n} \right] \cos \left[\frac{(2y+1)v\pi}{2n} \right] \quad (43)$$

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{n}} & \text{for } u = 0 \\ \sqrt{\frac{2}{n}} & \text{for } u = 1, 2, \dots, n-1 \end{cases} \quad (44)$$

$$\alpha(v) = \begin{cases} \sqrt{\frac{1}{n}} & \text{for } v = 0 \\ \sqrt{\frac{2}{n}} & \text{for } v = 1, 2, \dots, n-1 \end{cases} \quad (45)$$

To suppress illumination changes, an ideal low pass filter is applied in the frequency domain in which the first 10×10 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 (5×5) 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.

$$blobs_{scale} = \{blobs, ecc_{min} < ecc \wedge d_{min} < d < d_{max}\} \quad (46)$$

$$ecc_{min} = 0.7, d_{min} = 25 \text{ pixel}, d_{max} = 500 \text{ pixel}$$

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

Methods for electronic component recognition

the scaling symbol. The outer diameter of the larger blob is used as reference to calculate the image scale.

$$imagescale = \frac{\text{diameter [pixel]}}{\text{diameter [mm]}} \quad (47)$$

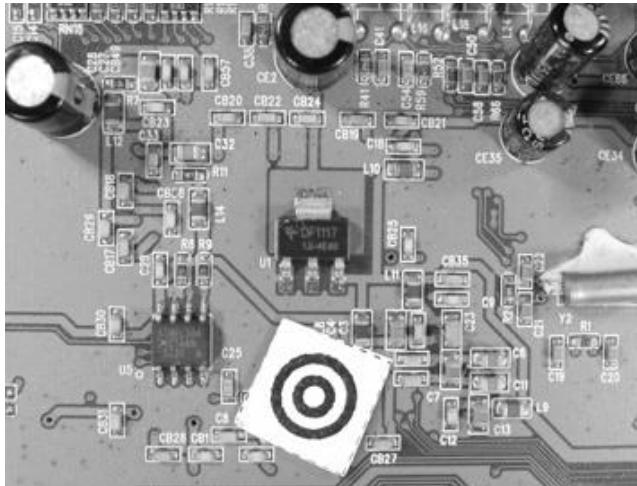


Figure 14: Value channel (brightness) of HSV color image

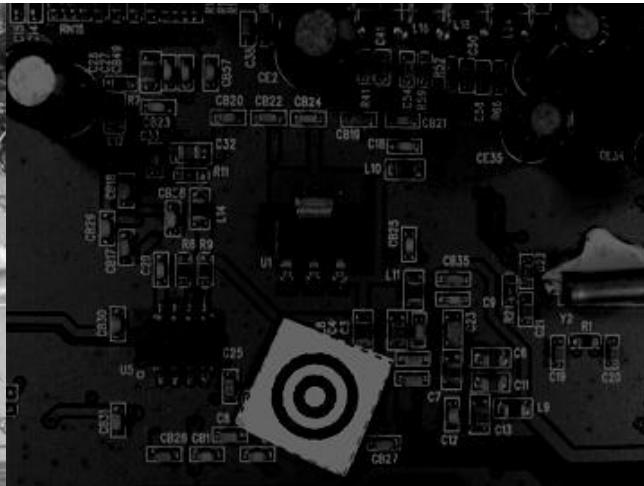


Figure 15: Cosine transform filtered image

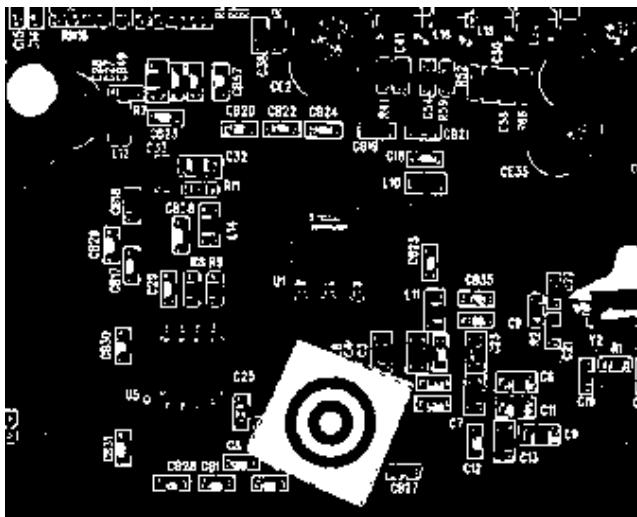


Figure 16: Otsu thresholding



Figure 17: Blobs of the scaling symbol

3.1.3 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 tradeoff 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.

$$area_{component} [mm^2] = width_{component}[mm] * height_{component}[mm] \quad (48)$$

$$PPMM(area_{component}) = a * \exp(-b (area_{component} [mm^2]) - c) [ppmm] \quad (49)$$

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 18.

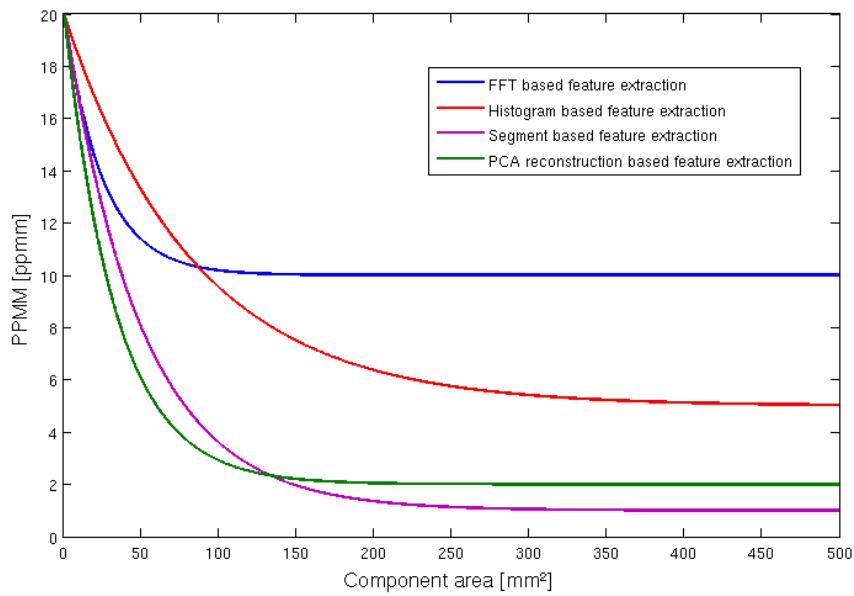


Figure 18: Resolution dependency from component area and feature extraction algorithm

3.2 Electronic component detection

A necessary processing step for component classification is component detection. The detection includes the determination of component positions without knowing to which component class the detected component belongs. The component detection is necessary because the component classification is time consuming and a classification of every possible component position in the image is impossible. The goal of component detection is to narrow the search space. False positive detections (component detections at positions where no component is located) can be corrected by the component classification step. False negative detections (component detections where no component is located) cannot be corrected by the component classification step. Several component detection approaches were studied. Approaches based on the PCB surface color (chapter 3.2.2) and based on 2D normalized cross correlation (chapter 3.2.3) are specified in this work. Component detection approaches based on laser triangulation (chapter 6.1.2) or PCB 3D models (chapter 6.1.1) were already specified in several papers.

3.2.1 PCB board segmentation

One of the steps before detecting electronic components is the segmentation of the PCB board to reduce the search area for electronic components. In this approach the PCBs were placed on a white sheet and images are acquired, what results in a white/bright background. In this approach the process flow shown in Figure 19 is applied.

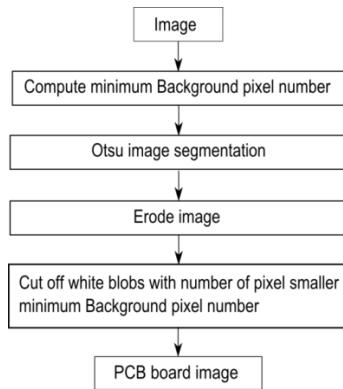


Figure 19: PCB board segmentation process flow

At first the minimum of background pixel is defined by 5% of the image pixel to do not cut off white regions from the PCB board.

$$\#Backgroundpixel_{min} = 0.05 * \#Imagepixel \quad (50)$$

Than Otsu segmentation is applied, followed by a morphological erode step with a 10x10 kernel to separate white regions from the PCB board which are connected with the background. In the last step all blobs with the number of pixels greater than the minimum background pixel number $\#Backgroundpixel_{min}$ are cut off whereby all remaining regions are mainly PCB regions. An example for the PCB board segmentation is shown in Figure 20, Figure 21, Figure 22and Figure 23.

Methods for electronic component recognition

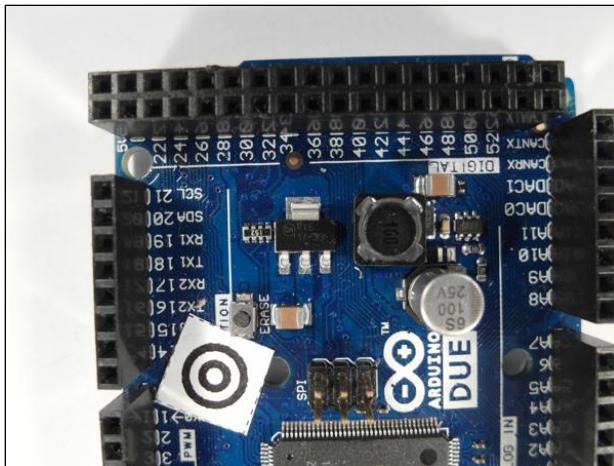


Figure 20: Acquired PCB image

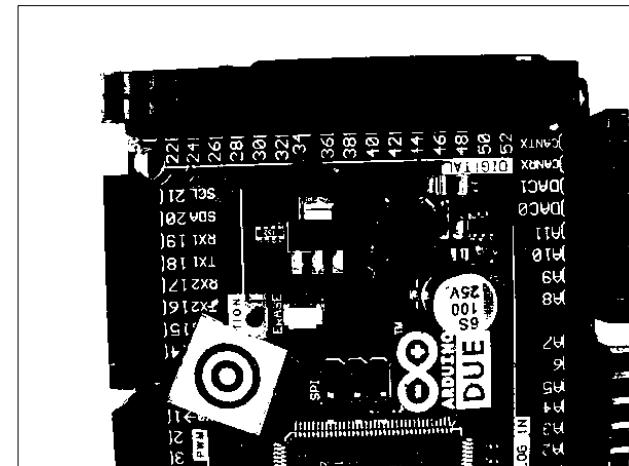


Figure 21: Otsu segmentation

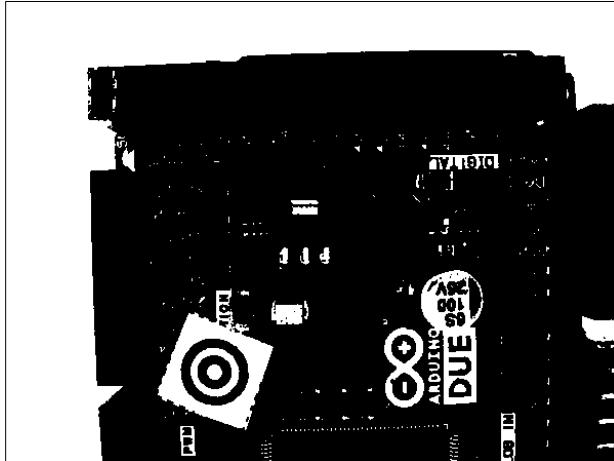


Figure 22: Morphological eroded image with 10x10 kernel

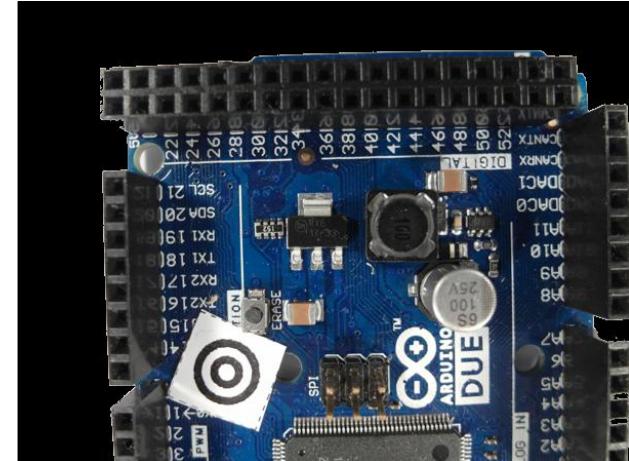


Figure 23: Segmented PCB board image

3.2.2 Color based PCB surface detection

To classify an electronic component it is necessary to know the position of the electronic component on the PCB board. One possibility process step is the segmentation of the PCB surface based on the color and distribution of the surface pixel over the PCB image.

This approach is based on the following assumption of PCB surfaces:

- Most PCB surfaces have striking colors compared to the color of the electronic components or PCB markings. That results in the mostly colored isolating protection lacquer whereas often used colors are green, blue, orange, red, etc.

Methods for electronic component recognition

- The number of surface pixel cluster is high compared to other pixel clusters caused by the mostly large surface area compared to individual components
- The surface pixels form mostly large connected areas of the PCB surface what results in a small number of segment blobs compared to other clusters
- The surface segments form mostly contiguous areas with the result that the number of edge pixels is smaller compared to other segment clusters

The process flow is shown in Figure 24.

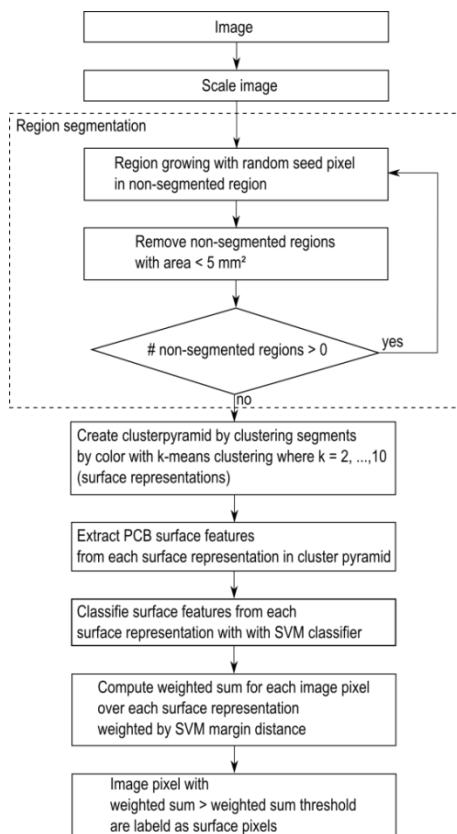


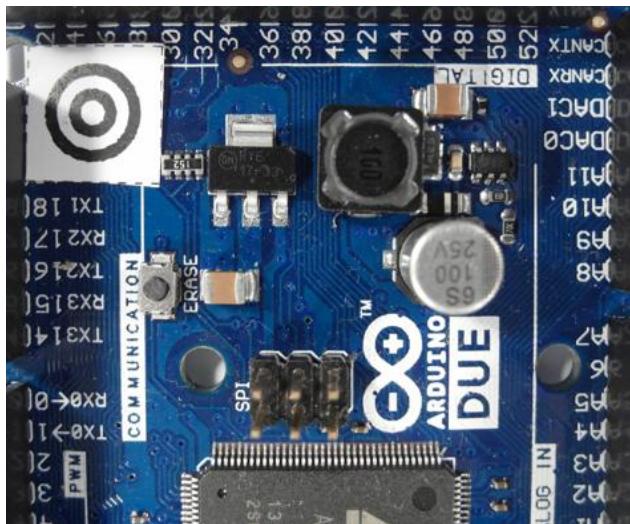
Figure 24: PCB surface segmentation process flow

The image is scaled to a resolution of 5 pixel/mm to speed up the PCB surface detection process. In the second process step a region growing approach is used to divide the image in regions with similar color. The seed points of the region growing algorithm are chosen randomly under the requirement that the seed points are placed in the non-segmented image region. The criterion to stop the growing process of a seed point is the similarity threshold value which is the Euclidian distance between the color of the neighboring pixel and the average

Methods for electronic component recognition

color of the region. Exceeds the distance a distance threshold value of 0.2, the neighboring pixel will not be considered as a region pixel. The growing process of a seed point stops if no neighboring is considered to the region. The region growing process is specified in chapter 2.1.1. After segmenting a region, all non-segmented regions with an area smaller than $5mm^2$ are removed from the non-segmented region to speed up the process. If there are still non-segmented regions, the region growing process is repeated with a new randomly selected seed point in the non-segmented region. If all image regions are segmented or rejected from the non-segmented region caused to their small region area the process stops.

The first 200 segments from the region segmentation process are shown in Figure 26.



Methods for electronic component recognition

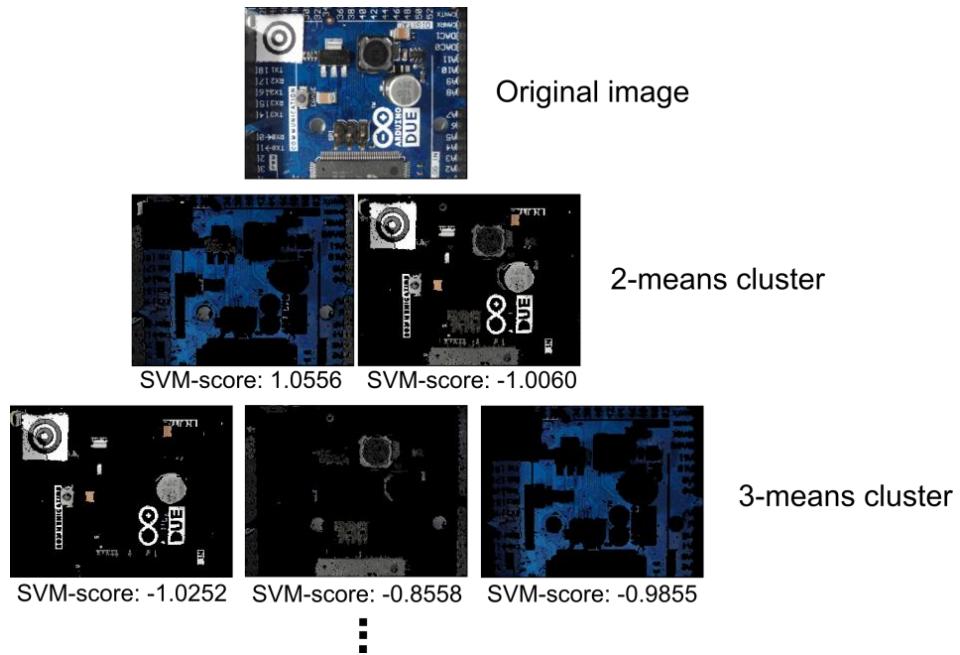


Figure 27: PCB surface cluster pyramid

After creating the cluster pyramid for all 54 surface representations ($2 + 3 + 4 + \dots + 10 = 54$) nine surface features are extracted which are:

- Color mean in all three color channels (3-Features)
- Number of surface representation pixel, normalized by the maximum number from surface representation pixels from a surface representation of the same pyramid level (1-Feature)
- Number of surface representation edge pixels normalized by the maximum number of surface representation edge pixels from the surface representations of the same pyramid level (Number of edge pixel is determined based on first derivative kernel in the gray scaled image) (1-Feature)
- Number of segments in the surface representation normalized by the maximum number of segments from the surface representations of the same pyramid level (1-Feature)
- Elements of the covariance matrix of color pixels from the surface representation (6-Features)

To separate good surface representations from bad ones, each surface representation is classified according to the nine features with an RBF-Kernel SVM ($\sigma = 1.0$, $C = 2.0$). The

Methods for electronic component recognition

RBF-SVM parameter where estimated with a grid search method and 3-fold cross validation on the surface training set.

To train the RBF-SVM, each surface representation in the cluster pyramid of 77 images were labeled according to their goodness of PCB surface representation. Surface representations in the cluster pyramid were labeled with 1 if the pixels represent mainly the surface and -1 if the pixels in the cluster are mainly pixel from electronic components or PCB markings. That results in a set of $54 * 77 = 4158$ clusters whereas 908 clusters were labeled as PCB surface and 3250 clusters were labeled as non-PCB surface representations. Ambiguous cluster representations were labeled as non-PCB surfaces.

The distances of the feature vectors from the decision boundary of the RBF-Kernel SVM were treated as scores $s_i, i = 1, \dots, 54$ whereas a high positive score identifies good surface representations and low negative scores represent bad surface representations. For each pixel of the image, the sum of scores over all 10 levels is computed. The scores are treated as weights of the surface representation in which the pixel was included. If pixels are not included in a cluster of a pyramid level because the region in which the pixel was included, was rejected caused by the small region area, the score is set to zero. Each Pixel $f(x, y)$ at the position x, y with score sum $w(x, y)$ greater than the weighted sum threshold w_{thr} is selected in the PCB surface set S .

$$w(x, y) = \sum_{i=1}^{54} s_i(x, y) \quad (51)$$

$$S = \{f(x, y) \mid w(x, y) > w_{thr}\} \quad (52)$$

In this approach w_{thr} was set to zero. All selected PCB surface pixel form the PCB surface.

3.2.3 Electronic component detection based on normalized 2D cross-correlation

Template matching is a technique in digital image processing for finding regions in an image that match a smaller image template. The normalized cross correlation is fast way of matching

Methods for electronic component recognition

templates in an image and is used in many object detection approaches. A detailed description about pattern matching with normalized 2D cross correlation is done in chapter 2.1.3.

In this approach the templates were generated by the training images of the electronic components. For each component the average values over all training images in all three color channels were computed. The average image is computed in the HSV color space and treated as the component template. The template of the DIP14 component is shown in Figure 20.



Figure 28: Image template for DIP14 component (RGB color space)

In this approach the spatial resolution depends on the component surface area. The relation between spatial image resolution and component surface for the normalized 2D-cross correlation is shown in Figure 29.

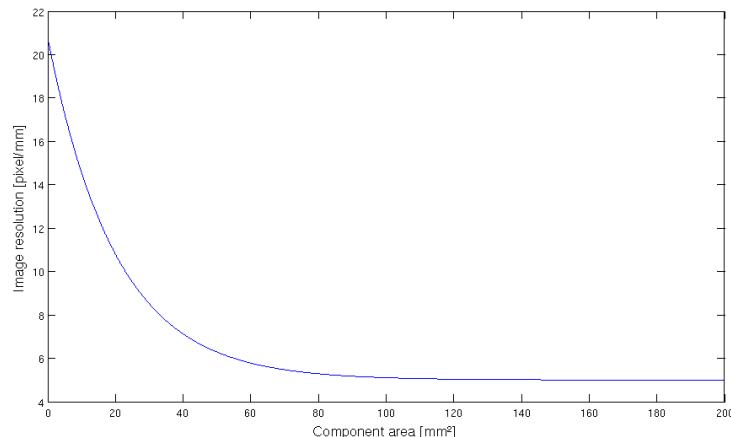


Figure 29: Spatial image resolution for 2D-cross correlation

To perform the 2D cross-correlation the examined image is also converted to the HSV color space and the cross-correlation is performed in all color channels. The average correlation values over all three color channels are determined and filtered by a 2D Gaussian kernel to get a score map $p(x, y)$. The Gaussian kernel has a size of $0.5 \text{ component}_{height} \times 0.5 \text{ component}_{width}$ and $\sigma = 1.5$. Scores $p(x, y)$ greater than a

Methods for electronic component recognition

correlation threshold $Corr_{thr}$ are treated as a set of potential component positions S . The correlation threshold $Corr_{thr} = 0.4$ seems to be a good trade off between false positive rate and true positive rate.

$$S = \{f(x, y) \mid p(x, y) > Corr_{thr}\} \quad (53)$$

An image and its determined potential component positions for the SOT223 component are shown in Figure 30 and Figure 31.

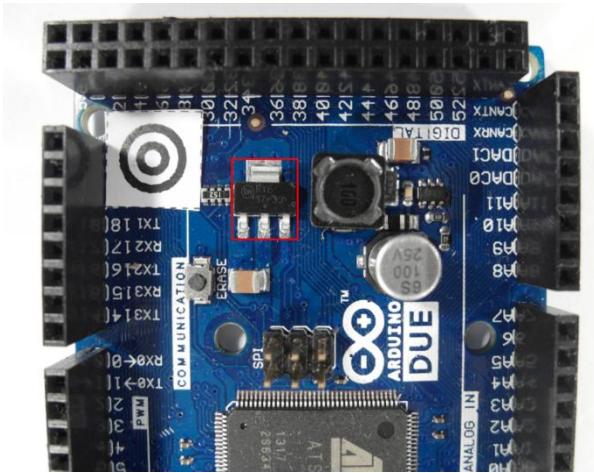


Figure 30: SOT223 transistor

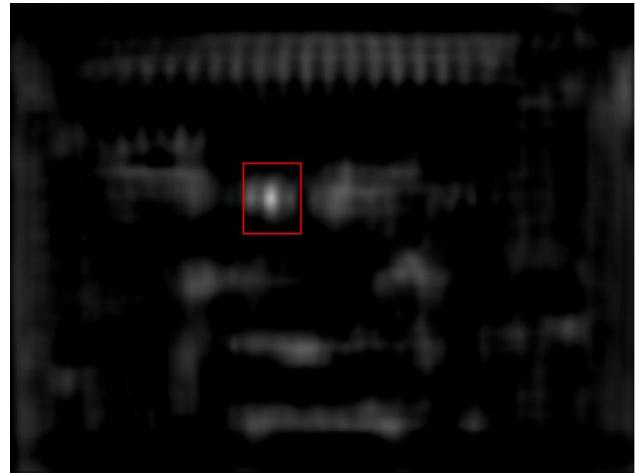


Figure 31: Determined potential component positions for SOT223 transistor

3.3 Feature extraction

In this approach features are extracted to measure values that are similar for one electronic component class and very different from electronic components from other classes. The four feature domains from which the features are extracted depends on the characteristic properties which distinct between the component classes.

The first feature domain consists of features which are extracted from the frequency domain and they are based on the idea that most of the electronic components have solder joints which are arranged equidistant at the border of the electronic components. This is typical for Integrated circuits (ICs) and can be measured in the values of specific Fourier coefficients.

The second feature domain is based on the idea that electronic components consist of different colors. A distinctive electronic component is the tantalum capacitor which is often

Methods for electronic component recognition

yellow/orange colored and differs from other components like ICs with their black/gray color. The color of the electronic components is measured by the image histogram values.

The third feature domain is based on the idea that electronic components consist of equally colored segments. These segments give some information of the spatial color distribution of the component, compared to the histogram based features which contain information about the global color distribution. The segments are extracted by a region growing approach which is based on image seed points. Measurements of the segmented regions (size, color, position) are used as features.

The fourth feature domain is based on the idea that Principal Components (PCs) can be used to compress optimally only the kind of images that were used to compute the principal components. The reconstruction error which was made if an image of a component is projected into the PCs and back again is measured as a value of reconstruction. This approach is applied on the edge images of the components to extract information about the edges of the component.

3.3.1 A priori knowledge generation

The extraction of representative features depends in two of the feature extraction algorithms on a priori knowledge. The a priori knowledge is generated by a subset of the training data (a priori subset) which was not used for feature extraction and classifier training.

A priori knowledge for seed point position estimation

The segmentation based feature extraction in chapter 3.3.4 requires seed point positions to extract segment features. Therefore the training set was spitted in two subsets, whereas the first subset (30% of the training data, the so classed a priori subset) was used to find important seed point positions and the second subset (70% of the training data) was used to extract features which were used for features extraction and classifier training from the most important seed points. To estimate the positions of seed points for the feature extraction step, a uniform grid of 30 seed points is created and features where extracted for each seed point according to the segment based feature extraction algorithm. The seed point grid for the ceramic capacitor 1210 is shown in Figure 32.

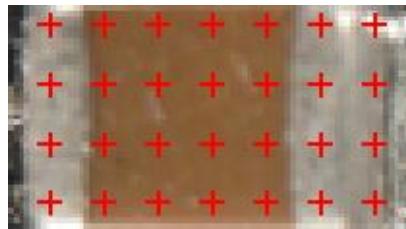


Figure 32: Seed point grid (30 seed points)

The fisher score feature selection method was used to select the 20 most important features. The seed points from which the most important features were extracted are used as seed point positions for the segment based feature extraction.

A priori knowledge for PCA reconstruction error based feature selection

The PCA reconstruction error based feature extraction requires the computation of principal components (PCs) from the component images and non-component images. Therefore a subset of the training images (30% of the training data, , the so classed a priori subset) is used to find principal components. The rest of the training set (70% of the training data) is used for the extraction of PCA reconstruction error based feature, specified in chapter 3.3.5 and is used to train the classifier. The projection matrix P_{ep} and the mean μ_{ep} are computed from the a priori subset of component images and the projection matrix P_{en} and mean μ_{en} are computed from the non-component images from the a priori subset. The computation of the matrices and the means are specified in chapter 2.1.4. The a priori data set is not used for training or testing the classifier.

3.3.2 Fourier coefficients based feature extraction

Every periodic infinite signal can be decomposed in trigonometric functions (Fourier transform). This transform can also be allied for 2D images. Fourier descriptors as features were already used in applications for face recognition and object recognition (deCampos, et al., 2000).

The idea of using Fourier coefficients as features comes from the shiny equidistant solder joints, which can be seen by most electronic component images. Many computer vision systems for solder joint detection, localization and segmentation have been developed. Specular reflections of solder and different shapes and size of solder joints makes it difficult to create a stable

Methods for electronic component recognition

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 33. Since the solder joints appear in the gray scaled image as bright equidistant spots there should be representative frequencies in the 2D Fourier spectrum with the period of the solder joint distance (pitch).



Figure 33: DIP14 package with equidistant solder joints

The 2D discrete Fourier transform for an $M \times N$ image is defined as

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (54)$$

$u = 0, 1, 2, \dots, M - 1$ and $v = 0, 1, 2, \dots, N - 1$ where $f(x, y)$ is the image of size $M \times N$ (Gonzalez, et al., 2006). 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 $M \times N$ image consists of $M \times N$ Fourier coefficients which produce $2 \times M \times N$ frequency features which are of interest. 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. 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 cutoff period is equivalent to the pitch of the package.

$$f_{cutoff} = \frac{1}{T_{cutoff}} = \frac{1}{pitch \text{ mm}} \quad (55)$$

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

$$\#frequency\ features = \left\lceil \frac{length\ [mm]}{T_{cutoff}\ [mm]} + 1 \right\rceil * \left\lceil \frac{width\ [mm]}{T_{cutoff}\ [mm]} + 1 \right\rceil \quad (56)$$

Another interesting feature extraction based on wavelets could analyze frequencies and their temporal occurrence which could improve the classification results.

3.3.3 Histogram based feature extraction

Color image segmentation algorithms for automated optical inspection in electronics have already been investigated (Tarnawski, 2003). Electronic components differ in color, such as several tantalum capacitors, ICs or SMD electrolyte capacitors. To find representative features the HSV (hue-saturation-value) color model was used because the channels are less correlated than the RGB color model and more stable against illumination changes or shadows (Cheng, et al., 2001), (Journal, 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 used as features that result in 30 color features. The histogram of a tantalum capacitor is seen in Figure 34, Figure 35, Figure 36 and Figure 37.

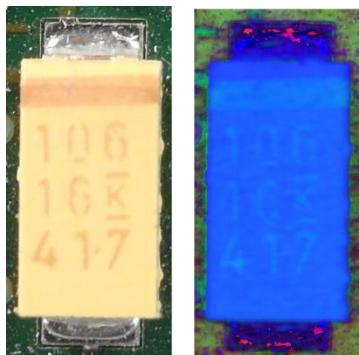


Figure 34: Tantalum capacitor in RGB color model (left) and HSV color model (right)

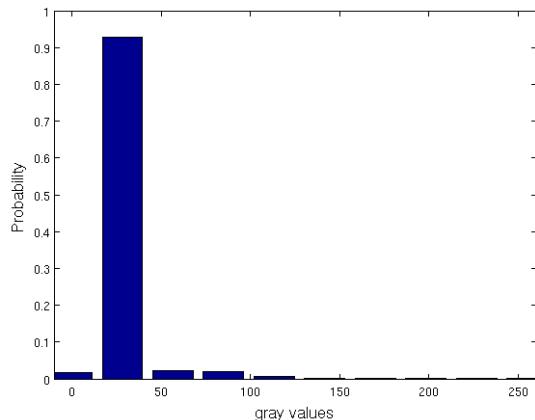


Figure 35: Normalized histogram of hue channel (tantalum capacitor)

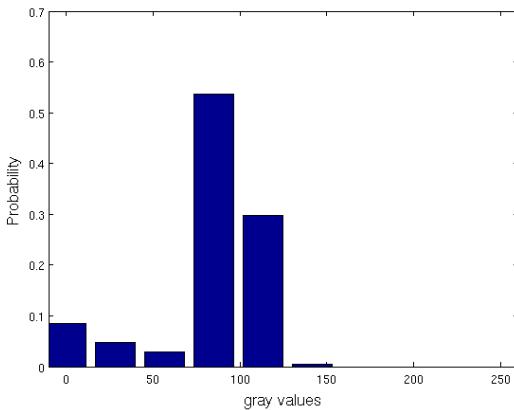


Figure 36: Normalized histogram of saturation channel (tantalum capacitor)

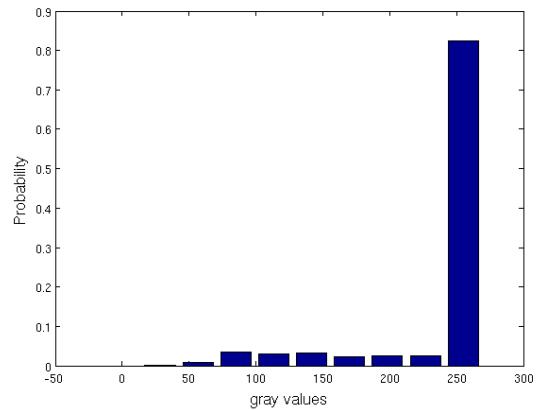


Figure 37: Normalized histogram of value channel (tantalum capacitor)

3.3.4 Segment based feature extraction

The segment based feature extraction is based on the idea that electronic components can be identified by characteristic color regions. One approach to extract information about spatial proximity of pixels is the region growing algorithm. The region growing starts with seed points where the pixel position is the most important drawback.

The seed point positions for the segment based feature extraction algorithm are determined by the a priori knowledge generation step specified in chapter 3.3.1. The region growing and feature extraction of the segments is done in HSV color space. In the region growing segmentation approach the neighboring pixel of the seed pixels are added to the segment if the distance between the color of the seed point and the neighboring pixel is smaller than a certain threshold value. This process is iterated until no more pixels are added to the new segment (Petrou, et al., 1999). As an example the multi-layer ceramic capacitor and important seed points are shown in Figure 38.

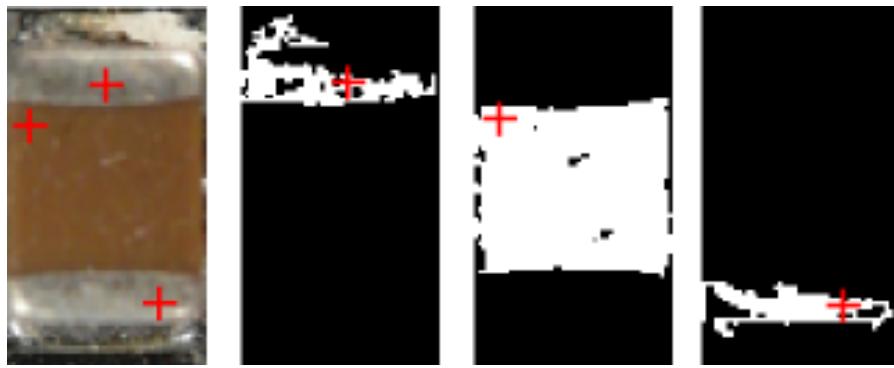


Figure 38: Three important seed points from the priori knowledge generation

Seven Features are extracted for every segmented region which are:

- x-coordinate of center of gravity (1 Feature)
- y-coordinate of center of gravity (1 Feature)
- Bounding box height (1 Feature)
- Bounding box width (1 Feature)
- Arithmetic mean color value in all three color channels (3 Feature)

3.3.5 PCA reconstruction error based feature extraction

Object detection based on image reconstruction with Principal Component Analyses was already applied for pedestrian recognition (Malag\{'o}n-Borja, et al., 2009). A similar approach was used to extract a PCA reconstruction feature. In this approach the PCA reconstruction is based on edge images of the component images. At first a subset of the training images of components are used to find principal components which can compress optimally only the kind of images that were used to compute the principal components. The estimation of the principal components is specified in chapter 3.3.1.

A set of PCs from images of 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 39 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 reconstruction error of the projection into the non-component PCs.



Figure 39: 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 depending to the size of the component. Afterwards 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 matrices and the image means $\mathbf{P}_{ep}, \boldsymbol{\mu}_{ep}$, $\mathbf{P}_{en}, \boldsymbol{\mu}_{en}$ are computed from the a priori subset specified in chapter 3.3.1. The reconstruction based on the component PC projection is computed by (57) and the reconstruction based on the non-component PC projection is computed by (58).

$$\mathbf{r}_{ep} = \mathbf{P}_{ep}^T \mathbf{P}_{ep} (\mathbf{e} - \boldsymbol{\mu}_{ep}) + \boldsymbol{\mu}_{ep} \quad (57)$$

$$\mathbf{r}_{en} = \mathbf{P}_{en}^T \mathbf{P}_{en} (\mathbf{e} - \boldsymbol{\mu}_{en}) + \boldsymbol{\mu}_{en} \quad (58)$$

The reconstruction error of component images projected by component PCs should be smaller for the component images than non-component images. The generated feature is the difference between the reconstruction error projected in the component PCs and the error projected in the non-component PCs shown in (59).

$$f_{pca} = \sum |\mathbf{r}_{ep} - \boldsymbol{\mu}_{ep}| - \sum |\mathbf{r}_{en} - \boldsymbol{\mu}_{en}| \quad (59)$$

Methods for electronic component recognition

The process is shown in Figure 40.

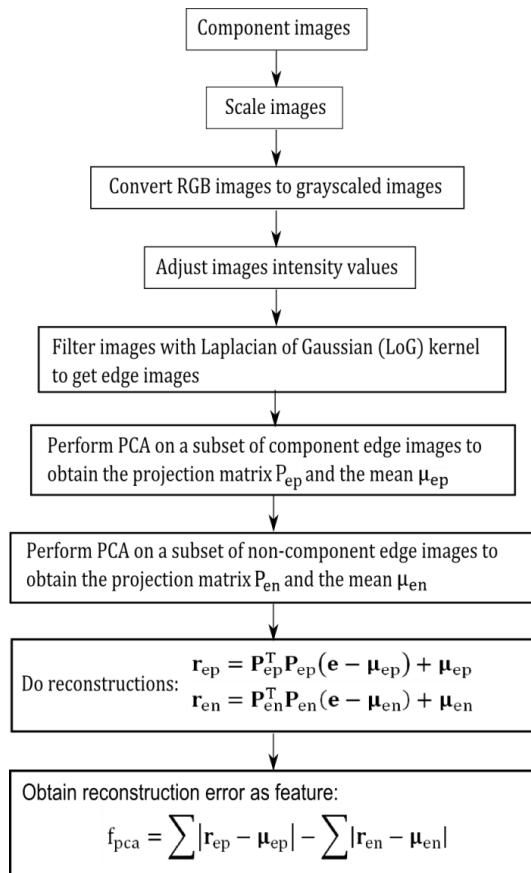


Figure 40: PCA feature construction process

3.4 Feature selection based on Fisher score and Random forest

In practice random forest cannot handle many 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 (Chen, et al.). 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 features selection step. In the second step the random forest based feature selection specified in chapter 2.2.2 is applied to select the most important features from the most important features from the first step. The process chain of the feature selection approach is shown in Figure 41.

Methods for electronic component recognition

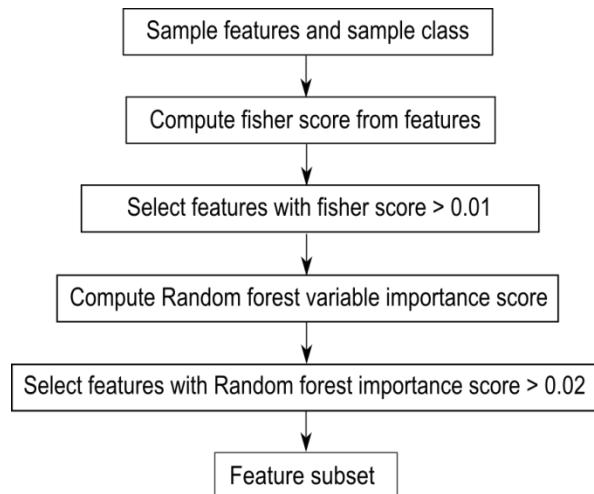


Figure 41: Feature selection process chain

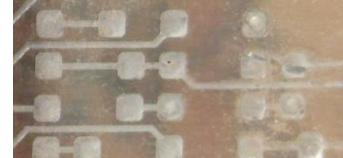
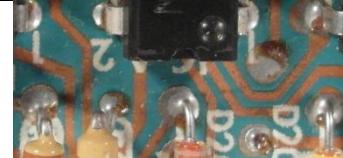
3.5 Classification

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

There are two approaches to select the non-component images in the dataset. The first approach is based on the idea that the component detection algorithm detects almost all component from the PCB image and that most of the components are in the recognition database. Under these requirements the non-component images consist of images from components of different component classes. The second approach is based on the idea that the non-component images should represent a plurality of possible images and therefore the non-component images are arbitrary image sections from the PCB images. An example of both approaches for the DIP14 component is shown in Table 2.

Methods for electronic component recognition

Table 2: Dataset approaches for non-component images

Component images for DIP14	Non-component images for DIP14 (images from different components)	Non-component images for DIP14 (images from arbitrary image section)
		
		

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

3.5.1 Random forest classifier

The random forest classification was implemented with the MATLAB Class “TreeBagger” which forms an ensemble of bagged decision trees. The number of trees was set to 100, this number seems to be enough compared to the dependency between misclassification rate and number of trees in Figure 67. The samples which were used to train a single decision tree were randomly selected with replacement. The number of samples which were used to create a decision tree is the root of the number of variables, which is default and used in many applications. All other parameters where set to default.

3.5.2 Support vector machines

The support vector machine is the second category of classifiers which was tested in this work for electronic component classification.

Linear support vector machine

The support vector machine classifier was implemented with the MATLAB function “`svmtrain`”. The data points were centered at the mean and scaled to have unit standard deviation before training. The box constrain C was determined by the grid search method specified in chapter 2.3.2. The linear kernel function was used to map the data into kernel space.

Support vector machine with RBF-kernel

The support vector machine with RBF-kernel was also implemented with the MATLAB function “`svmtrain`”. The data points were centered at the mean and scaled to have unit standard deviation before training. The box constrain C and the kernel parameter σ , which is the scaling factor for the radial basis function kernel were determined by the grid search method specified in chapter 2.3.2. The RBF kernel function was used to map the data into kernel space.

3.6 Data fusion model

The data fusion model for electronic component recognition is based on the following abstraction levels:

- Feature level fusion – selects the most important features from the extracted features from feature domains
- Classifier level fusion – fuses the outputs of the four classifiers from the four feature domains and the classifier of the most important features from all feature domains
- Decision level fusion – decides to which component class in the recognition database the component belongs. If the component does not belong to none of the classes it is classified an unknown component

The fusion levels are specified in the following chapters. The data fusion model is shown in Figure 42.

Methods for electronic component recognition

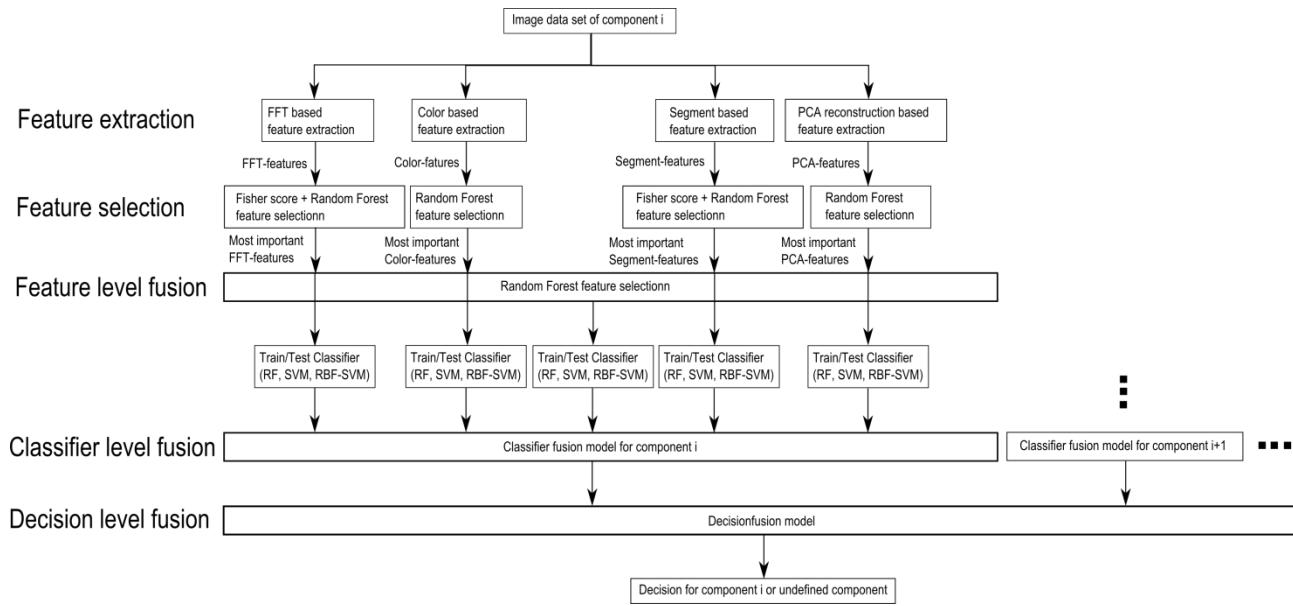


Figure 42: Data fusion model

3.6.1 Feature level fusion

The inputs for the feature selection process are the extracted features from the feature extraction process. The features are extracted from four different ranges of properties which are based on FFT features, color based features, segment based features and the PCA reconstruction error based feature.

The feature level fusion is based on the feature selection approach whereas 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.

The feature selection of the most important features from every feature range is based on the fisher score and random forest feature selection algorithm specified in chapter 3.4. The most important features from all four features ranges are merged together and a random forest feature selection is applied to get the most important features.

One of the main difficulties in merging features from different feature ranges is the problem of missing values. In this approach the features based on the PCA reconstruction error and the segmentation based features contain missing values caused by the fact that a subset of the

samples is used to generate a priori knowledge for the feature extraction process. The a priori knowledge generation is specified in chapter 3.3.1. The missing value of a sample from a variable m is replaced by the median over all samples from the variable m . The replacement values are called fills (Breiman, 2014). All missing values were used for training the classifier, so that the test data do not contain replaced values. This median replacement approach and alternatives for the replacement of missing values are specified in chapter 2.2.2.

After replacing missing values, the features with a random forest importance score greater than an importance score threshold of 0.02 are selected. The process is shown in the data fusion process chain in Figure 42.

3.6.2 Classifier level fusion

The data fusion on classifier level (classifier level fusion) is performed to make the recognition rate 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, 2014). In 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 the 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] (Ruta, et al., 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, 2014). 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 S_i which represents the probability that the sample is from class i . $S_{i,k}$ represents the score of

classifier k to be component i . $S_{i,k,test}$ represents the score of classifier k to be component i based on the true positive rate of the test set.

$$S_i = \sum_{k=1}^5 w_{i,k} * S_{i,k} \quad (60)$$

$$w_{i,k} = \frac{\sum_{j=1}^5 S_{i,j,test}}{S_{i,k,test}} \quad (61)$$

3.6.3 Decision level fusion with Dempster-Shafer theory

In this approach the outputs of the classifier fusion models at the classifier fusion level are soft outputs between 0 and 1. For the random forest classifier, the value $S_i = 0$ corresponds to the situation where zero percent of the trees from the classifier i decided that the component is from class i . The value $S_i = 1$ corresponds to the situation where all of the trees from the classifier i decided that the component is from class i . The output of the classifier fusion model for component i can be interpreted as a score that the detected component is from component i . The outputs from all classifier fusion outputs are combined to make a final decision to which component the examined component belongs (Dong, et al., 2009).

The dempster-shafer theory is based on probability assignments. The probability assignments are determined according to the distribution of the classifier outputs from a subset of the classifier test outcomes. Therefore the normal distribution parameter μ (mean value) and σ (standart deviation) where determined according to the classifier outcome and the maximum likelihood.

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (62)$$

The normal distribution of the Resistor network component classifier outcome of the test images are shown in Figure 43.

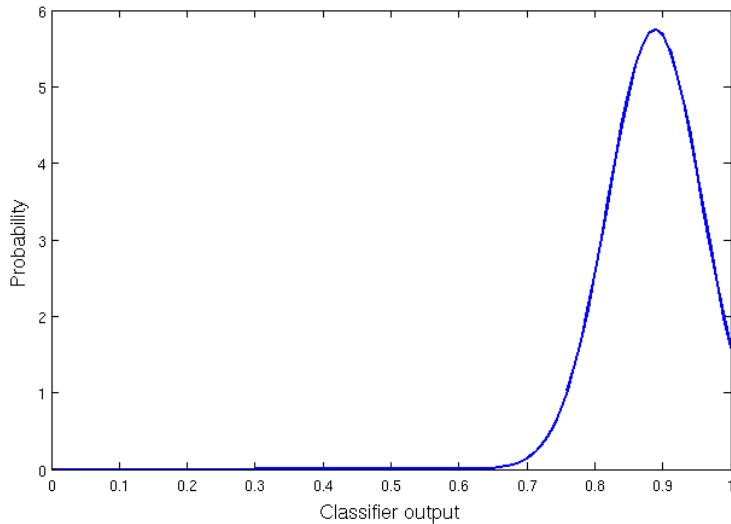


Figure 43: Normal distribution of Resistor network classifier (positive test data)

The basic probabilities are determined from the normal distribution and normalized as follows:

$$P_i = \frac{f(S_i, \mu, \sigma)}{f(S_i, \mu, \sigma) + f(1 - S_i, \mu, \sigma)} \quad (63)$$

$$1 - P_i = \frac{f(1 - S_i, \mu, \sigma)}{f(S_i, \mu, \sigma) + f(1 - S_i, \mu, \sigma)} \quad (64)$$

The number of possible component classes is n and consist of the number of classifiers (equal the number of components) s and additionally a class for unknown components ($n = s + 1$).

The basic probability assignment is made to each subset of the power set,

$$2^h = \{\emptyset, \{h_1\}, \dots, \{h_n\}, \{h_1, h_2\}, \dots, \{h_2, h_3\}, \{h_1, h_2, h_3\}, \dots, \{h_2, h_3, h_4\}, \dots, \Omega\} \quad (65)$$

whereas just the subset M contains masses unequal to zero:

$$M \subseteq 2^h \quad (66)$$

$$M = \{\{h_1\}, \{h_2, \dots, h_n\}, \{h_2\}, \{h_1, h_3, \dots, h_n\}, \dots, \{h_n\}, \{h_1, \dots, h_{n-1}\}\} \quad (67)$$

Methods for electronic component recognition

The set $\{h_1\}$ is the set of all components of the first component class. The set $\{h_1, \dots, h_n\}$ corresponds to the set of components which correspond to the or function of class one, class two, ..., class n .

$$\{h_1, \dots, h_n\} = \{h_1 \cap \dots \cap h_n\} \quad (68)$$

Each classifier i from the classifier fusion level assigns a basic probability to the hypotheses in the set M according to the following rule.

$$m_i(\{h_j\}) = P_i \text{ for } i = j \quad (69)$$

$$m_i(\{h_1, \dots, h_n\} \setminus \{h_j\}) = 1 - P_i \text{ for } i \neq j \quad (70)$$

Where

$$i = 1, \dots, s \text{ and } j = 1, \dots, n \text{ and } n = s + 1 \quad (71)$$

All other basic probability assignments to the sets in the power set 2^h are zero.

The assigned probabilities of all s classifiers are combined to assignments for the set of hypotheses according to the combination rule (41).

$$m_{1, \dots, s}(\{h_i\}) = K \cdot (1 - P_1) \cdot \dots \cdot (1 - P_{i-1}) \cdot P_i \cdot (1 - P_{i+1}) \cdot \dots \cdot (1 - P_s) \quad (72)$$

$$m_{1, \dots, s}(\{h_n\}) = K * (1 - P_1) \cdot \dots \cdot (1 - P_s) \quad (73)$$

The measure of contradiction K is calculated as follows:

$$K^{-1} = (1 - P_1) \cdot \dots \cdot (1 - P_s) \\ + \sum_{i=1}^s (1 - P_1) \cdot \dots \cdot (1 - P_{i-1}) \cdot P_i \cdot (1 - P_{i+1}) \cdot \dots \cdot (1 - P_s) \quad (74)$$

The belief according to (36) is calculated as follows:

$$bel(\{h_j\}) = m(h_j) \quad (75)$$

The plausibility according to (38) is calculated as follows:

$$pl(\{h_j\}) = m(h_j) \quad (76)$$

In that case the belief and the plausibility of class j are equal and the uncertainty is zero. The Dempster-Shafer theory corresponds to the Bayesian special case of the DS theory (Kohlas, et al.).

The component class with the maximum belief is assigned as component class.

Dempster-Shafer decision fusion example

The following example corresponds to a fusion system which consists of a database with two components. A new component should be classified. The component could be from the first component class (Component class 1), the second component class (Component class 2), or can be a component that is not in the database (Undefined class). The classifier fusion level outputs two values. The output $S_1 = 0.85$ is the output of the classifier $i = 1$ which gives a score that the component belongs to Component class 1. The output $S_2 = 0.12$ is the output of the classifier $i = 2$ which gives a score that the component belongs to Component class 2.

The probabilities assignments are determined according to the normal distribution. Let's assume the probabilities assignments are $P_1 = 0.9 = f(S_1, \mu_1, \sigma_1)$ and $P_2 = 0.2 = f(S_2, \mu_2, \sigma_2)$. The outputs from the normal distributions are shown in Table 3.

Table 3: Normal distribution outputs from outputs from classifier fusion level

	Component class 1	Component class 2	Undefined class
Classifier 1	$P_1 = 0.9$	-	-
Classifier 2	-	$P_2 = 0.2$	-

The power set is defined as follows

$$2^h = \{\emptyset, \{h_1\}, \{h_2\}, \{h_u\}, \{h_1, h_2\}, \{h_2, h_u\}, \{h_1, h_u\}, \{h_1, h_2, h_3\}\} \quad (77)$$

The set $\{h_1\}$ is the hypotheses that the component is from the first class, $\{h_2\}$ is the hypotheses that the component is from the second class and $\{h_u\}$ is the hypotheses that the component is from the undefined class. The basic probability assignments are shown in Table 4.

Methods for electronic component recognition

Table 4: Basic probability assignments

2^h	Classifier 1	Classifier 2
$\{h_1\}$	$m_1(\{h_1\}) = 0.9$	$m_2(\{h_1\}) = 0$
$\{h_2\}$	$m_1(\{h_2\}) = 0$	$m_2(\{h_2\}) = 0.2$
$\{h_u\}$	$m_1(\{h_u\}) = 0$	$m_2(\{h_u\}) = 0$
$\{h_1 \cup h_2\}$	$m_1(\{h_1 \cup h_2\}) = 0$	$m_2(\{h_1 \cup h_2\}) = 0$
$\{h_2 \cup h_u\}$	$m_1(\{h_2 \cup h_u\}) = 1 - m_1(\{h_1\}) = 0.1$	$m_2(\{h_2 \cup h_u\}) = 0$
$\{h_1 \cup h_u\}$	$m_1(\{h_1 \cup h_u\}) = 0$	$m_2(\{h_1 \cup h_u\}) = 1 - m_2(\{h_2\}) = 0.8$
$\{h_1 \cup h_2 \cup h_u\}$	$m_1(\{h_1 \cup h_2 \cup h_u\}) = 0$	$m_2(\{h_1 \cup h_2 \cup h_u\}) = 0$

The measure of contradiction K is determined according to (74) as follows:

$$\begin{aligned}
 K^{-1} &= m_1(\{h_1\}) \cdot m_2(\{h_1 \cup h_u\}) + m_1(\{h_2 \cup h_u\}) \cdot m_2(\{h_2\}) \\
 &\quad + m_1(\{h_2 \cup h_u\}) \cdot m_2(\{h_2\}) = 0.9 \cdot 0.8 + 0.1 \cdot 0.2 + 0.8 \cdot 0.1 \\
 &= 0.82
 \end{aligned} \tag{78}$$

The combination of the assigned probabilities according to (72) and (73) is done as follows:

$$m_{1,2}(\{h_1\}) = K \cdot m_1(\{h_1\}) \cdot m_2(\{h_1 \cup h_u\}) = \frac{0.9 \cdot 0.8}{0.82} = 0.8780 \tag{79}$$

$$m_{1,2}(\{h_2\}) = K \cdot m_2(\{h_2\}) \cdot m_1(\{h_2 \cup h_u\}) = \frac{0.2 \cdot 0.1}{0.82} = 0.0244 \tag{80}$$

$$m_{1,2}(\{h_u\}) = K \cdot m_2(\{h_2\}) \cdot m_1(\{h_2 \cup h_u\}) = \frac{0.8 \cdot 0.1}{0.82} = 0.0976 \tag{81}$$

All other subsets of 2^h have zero basic assignment probability. The belief and plausibility for the component classes are shown in Table 5.

Methods for electronic component recognition

Table 5: Belief and plausibility of component classes

	Belief	Plausibility
Component class 1	$bel(\{h_1\}) = \sum_{A \subseteq \{h_1\}} m(A) =$ $m(\{h_1\}) = 0.878$	$pl(\{h_1\}) = \sum_{A \cap \{h_1\} \neq \emptyset} m(A) = m(\{h_1\})$ $= 0.878$
Component class 2	$bel(\{h_2\}) = \sum_{A \subseteq \{h_2\}} m(A) = m(\{h_2\})$ $= 0.0244$	$pl(\{h_2\}) = \sum_{A \cap \{h_2\} \neq \emptyset} m(A) = m(\{h_2\})$ $= 0.0244$
Undefined class	$bel(\{h_u\}) = \sum_{A \subseteq \{h_u\}} m(A) = m(\{h_u\})$ $= 0.0976$	$pl(\{h_u\}) = \sum_{A \cap \{h_u\} \neq \emptyset} m(A) = m(\{h_u\})$ $= 0.0976$

The component class with the maximum belief (component class 1) is assigned as component class.

3.7 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, passport check 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 (Luo, 2014). This approach is focusing on the inspection of IC markings whereas markings of other components like capacitors or coils are out of focus because of their complexity.

3.7.1 Optical character recognition difficulties

The difference between the inspection of IC markings in a PCB assembly line, lies in the quality of the IC 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

Methods for electronic component recognition

recognition of IC markings are caused by the fact that the ICs are from PCB scrap but they are also universal for similar OCR tasks.

- Company logos or symbols in character lines
- Symbols for component orientation confuse OCR software
- Dirt disturbs 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 44.



Figure 44: Difficulties of IC marking recognition

3.7.2 Optical character recognition flow chart

The most important step of this OCR approach is the character classification step where the binarized image of characters is mapped to the recognized ASCII 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 (Blaes, et al.). 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 46 were carried out.

Component properties which have to be known for the OCR algorithm and which 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 46.

The input of the process is the already recognized component image. At first the OCR-ROI is selected from the component 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 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 (83) and (82). The approximated discretized kernel mask is of size $h \times h$ 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 in real word coordinates ($h = 1\text{mm} * \text{imagescale} [\text{pixel/mm}]$) and in practice lies between 50 and 120 pixels. The standard deviation of the Gaussian is constant $\sigma = 0.5$.

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (82)$$

$$\nabla G(x, y) = \frac{\partial^2 G(x, y)}{\partial x^2} + \frac{\partial^2 G(x, y)}{\partial y^2} \quad (83)$$

The next step is the blob segmentation which is done by Otsu's segmentation method (Otsu, 1979). 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 the segmentation step a morphologic closing operator is applied to reduce holes in the character blobs. The size of the rectangular closing kernel changes linear with the image scale $h = 0.05 \text{ mm} * \text{imagescale} [\text{pixel/mm}]$.

Blobs that do not correspond to a character still exist in the segmented image. Therefore the area of the blobs is estimated and blobs with an area smaller area_{\min} and blobs with an area greater area_{\max} 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 estimating the probability density of the character position which is done with the MATLAB function `ksdensity`. The function returns a probability density estimation 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 σ 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 $\text{distance}_{\text{char}} \text{, potential baseline} = 0.25\text{mm}$. All characters that distance from baseline is smaller than $\text{distance}_{\text{char,baseline}}$ 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 component 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

Methods for electronic component recognition

baseline model with a distance error from the baseline smaller $distance_{char,baseline} = 0.1mm$. This is done with the lower and upper baseline of the character lines. The RANSAC algorithm is specified in chapter 2.5.2

Baselines with a number of assigned blobs less than three 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 OCRMax. The output of this software is the recognized word from the image. A comparison of the two OCR engines Tesseract and OCRMax is done in 5.6.2. The settings and difficulties of the two OCR engines are mentioned in 3.7.1. An example of the OCR process of a QFP144 from the grayscaled image to the segmented character lines (words) is shown in Figure 45



Figure 45: OCR of a QFP144 from top left to top right: grayscaled image, LoG filtered image, binarized image, blobs filtered image. From bottom left to bottom right: four character lines (words)

Methods for electronic component recognition

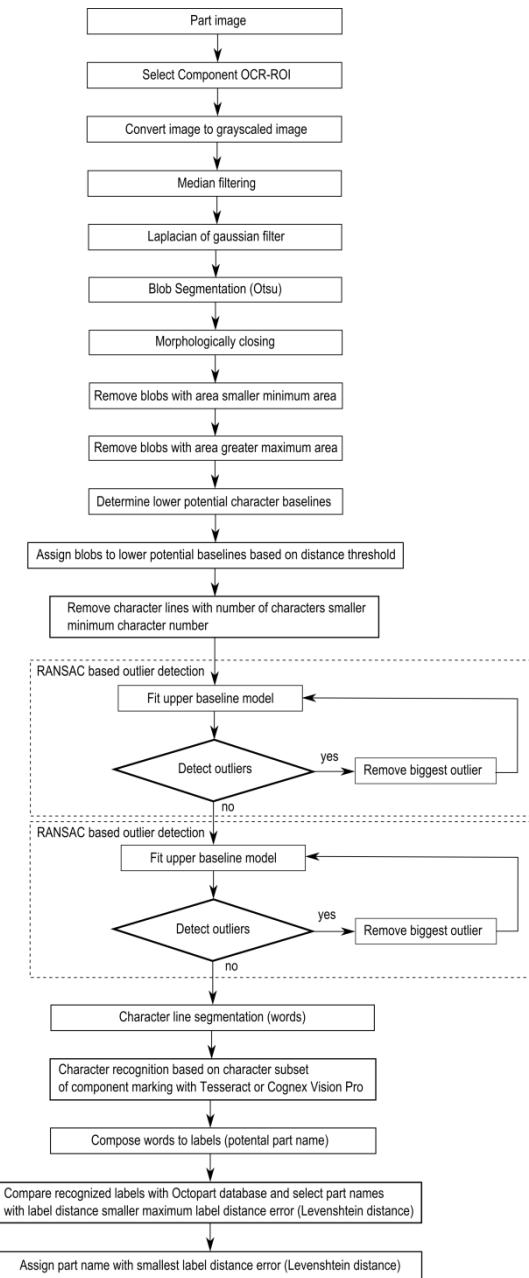


Figure 46: IC marking recognition flow chart

OCR engine Tesseract 3.0.1

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 various platforms. Since 2006 Tesseract development was sponsored by Google and provides support for various languages. A

Methods for electronic component recognition

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 (Heliński, et al., 2000).

For character recognition with Tesseract, the markings were decomposed in lines referring to the flow chart in Figure 46. The segmented binarized character line images were transferred to the Tesseract engine by the command-line interface in MATLAB and the recognized results were stored in a text file. Tesseract was trained with 1704 characters from 146 IC markings. The following settings were made to improve the accuracy rate.

- Character limitation subset was set to "0123456789ABCDEFGHIJKLMNPQRSTUVWXYZ/"
- Tesseract pagesegmode: 7 = Treat the image as single text line

OCR engine Cognex OCRMax

Cognex image processing software suite VisionPro® includes OCRMax™ which is a font-trainable OCR and OCV (Optical character recognition and Optical character verification) tool (VisionPro, 2014). In this approach the OCR engine OCRMax™ was used to recognize characters from segmented binarized character line images similar to the Tesseract OCR engine. A training data set was composed consisting of electronic component markings. The Software was trained with 1704 characters from 146 IC markings. The following settings were made to improve the accuracy rate.

- Character limitation subset was set to "0123456789ABCDEFGHIJKLMNPQRSTUVWXYZ/"

The character level accuracy of both OCR engines depends on the number of characters which were used to train the Tesseract OCR engine and the OCRMax engine. In this approach characters from 37 classes:

$$character\ set = \quad (84)$$

$$\{0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, /\}$$

where used to train the OCR engines. The Dependency of the Tesseract character recognition accuracy of tested characters from the number of characters used for training is shown in Figure 47. It can be seen that the accuracy rate converges and therefore the OCR character recognition accuracy will not increase significantly by training the OCR engine with more characters.

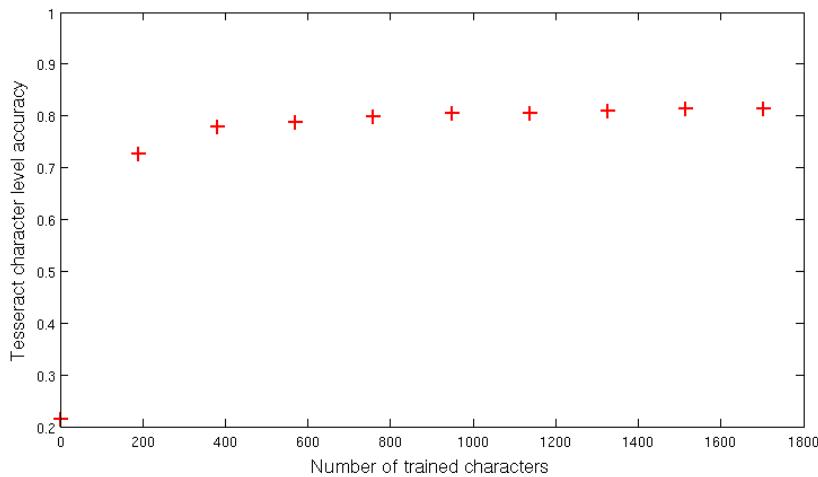


Figure 47: Dependency of Tesseract character recognition accuracy from number of characters

3.7.3 Optical character recognition evaluation scheme

To evaluate the OCR approach with the OCR engines, the analysis of the OCR results are compared on different levels.

Character level evaluation

The lowest level is the character level whereas each word is recognized by an OCR engine and compared to the manual assigned word. The two words are compared based on the Levenshtein distance which is a string metric for measuring the difference between two sequences and is specified in chapter 2.5.1. The error $e_{c,i}$ is the number of character errors (insertions, substitutions and deletions) of the component marking i . An example of OCR the evaluation on character level with Levenshtein distance is shown in Figure 48.

Methods for electronic component recognition

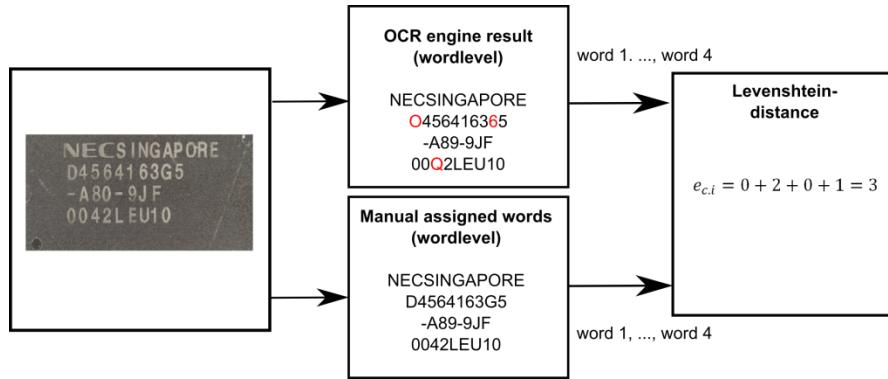


Figure 48: OCR evaluation on character level

Word level evaluation without Octopart database

The word level evaluation is based on correctly assigned words. If two words are not equal (at least one character is not equal) the number of word errors $e_{w,i}$ of the component marking i increases by one. The word was not verified by the electronic component database Octopart. An example of OCR evaluation on word level is shown in Figure 49.

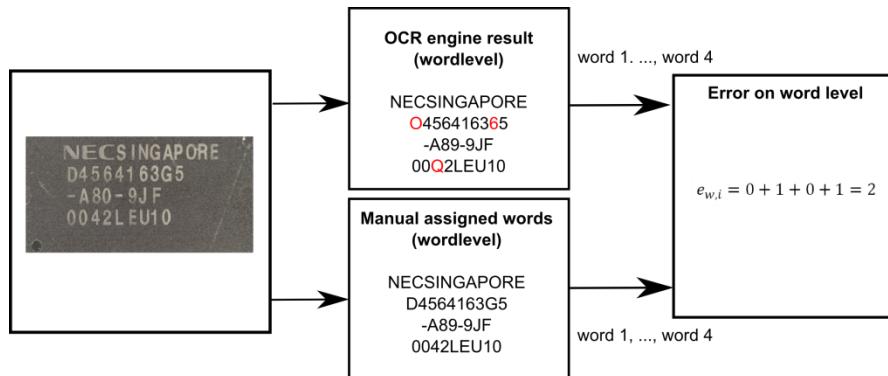


Figure 49: OCR evaluation on word level without Octopart

Word level evaluation with Octopart database

The word level verification with Octopart database is shown in Figure 50. The difference between the OCR evaluations on word level without Octopart database is the assignment of one of the two classes (component -name, non-component -name) to the words. In Figure 50 the component -names are marked in black and the non-component names are marked in red.

Methods for electronic component recognition

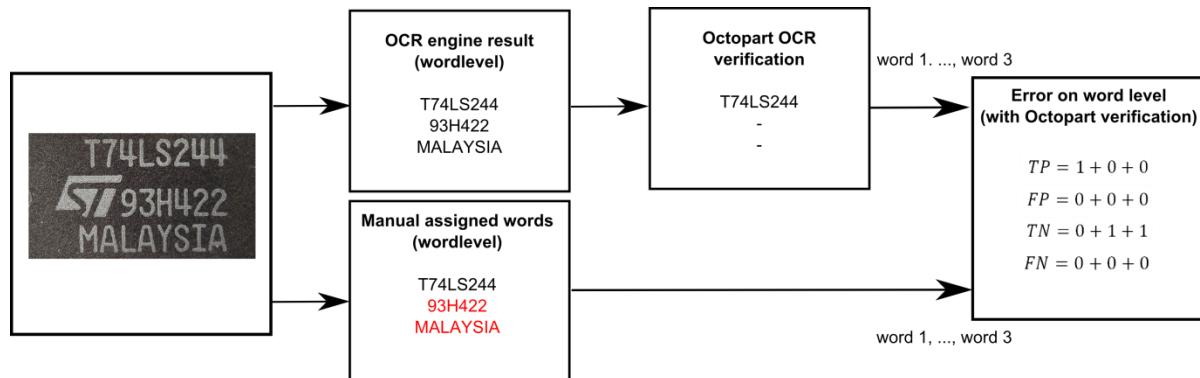


Figure 50: OCR evaluation on word level with Octopart

All recognized words from the OCR engines are requested at the Octopart database to verify if the word is a component -name or a non-component -name. If the Octopart database response a component-name with the same or similar component-name (Levenshtein distance < 2) the OCR system references the word to a component in the Octopart database. If the assigned class to a word is a component-name and the Octopart database response a component-name, the true positive rate (TP) increases by one. If a word was assigned as component-name but Octopart cannot match the word to a component in the database the false negative rate (FN) increases by one. If a word was assigned as non-component-name and the Octopart database could not reference the word to a component in the database, the true negative rate (TN) increases by one. If a word was assigned as non-component-name and the Octopart database referenced the word to a component in the database, the false positive rate (FP) increases by one.

Label level evaluation with Octopart database

The label level evaluation is done because of the fact that component-names are sometimes composed of multiple words. On the label level evaluation the words are composed to labels and requested at the Octopart database. If the assigned class to a label is a component-name and the Octopart database response a component-name, the true positive rate (TP) increases by one. If a label was assigned as component-name but Octopart cannot match the word to a component in the database the false negative rate (FN) increases by one. If a label was assigned as non-component-name and the Octopart database could not reference the word to a component in the database, the true negative rate (TN) increases by one. If a label was assigned

Methods for electronic component recognition

as non-component-name and the Octopart database referenced the label to a component in the database, the false positive rate (FP) increases by one. An example of OCR evaluation on label level is shown in Figure 51.

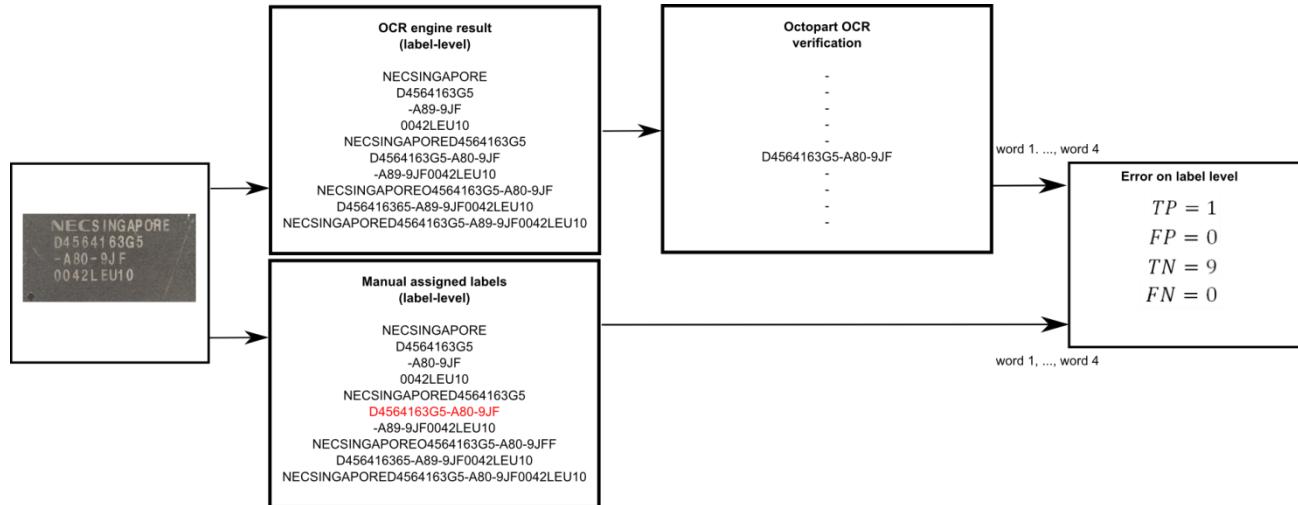


Figure 51: OCR evaluation on label level with Octopart

Component level evaluation with Octopart database

The accuracy rate on part-level shows how many parts were assigned correctly to a component in the Octopart database whereas the potential component-names were evaluated first at label level. If at least one label (potential component-name) was correctly assigned to a component in the Octopart database the True part assignment rate (TPA) increase by one. If no label was correctly assigned to a part in the Octopart database, the false part assignment rate (FPA) increases by one. An example of two components is shown in Figure 52.

Methods for electronic component recognition

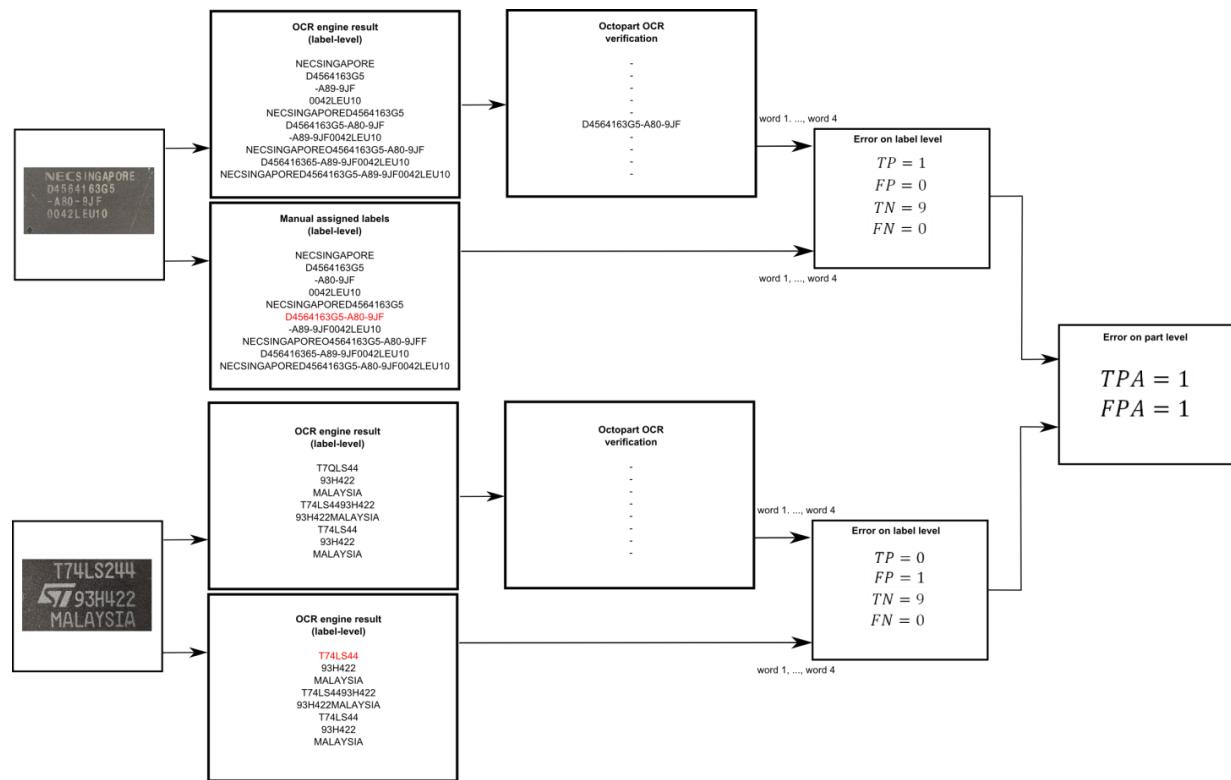


Figure 52: OCR evaluation on part level with Octopart

The OCR evaluation results on character level, word level and part level of the OCR system are shown in detail in chapter 5.6.2.

4. Life-cycle inventory model analyses of printed circuit boards

An improved recycling process requires a precise model of the PCBs and its material composition. The knowledge about the content of valuable materials in electronic components is of particular importance. To generate a precious composition model and an LCI-model, information about electronic components on the PCB is necessary. The automatic optical inspection approach in this work can help to estimate these models. The ILCD-format is used to automatically create and transfer the LCI-model and the composition model into common LCA-Software like GaBi or OpenLCA.

4.1 Printed circuit board region classification based on electronic component recognition results

In this work an LCI-PCB-model and a PCB-composition-model are automatically generated. Both models are based on the determination of four regions of the PCB. The regions are based on the surface of the PCB and the electronic components. The PCB is divided in the following four regions:

- 1) PCB support material (epoxy) - $A_{PCB,surface}$
- 2) Component was detected and classified as unknown component or PCB areas could not be recognized as PCB support material and therefore are treated as unknown components $A_{PCB,mounted}$
- 3) Component was detected and classified as a known electronic component (SOT223, Resistor network, etc.) but component markings could not be recognized or components do not have of any component markings
- 4) Component was detected, classified and a component marking was recognized

The four PCB regions for a sub image of the Arduino Due board are shown in Figure 53. The red colored regions are components which are detected, classified and the component marking was recognized. The green colored components are detected components which were classified as known component class but the marking could not be recognized or the component does not consist of a marking. The yellow colored regions are components which were detected but where classified as unknown component class or could not be recognized as PCB support

material and therefore are treated as unknown components. PCB support material is the blue colored region in the PCB image.

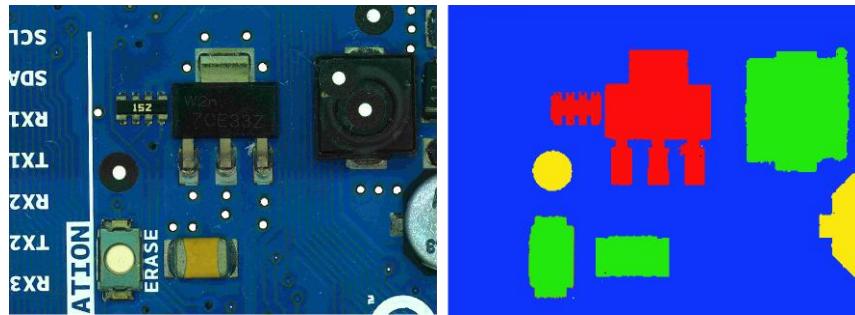


Figure 53: PCB model regions

4.2 Estimated PCB-LCI model and PCB-composition model

The PCB is divided in four regions based on the region definition in chapter 4.1 and the component detection and component classification results of the PCB.

The ILCD format is used to import ILCD models of the electronic components from databases, create automatically PCB models in MATLAB and export the model. They can be imported in any LCA software which supports imports of ILCD data like GaBi or OpenLCA. The LCI-model in this approach is a generalized model for printed circuit boards and is developed to handle PCBs from scrap automatically. There are two ILCD-PCB models which are created and can be imported in LCA software, the PCB-LCI model and the PCB-composition model.

The PCB-LCI model represents the LCI model of the PCB and uses full aggregated data to quantify energy, raw material requirements, emissions, solid waste and other releases. The flow diagram for a generalize PCB model is shown in Figure 54. The PCB consists of the four different PCB regions which are modeled as follows:

- 1) Leiterplatte (FR4;2l;2s)
- 2) Leiterplatte 2-Lagen starr FR4 mit HASL Finish (subtractive Methode)
- 3) ILCD component package from GaBi database (Electronic component)
- 4) ILCD component package from GaBi database

Life-cycle inventory model analyses of printed circuit boards

Solder paste (Lotpaste SnAg3.6) is additionally added to the PCB model.

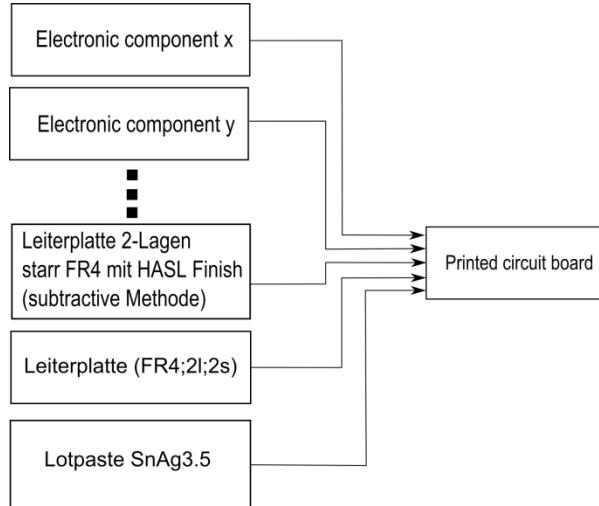


Figure 54: PCB flow diagram for LCI-model

For the PCB-LCI model ,the full aggregated ILCD process models of the components, the “Printed wiring board HASL 2-layer (subtractive method)”, the PCB support material model “Leiterplatte (FR4;2l;2s)” and the “Lotpaste SnAg3.6” model are added to the LCI model. The main process is the printed circuit board process with the flow inputs from all other processes.

Each flow owns flow properties with information about the amount of the composed materials. An additional flow property containing the purchase price of the component which is added to the flow properties, if the price could be estimated with the component marking recognition and the Octopart database. The estimation of the purchase price can help recyclers to determine components which could be valuable for component reuse.

The PCB-LCI model is exported as an ILCD model and can be imported in any LCA software which supports ILCD import.

The PCB-composition 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. The model quantifies the amount of materials which are included in the electronic component (gold, palladium, ceramic, plastic, etc.). 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 55. The flows in the figure between the PCB components and the materials are symbolic and depend on the content of the components.

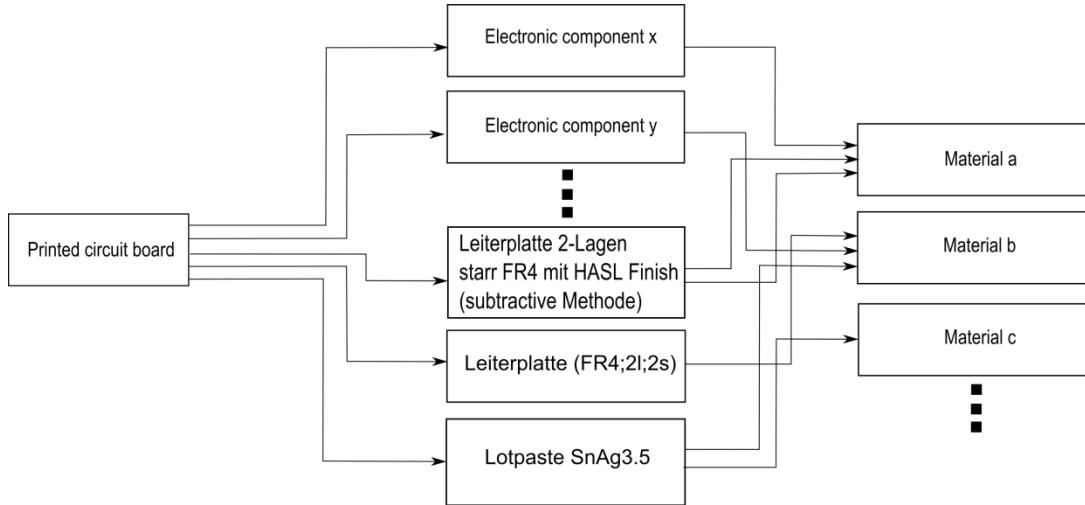


Figure 55: PCB flow diagram for composition model

The PCB-composition model data are mainly extracted from the flow properties of the electronic components. Each component flow owns flow properties which include information about the material composition of the component. This flow properties (amount of gold, palladium, silver ...in the component) where extracted and used to create new flows based on this materials. For each component a new process is created with the flow inputs which were created of the flow properties. The new processes are models of the material compositions of the components.

4.3 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 from the GaBi database is assigned. In this work the ILCD models of the components where exported from the GaBi database, but any ILCD component model can be used. Most of the database models are based on the component package and are independent from the function of the electronic component.

The amount of the model components is determined according to the detected areas or the number of classified components. The PCB support material (PCB surface area) is determined as $A_{PCB,surface}$ and is modeled as “Leiterplatte (FR4;2l;2s)”. The amount unit is mass and is calculated by the region area recognized in the image and the basis weight. The basis weight $w_{PCB,surface} = 3,92 \frac{kg}{m^2}$ is based on the information on <http://www.leiton.de> (Leiton, 2014).

$$N_{PCB,surface} = w_{PCB,surface} * A_{PCB,surface} \quad (85)$$

$A_{PCB,surface}$ – Area of PCB support material [m^2]

$N_{PCB,surface}$ – Amount of PCB support material [kg]

Electronic components which are detected, but the component was classified as unknown component based on the recognition database or areas and PCB support material could not be recognized are modeled by the “Printed wiring board HASL 2-layer (subtractive method)”. The amount is calculated by the region area recognized in the image and the basis weight. The basis weight $w_{PCB,mounted} = 0,75 \frac{g}{cm^2} = 7,5 \frac{kg}{mm^2}$ was determined by the average value of 25 PCBs which are listed in Appendix F.

$$N_{PCB,mounted} = w_{PCB,mounted} * A_{PCB,mounted} \quad (86)$$

$A_{PCB,mounted}$ – Area of PCB mounted components (unknown components) [m^2]

$N_{PCB,surface}$ – Amount of PCB mounted components (unknown components) [kg]

Detected and classified electronic components are modeled by the ILCD component models which are assigned in the recognition database. 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 merely in size, the amount of the component is scaled by mass and assigned to the component in the recognition database see (87).

Life-cycle inventory model analyses of printed circuit boards

$$N_{PCB,component,model} = N_{PCB,component} * \frac{m_{PCB,component}}{m_{GaBi,component}} \quad (87)$$

$m_{PCB,mounted}$ – Mass of the component [kg]

$m_{GaBi,component}$ – Mass of the component in GaBi database [kg]

$N_{PCB,component}$ – Number of a specific component on the PCB board [N]

$N_{PCB,component,model}$ – Number of a specific component in the PCB model [N]

Solder paste is modeled by “Lotpaste SnAg3.5” and the amount is determined as follows:

$$N_{PCB,solder} = w_{PCB,solder} * A_{PCB} \quad (88)$$

A_{PCB} – Area of PCB [m^2]

$N_{PCB,solder}$ – Amount of solder paste in the PCB model [kg]

The basis weight of $w_{PCB,solder} = 0,5 \frac{kg}{m^2}$ was determined as the basis weight of the solder paste based on the area of PCB.

5. Implementation and experiments

The electronic component recognition algorithm was mainly implemented under MATLAB® 2010a with additional MATLAB toolboxes. The implementation was not focused on runtime efficiency to prevent restriction of code readability and changeability.

The character classification step in the optical character recognition system was done by the Cognex VisionPro OCRMax engine and Tesseract 3.02 engine.

The communication between MATLAB and the electronic component database Octopart for component name verification was done by the software tool “cURL” which is command line tool for getting or sending URL syntax. It is based on “libcurl” which is a free client-side URL transfer library.

The ILCD models for material composition estimation of the electronic components in the recognition database are exported from the GaBi 6 Extension database XI: Electronics from PE INTERNATIONAL.

5.1 Dataset creation

The recognition dataset consist of 15 electronic components which are analyzed and 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. 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 components of a component and different printed circuit boards, to create a representative dataset.

Additional important information and properties of the components are listed in Table 6.

Implementation and experiments

Table 6: Component properties

Component properties	Description
Package properties	
Component length	Length of the component package [mm]
Component width	Width of the component package [mm]
Component border size	Size of the border which is cropped with the component image [mm]
Package DOF	Degree of freedom of the component rotation in 90° steps (between one and four)
OCR properties	
ROI for optical character recognition	Region of interest of the component marking (based on the upper left component corner)
Subset of characters for optical character recognition	Subset of characters which can be included in the component marking
Maximum and minimum number of OCR lines	Maximum number of character lines of the component marking
Frequency features generation properties	
Image scale for frequency feature generation	Computed according to chapter 3.1.3.
Number of maximum Fourier coefficient features	Computed according to formula (56)
Border cut information	Boolean if border pixel is selected by frequency feature extraction algorithm (true/false)
Color histogram features	
Image scale for histogram feature generation	Computed according to chapter 3.1.3.
Number of histogram bins	Default: 10
Segment features	
Image scale for histogram feature generation	Computed according to chapter 3.1.3.

Implementation and experiments

Number of initial seed points for region growing approach	Computed according to chapter 3.3.1
PCA reconstruction features	
Image scale for histogram feature generation	Computed according to chapter 3.1.3.
Number of principal components (PCs)	Default: 50
LCI properties	
ILCD-model full aggregated model	Extracted from GaBi database
ILCD-model composition model	Extracted from GaBi database

To detect the edges of the component border, border pixels are also selected from the printed circuit board images as it can be seen in Figure 56.

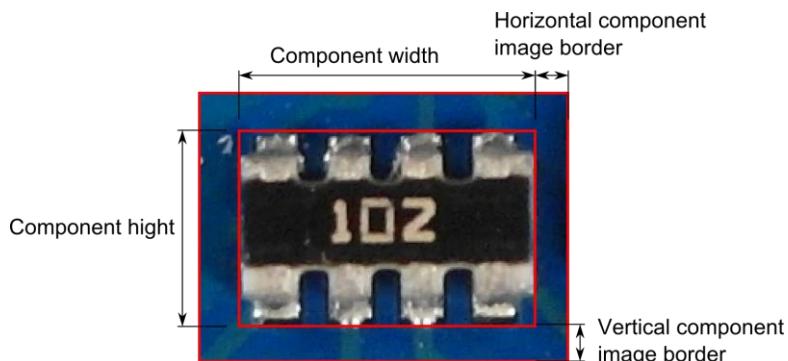


Figure 56: Component border definition

A section of the component database images is shown in Figure 57.

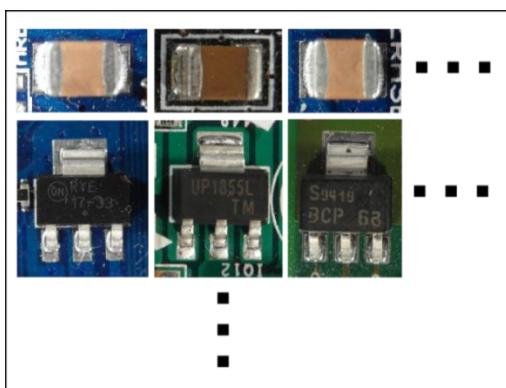


Figure 57: Database section

Implementation and experiments

The ILCD-model of each component which contains the full aggregated data and the composition model comes from the GaBi Extension database XI: Electronics from PE INTERNATIONAL. The verification of the component names is based on the Octopart database API.

5.1.1 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 (Puchheim, 2010). 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 58.



Figure 58: Image acquisition system

The camera is a consumer camera and no industry camera with adjustable depth of focus and a massive distortion by a small working distance. Therefore a region of interest (ROI) placed in the center of the image is used for electronic component recognition. Shadows from height electronic components disturbed the uniformly illumination, decrease the recognition rate and decrease the OCR accuracy.

Implementation and experiments

5.1.2 Dataset composition

The dataset used in the experiments consist of 1982 component images from 15 component classes. The dataset composition is shown in Table 7.

Table 7: Dataset composition

	Number of non-component images in training dataset	Number of component images in training dataset	Number of non-component images in test dataset	Number of component images in test dataset
Tantalum capacitor	30	30	29	29
SMD Aluminum electrolytic capacitor	56	56	56	56
QFP100	40	40	39	39
SMD Resistor Network array 1206, 4 Resistors	133	133	133	133
SMD Transistor SOT23-3	131	131	131	131
DIP14	57	57	57	57
DIP16	36	36	36	36
SMD Resistor 1206	133	133	133	133
SOIC-8	53	53	53	53
Ceramic capacitor 1210	21	21	21	21
SOT223-3	69	69	57	68
SMD Resistor 0806	154	154	154	154
TO263	18	18	18	18
Quartz HC-49/S	23	23	23	23
PCI connector	39	39	38	38

Implementation and experiments

The dataset of each component includes component-images and non-component images and is divided in data subsets. The first subset is for a priori knowledge generation (21%) and is not used for training or testing classifiers, to avoid classifier overfitting. The data subset for feature extraction and feature selection (49%) is the biggest dataset and is not used for testing the classifier to avoid classifier overfitting. The subset for classifier testing (30%) is divided in a subset for creating a decision fusion system based on the outputs from the classifiers (15%) and evaluating the decision fusion model (15%). The dataset for classifier training is not used for decision model estimation or evaluation because the classifier model can tend to overfit the data samples. The splitting of the dataset database from components is shown in Figure 59.

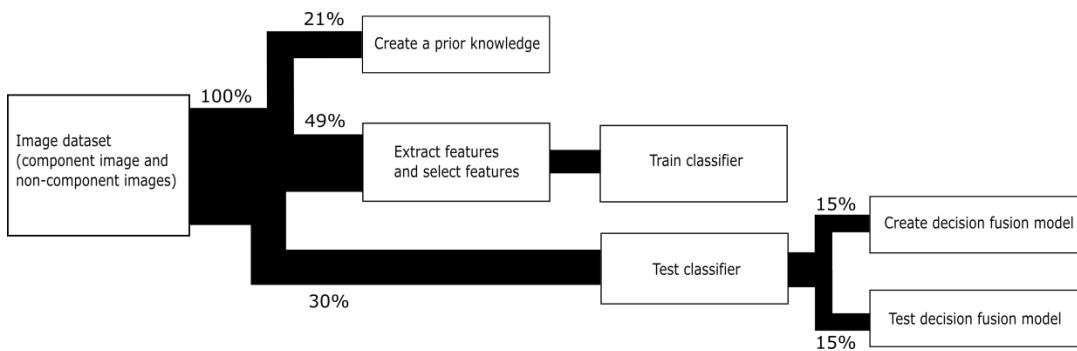


Figure 59: Component dataset splitting

It can be seen that a big component dataset is necessary to estimate and evaluate a stable electronic component recognition system.

5.2 PCB surface detection results

The process of PCB surface detection based on the PCB surface color is specified in chapter 3.2.2. The 54 PCB surface representations from the PCB surface detection test set were classified based on an RBF-SVM. The distance between the feature vector and the hyperplane of the RBF-SVM is a measure of goodness for PCB surface representation. The surface representation dataset was extracted from 110 images with different PCB surface colors (green, blue, red, yellow and others). The dataset consists of 5940 (110*54) surface representations of which 4653 were non-surface images and 1287 were surface images. The distinction between surface-image and non-surface image was determined manually and could not be

Implementation and experiments

determined clearly for each PCB surface representation. The confusion matrices for the training set and testing set are shown in Table 8 and Table 9.

Table 8: Confusion matrix of the predicted PCB surface training data

	Condition: surface image	Condition: non- surface image
Train outcome: surface image	915/932 (98.2%)	284/3523 (8.1%)
Train outcome: non- surface image	17/932 (1.8%)	3239/3523 (91.9%)

Table 9: Confusion matrix of the predicted PCB surface test data

	Condition: surface image	Condition: non- surface image
Test outcome: surface image	323/355 (91.9%)	107/1130 (9.5%)
Test outcome: non- surface image	32/355 (8.1%)	1023/1130 (90.5%)

The weighted sum of scores of the image Figure 60 is shown in Figure 61. It can be seen that the PCB surface pixel have much higher score values than others.

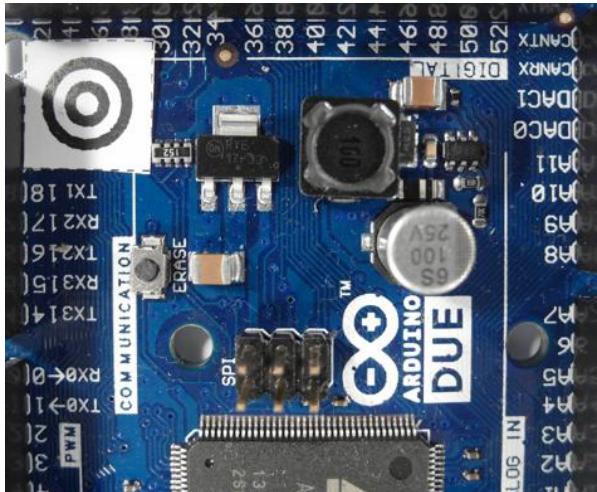


Figure 60: original PCB image

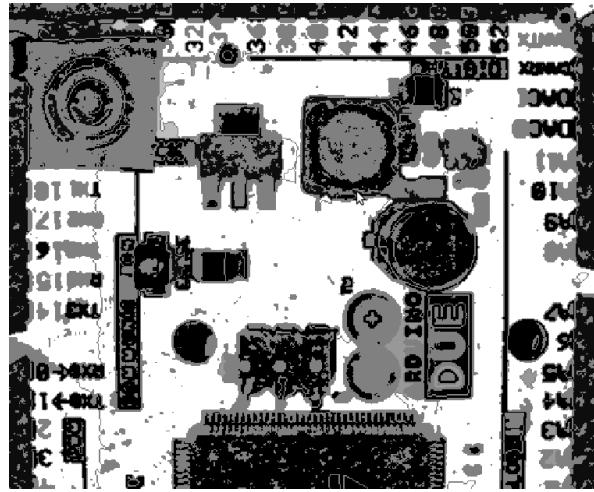


Figure 61: Sum of RBF-kernel SVM scores $w(x,y)$ (grayvalues are scaled between -20 and 20)

Implementation and experiments

A detailed analysis of the segmentation performance for the PCB surface recognition algorithm was not carried out due to the complexity of the PCB surfaces.

5.3 Feature selection results

The out-of-bag error depends on the number of random forest decision trees. The OOB-error depending on the number of decision 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.

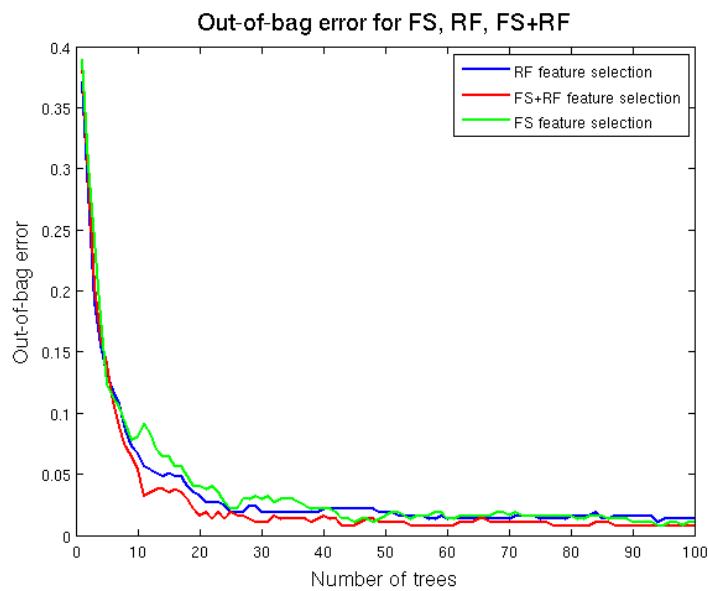


Figure 62: A comparison of different feature selection approaches

In this approach the feature selection algorithm based on Fisher score and Random forest described in chapter 3.4 was used to select a subset of important features for classification. The most important features depend on the component and therefore feature selection was

Implementation and experiments

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

5.3.1 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 63.

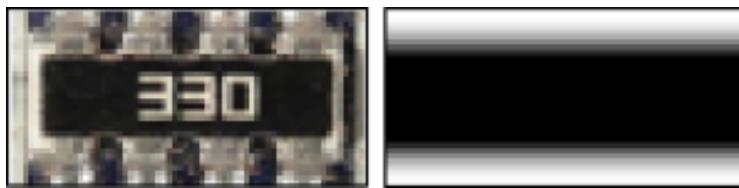


Figure 63: Resistor network 1206 and the most significant real part elementary image

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

5.3.2 Color features

The most important feature of the tantalum capacitor is a color feature which seems highly probably because 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. The tantalum capacitor and the normalized color histogram in the HSV color space is shown in Figure 64. The first two most important color features are marked in red.

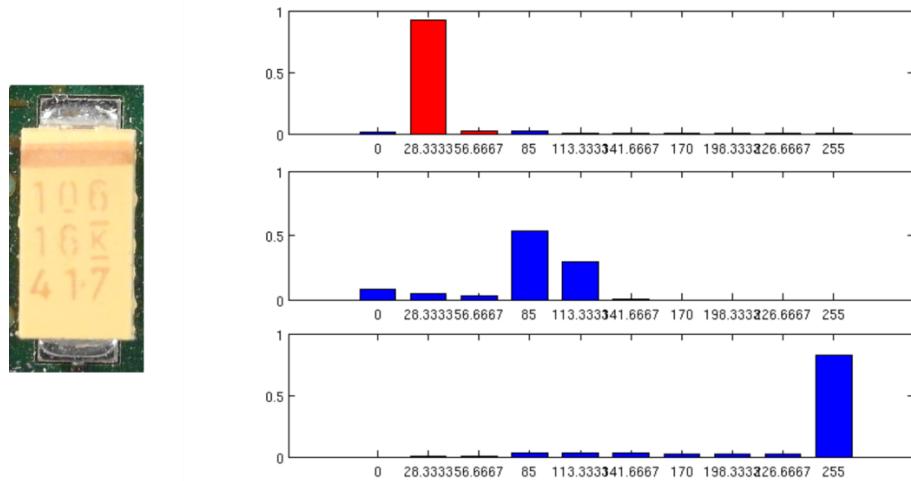


Figure 64: Tantalum capacitor and the most important histogram color features (HSV color space)

5.3.3 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 $y = 1.70 \text{ mm}$, $x = 0.26 \text{ mm}$. 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.

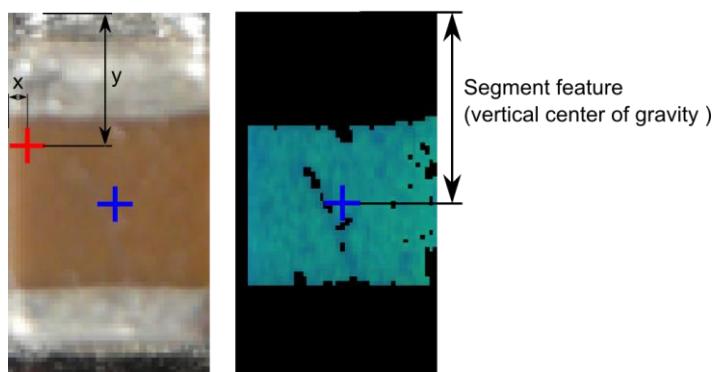


Figure 65: Most important segment and seed point from ceramic capacitor

5.3.4 PCA reconstruction feature

The most important feature of the SMD Aluminum electrolytic capacitor is the PCA-reconstruction feature. That can be specified by looking at the circular border of the cylindrical component . 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 component 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 66.

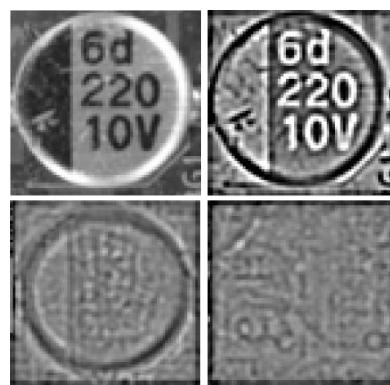


Figure 66: SMD Electrolyte capacitor (top, left), SMD Electrolyte capacitor edge image (top, right), unit matrix projection into component PCs (bottom, left), unit matrix projection into non-component PCs (bottom, right)

5.4 Classification results

The One-vs.-rest classification strategy is based on the approach that for each component a classifier is trained and tested. In this approach the random forest classifier, linear support vector machine, and support vector machine with RBF-kernel were trained and tested. The results are shown in the next chapters.

5.4.1 Random forest classifier results

Five random forest classifiers were trained whereas the first four are based on the four selected feature sets from the frequency feature domain, color feature domain, segment feature domain and PCA-reconstruction feature domain which were extracted from the four feature

Implementation and experiments

extraction algorithms specified in chapter 2.1. The average accuracy over all fifteen components is shown in Table 10. A detailed breakdown can be found in Appendix B.

Table 10: 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	1911/1971 (97.0%)	1893/1971 (96.0%)	1595/1682 (94.9%)	1559/1659 (93.9%)	1958/1971 (99.3%)
	True negative	1915/1982 (96.6%)	1760/1982 (88.8%)	1422/1694 (83.9%)	1641/1694 (96.8%)	1955/1982 (98.6%)

The random forest classifier result is based on the number of trees, whereas the accuracy rate converges if the number of trees increases. The dependency between the number of trees and the misclassification rate for the resistor network based on the most important features from all feature domains is shown in Figure 67. It can be see that the minimum of the misclassification rate is already reached with around 20 decision trees.

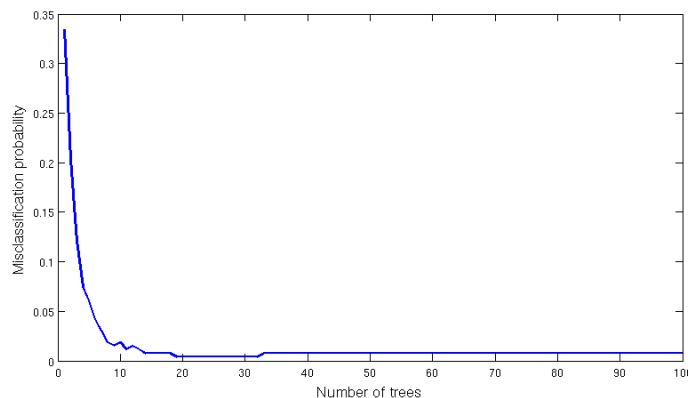


Figure 67: Dependence of the misclassification rate based on the number of trees (Resistor network, most important features from all feature domains)

Another important influence of the classification rate is the number of features. The dependency between the true positive rate and the 20 most important features and the dependency between the false positive rate and the 20 most features of the DIP14 component

Implementation and experiments

classifier are shown in Figure 68. It shows that already a small number of features can generate good classification results.

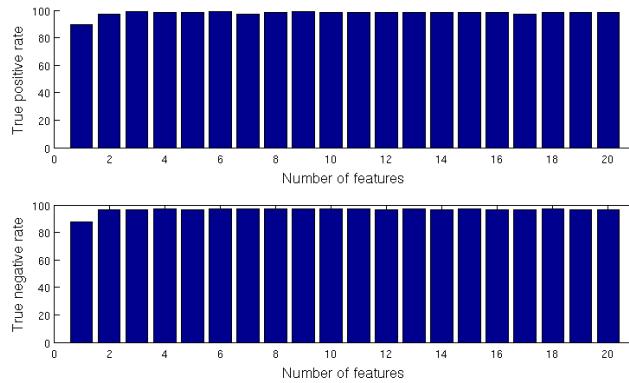


Figure 68: Dependency between the true positive and false positive rate from the number of features for the DIP14 component classifier and random forest classifier

5.4.2 Support vector machine classifier results

A linear support vector machine (Linea-SVM) was tested for component classification. The regularization constant C was estimated by the grid search method with exponential growing specified in 2.3.2. The search grid was determined by $C = \{2^{-7}, 2^{-6}, \dots, 2^{11}, 2^{12}\}$, the dependency between the constant and the error rate determined by cross-validation of the Resistor 0806 component is shown in Figure 69. It can be seen that the influence of the regularization constant C is small because the accuracy is very high and the soft margin constant C is regularization constant for false classified samples. The average accuracy over all fifteen components is shown in Table 11. A detailed breakdown can be found in Appendix C.

Implementation and experiments

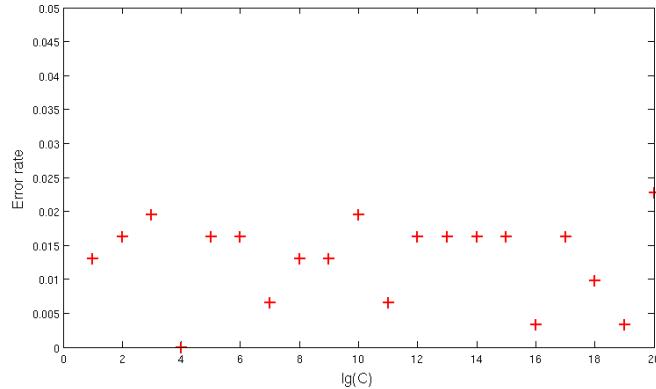


Figure 69: Dependency of the error rate from the regularization constant C (Resistor 0806)

Table 11: Linear-SVM 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	1837/1971 (93.2%)	1858/1971 (94.3%)	1569/1663 (94.3%)	1564/1666 (93.9%)	1908/1971 (96.8%)
	True negative	1871/1982 (94.4%)	1705/1988 (85.8%)	1419/1691 (83.9%)	1595/1687 (94.5%)	1909/1982 (96.3%)

A support vector machine with radial basis function was also tested. The parameter C (regularization constant) and γ (Gaussian kernel constant) were determined with the grid search method with exponential growing specified in chapter 2.3.2. The search grid was determined by $C = \{2^{-2}, 2^{-6}, \dots, 2^{11}, 2^7\}$ and $\gamma = \{2^{-6}, 2^{-5}, \dots, 2^6, 2^7\}$. The dependency between the constants and the error rate determined by cross-validation of the Resistor 0806 component is shown in Figure 70. It can be seen that the influence of the regularization constant C is very small compared to the RBF kernel parameter γ . The average accuracy over all fifteen components is shown in Table 12. A detailed breakdown can be found in Appendix D.

Implementation and experiments

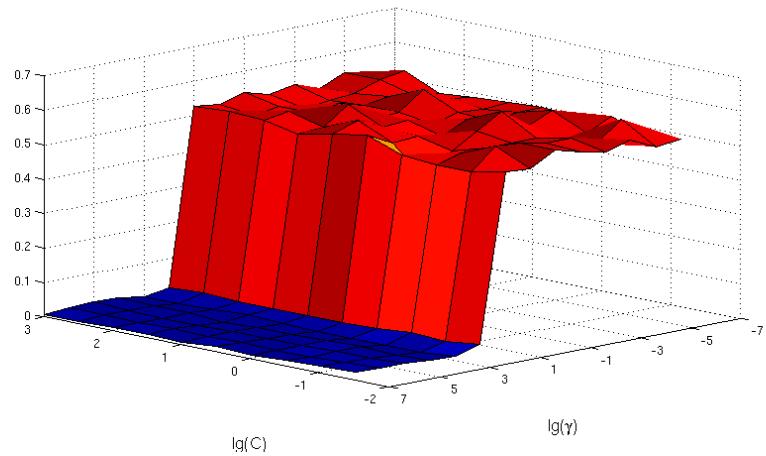


Figure 70: Dependency of the error rate from the regularization constant C and kernel constant γ (Resistor 0806)

Table 12: RBF-SVM 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	1909/1971 (96.9%)	1918/1971 (97.3%)	1565/1656 (94.5%)	1560/1656 (94.2%)	1953/1971 (99.1%)
	True negative	1956/1983 (98.6%)	1871/2382 (78.5%)	1551/1695 (91.5%)	1589/1695 (93.7%)	1967/1982 (99.2%)

5.5 Decision level fusion results with Dempster-Shafer theory

The experimental results of the decision fusion where made for 15 classes (14 component classes and one undefined class) of electronic components. All components from the recognition database components where used to test the decision fusion level except the PCI-slot component. The PCI-slot component needs a wide field of view because of the length of the PCI-slot. With this camera configuration it is not possible to make images with a large field of the size from the PCI-slot and a sufficient resolution for the Resistor 0806. Therefore the PCI-slot is out of focus for testing the decision fusion level. For testing the decision fusion level, all

Implementation and experiments

test images from the component database where used to determine the accuracy for the multi class classification process.

The One-vs.-rest strategy was used to evaluate the decision level fusion step. A single classifier per class was trained with samples from this class as positive samples and randomly selected image regions as negative samples. The outputs are confidence scores between zero and one, according to the classifier fusion level outputs. The component class with the maximum belief according to (69) is assigned as component class. The resulting confusion matrix of the components is shown in Appendix E. The accuracy rate is 95.0% based on 15 component classes.

5.6 Optical character recognition results

To evaluate the optical character recognition results different recognition levels are defined in chapter 3.7.3. The OCR experimental results are based on components specified in the OCR dataset limits in chapter 5.6.1.

5.6.1 Optical character recognition dataset and limits

The optical character recognition dataset consists of 85 ICs which were acquired with an image resolution of *60 pixel/mm*. All components were manually labeled according to the accuracy level scheme in 3.7.2.

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

- 1) 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).
- 2) Marking characters have a minimum height of 1.0 mm
- 3) Makings made by laser engraving are out of focus
- 4) The IC markings have to be readable by humans

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

5.6.2 Optical character recognition accuracy results on character level, word level, label level and part level

To evaluate the optical character recognition results, the manually labeled component markings and the recognized markings with OCR software are compared on different accuracy levels (Heliński, et al., 2000).

The character level accuracy of the OCR engine recognition is calculated as follows:

$$A_{c,i} = 1 - \frac{e_{c,i}}{c_i} \quad (89)$$

were $e_{c,i}$ is the number of character errors (insertions, substitutions and deletions) of the component marking i and c_i is the number of all characters of the marking i . The average character level accuracy over all 85 component markings is calculated as follows:

$$A_c = 1 - \frac{\sum_{i=1}^{85} e_{c,i}}{\sum_{i=1}^{85} c_i} \quad (90)$$

The word level accuracy of the OCR engine recognition is calculated as follows:

$$A_{w,i} = 1 - \frac{e_{w,i}}{w_i} \quad (91)$$

were $e_{w,i}$ is the number of word errors of component marking i and w_i is the number of words of component marking i . The average word level accuracy over all 85 component markings is calculated as follows:

$$A_w = 1 - \frac{\sum_{i=1}^{85} e_{w,i}}{\sum_{i=1}^{85} w_i} \quad (92)$$

The accuracy results of the OCR engines Tesseract and OCRMax on all accuracy levels is shown in Table 13.

Table 13: OCR accuracy results

	Tesseract	OCRMax
Character level accuracy A_c	1352/1704 (79.3%)	1342/1704 (78.8%)
Word level accuracy A_w	123/234 (52.6%)	126/234 (53.9%)

Implementation and experiments

The label level accuracy was not studied because of the high number of non-component labels with many characters which would result in a high error rate and is not representative caused by the fact that labels have to be filtered based on a component-name database. An investigation of the accuracy on label level with the Octopart database is done in chapter 5.6.3.

5.6.3 Octopart based component name assignment

The online electronic component database Octopart gives the opportunity to verify recognized component markings. The OCR dataset was used to test the assignment of recognized markings to components in the Octopart database. Therefore the labeled markings were decomposed in words (word-level) and the words were composed to labels (label-level). The words and labels were requested with the Octopart-API at the Octopart database and the results were analyzed according to the method in chapter 3.7.3. One of the classes “component-name” and “non-component-name” is assigned to each of the words and labels. The analyzed results were evaluated according to the assignment. The words/labels that are component-names and which are assigned to the right component in the Octopart database are true positive labeled results. Words/labels that are non-component names like manufacturer names, country of manufacture, production numbers etc. and which are not assigned to components in the Octopart database are labeled as true negative. Words/labels that are not component names but assigned to components in the Octopart database are labeled as false positive. Words/labels that are components but are not assigned to components in the Octopart database or assigned to wrong components in the database are labeled as false negative. The confusion matrix for the manual labeled words is shown in Table 14.

Table 14: Confusion matrix of the manual labeled words (word-level) verified with Octopart database

	Condition: component name	Condition: non-component name
Test outcome: component name	60/73 (82.2%)	6/161 (3.7%)
Test outcome: non-component name	13/73 (17.8%)	155/161 (96.3%)

Implementation and experiments

The confusion matrix for the manual labeled labels is shown in Table 15.

Table 15: Confusion matrix of the manual labeled labels (label-level) verified with Octopart database

	Condition: component name	Condition: non-component name
Test outcome: component name	61/75 (82.2%)	6/395 (1.5%)
Test outcome: non-component name	14/75 (17.8%)	389/395 (98.5%)

The accuracy rate on component-level in Table 16 shows how many components were assigned to a component in the Octopart database whereas the component names were manual labeled and verified with Octopart database on word level.

Table 16: Accuracy rate of component assignment with manual labeled components on word level verified with Octopart database (part-level)

Part assignment true (TPA)	59/85 (69.4%)
Part assignment false (FPA)	26/85 (30.6%)

The accuracy rate on part-level in Table 17 shows how many components were assigned to a component in the Octopart database whereas the component names were manual labeled and verified with Octopart database on label level.

Table 17: Accuracy rate of part assignment with manual labeled parts on label level verified with Octopart database (part-level)

Part assignment true (TPA)	60/85 (70.6%)
Part assignment false FPA	25/85 (29.4%)

The results in Table 16 and Table 17 show that the analyses of the component names on label level increases the accuracy rate on part level compared to the analyses on word level.

Implementation and experiments

The confusion matrices for the recognized component markings with the OCR engine Tesseract on word-level is shown in Table 17.

Table 18: Confusion matrix of the Tesseract recognized words (word-level) verified with Octopart database

	Condition: component name	Condition: non-component name
Test outcome: component name	31/73 (42.2%)	9/161 (5.6%)
Test outcome: non-component name	42/73 (57.8%)	152/161 (94.4%)

The confusion matrices for the recognized part markings with the OCR engine Tesseract on label-level is shown in Table 15.

Table 19: Confusion matrix of the Tesseract recognized labels (label-level) verified with Octopart database

	Condition: component name	Condition: non-component name
Test outcome: component name	33/75 (44.0%)	8/473 (1.7%)
Test outcome: non-component name	42/75 (56.0%)	465/473 (98.3%)

The accuracy rate on part level is shown in Table 16.

Table 20: Accuracy rate of part assignment with Tesseract OCR engine on word level verified with Octopart database (part-level)

Part assignment true	30/85 (35.3%)
Part assignment false	55/85 (64.7%)

Implementation and experiments

Table 21: Accuracy rate of part assignment with Tesseract OCR engine on label level verified with Octopart database (part-level)

Part assignment true (TPA)	31/85 (36.4%)
Part assignment false (FPA)	55/85 (63.6%)

The confusion matrices for the recognized component markings with the OCR engine OCRMax on word-level is shown in Table 22.

Table 22: Confusion matrix of the OCRMax recognized words (word-level) verified with Octopart database

	Condition: component name	Condition: non-component name
Test outcome: component name	44/73 (60.3%)	13/161 (8.1%)
Test outcome: non-component name	29/73 (39.7%)	148/161 (91.9%)

The confusion matrices for the recognized part markings with the OCR engine OCRMax on label-level is shown in Table 22.

Table 23: Confusion matrix of the OCRMax recognized labels (label-level) verified with Octopart database

	Condition: component name	Condition: non-component name
Test outcome: component name	44/75 (58.7%)	9/473 (1.9%)
Test outcome: non-component name	29/75 (41.3%)	464/473 (98.1%)

The accuracy rate on part level with OCR engine OCRMax is shown in Table 24.

Table 24: Accuracy rate of part assignment with OCRMax OCR engine on word level verified with Octopart database (part-level)

Part assignment true	44/85 (52.0%)
Part assignment false	41/85 (48.0%)

Implementation and experiments

Table 25: Accuracy rate of part assignment with OCRMax OCR engine on label level verified with Octopart database (part-level)

Part assignment true (TPA)	44/85 (52.0%)
Part assignment false (FPA)	41/85 (48.0%)

5.6.4 Octopart based part price assignment

To evaluate the economic sustainability of reuse of electronic components it is necessary to estimate the economic value of recognized parts. One indicator of valuable components is the original price of the component. The Octopart database gives the possibility to request the price for a components if the components could be assigned to a components in the database. Unfortunately not all suppliers publish their prices and therefore a price can just be assigned for a subset of the parts. The prices of all manual labeled parts were requested, and the price rate was calculated as follows:

$$A_{price} = \frac{\#parts_{price}}{\#parts_{assigned}} = \frac{30}{59} = 0.509 \text{ (50.9\%)} \quad (93)$$

where $\#parts_{price}$ is the number of components where a price could be estimated and $\#parts_{assigned}$ is the number of components that could be assigned to a component in the Octopart database. The price rate shows that for around 51% a component price could be estimated with the Octopart database.

To estimate the reuse potential of electronic components a critical economic value for the components can be estimated which represents the balance between the costs of desoldering the component and the costs of quality check one hand and the economic value of a component on the other hand. The AutDem project (Automated disassembly of PWBS) estimates the cost for automated desoldering between 1.2 and 2.5 Euro depending on desoldering time, line configuration and utilization (Griese, et al., 2002).

The maximum value of 2.30 Euro was used to estimate the critical price rate which was calculated as follows:

$$A_{price,critical} = \frac{\#parts_{price,critical}}{\#parts_{assigned}} = \frac{10}{59} = 0.17 \text{ (17.0\%)} \quad (94)$$

where $\#parts_{price,critical}$ is the number of components where a price could be estimated and the price was greater than 2.30 Euro and $\#parts_{assigned}$ is the number of components that could be assigned to a component in the Octopart database. The critical price rate shows that for around 17% of the assigned components, a price could be estimated which is greater than the critical price of 2.30 Euro based on the Octopart database. A detailed discussion about the reuse potential is made in chapter 7.3.

5.7 Life-cycle inventory analyses evaluation and results

The results of the two models are different in a way that the estimated PCB-composition model quantifies the materials which make up the PCB. Components 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 is therefore an important factor for an economic recycling process.

The PCB-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 components containing hazard materials that can be specially treated.

5.7.1 GaBi-Software and LCI data availability of electronic components

The ILCD component models in this work are exported from the GaBi Extension database XI: Electronics from PE INTERNATIONAL which consists of around 180 electronic components. Alternative electronic component databases can also be used if ILCD models can be exported. New ILCD models of components for the composition PCB model can be created based on measurements of component composition.

The ILCD-model selection for an electronic component is an important step to create a realistic LCI- and composition model. The ecological impacts as well as the material composition of an electronic component can change strongly if the same component package with a different component design is selected. An example is the SMD resistor in the 1206 package. The composition model of the “Resistor thick film flat chip 1206 (8.9mg)” from the GaBi database

Implementation and experiments

consist of a considerable amount of palladium whereas the “Resistor flat chip 1206 (9.2mg)” does not contain any palladium. Unfortunately the optical inspection system with only a 2D image sensor cannot distinguish between a resistor containing palladium or not. If electronic components without a comparable ILCD models in a database are replaced by an ILCD replacement model, it must be ensured that the replacement model does not overestimate the content of special materials. Especially if the likelihood of occurrence from the components in the PCB waste is huge, the ILCD model has to be determined carefully.

The material composition model and the LCI-model for the Arduino Due board are specified in chapter 5.7.3 as an exemplary PCB model.

5.7.2 Tantalum as an example for concentration increasing by selective dismantling

Tantalum is one of the materials which production increases every year. Around 1400 tons of tantalum is produced worldwide per year. Around 60% of the tantalum is used in capacitors for electronic equipment like Desktop PCs, Mobile phones or others (Chancerel, et al., 2013).

The concentration of tantalum in electronic scrap is low and the present economic value is not very high compared to other metals like gold or palladium which makes challenging to recycle tantalum from electron scrap. In the present recycling process is focused on the recycling of precious metals caused by the fact that the economic value is much higher compared to other materials.

The concentration of tantalum in tantalum capacitor scrap is between 35% and 50% which makes it economically attractive to recycle tantalum capacitors (Chancerel, et al., 2013). The approach of automatic optical inspection (AOI) for tantalum capacitor localization on PCBs and the automatically selective disassembly of the tantalum capacitors can increase the recycling rate and prevent from a worldwide lack of tantalum caused by higher production rates. A market for tantalum capacitor scrap already exists (Tantalumrecycling, 2015).

5.7.3 Arduino Due board LCI-model

The Arduino Due is a microcontroller board based on the Atmel SAM3X8E ARM Cortex-M3 CPU (Arduino, 2014). The Arduino board consists of an open-source hardware design and was used as a reference board in the INPIKO-Project. The Arduino Due board was used as LCI- and composition model example reference due to the fact that an open-source eagle layout is available and a component list can be easily exported in the eagle software.

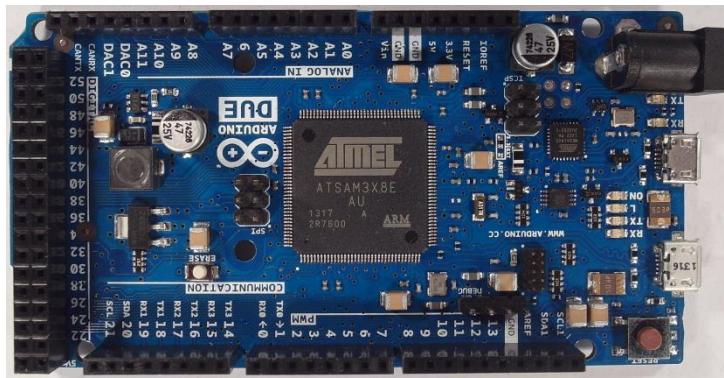


Figure 71: Arduino Due board

The Arduino Due board consists of 125 components from 32 different component classes. The LCI-model (Life-cycle-inventory-model) was created based on the assumption that all components are correctly detected, classified and all IC markings were correctly recognized. All components are correctly assigned to the component in the Octopart database.

Each component of the Arduino Due board was modeled by an ILCD-model whereas ILCD models were exported from the GaBi Extension database XI: Electronics from PE INTERNATIONAL and scaled according to the component size. Electronic components that could not be assigned with an associated component from the GaBi database were replaced with a replacement model. 16 of the 32 components of the Arduino Due board could be assigned to a model in the Gabi database and 17 of the 32 components had to be replaced by replacement models. The replacement models are also electronic components from the GaBi Extension database XI which were selected according to a similar structure and similar characteristics as the component. The assignments and the replacement models are listed in Appendix G.

Implementation and experiments

The resulting process model input components for the Arduino Due model are shown in Table 26.

Table 26: Arduino Due parts of the LCI model

Input	Amount
Kondensator Keramik MLCC 0603 (6mg) 1.6x0.8x0.8 PE	10
Kondensator Keramik MLCC 01005 (0,054 mg) 0,4x0,2x0,22	33
Transistor signal SOT23 3 leads (10mg) 1.4x3x1	8
Diode MELF (130mg) D2.6x5.2	2
Diode power DO214_219 (93mg) 4.3x3.6x2.3	1
Schalter Tact (242mg) 6.2x6.3x1.8	2
Spule Multilayer Chip 1812 (108mg) 4.5x3.2x1.5	2
IC TSSOP 8 (28mg) 3.0x2.9x1.2 flash	2
Transistor signal SOT223 3 leads (110mg) 3.8x7.65x2.3	1
IC TQFP 32 (70mg) 5x5x1.0	1
Widerstand Dickfilm Flat Chip 0402 (0.75mg)	44
LED SMD low-efficiency max 50mA (35mg) without Au 3.2x2.8x1.9	6
Spule Miniatur gewickelt SDR1006 (1.16g) D9.8x5.8	1
Kondensator Al-Elko SMD (300mg) D6.3x5.4	2
Widerstand Dickfilm Flat Chip 0603 (2.1mg)	8
Widerstand Dickfilm Flat Chip 1206 (8.9mg)	16
IC TQFP 100 (520mg) 14x14x1.0	1
Quartz Crystal (500mg) 11.05x4.65x2.5	3
Stecker, für Netzwerkkabel, ab Werk	2
Lotpaste SnAg	0.003 kg
Leiterplatte 2-Lagen starr FR4	0.0028 kg

The electronic components consist of materials which can be recycled under certain circumstances. The estimated material composition of the Arduino Due board is shown in Figure 72. The largest material portion of the components is the copper with around 18% of the weight, followed by epoxy resin with 17% and tin in alloys and glass fibre with 16%.

Implementation and experiments

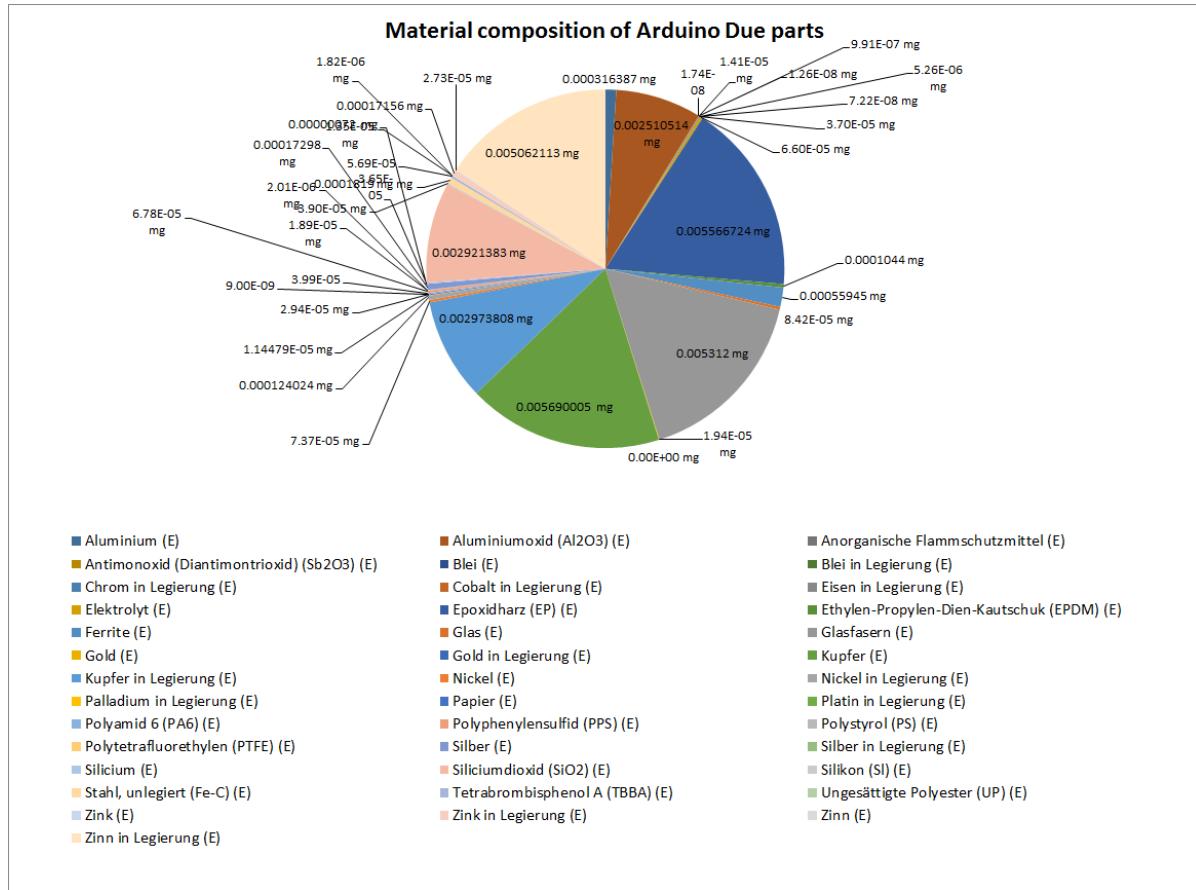


Figure 72: Estimated material composition of Arduino Due components

For recyclers the amount of material in an electronic component is not the main interest. It is a combination of material amount, price and recyclability of materials. The estimated materials with the highest prices multiplied by the material amount of in the Arduino Due PCB are shown in Figure 73. The prices of the materials are estimated by the average material price on the world markets in the past few years. A detailed composition of the determined material prices is shown in Appendix I. The added up material price of the whole Arduino Due PCB is around 1.63\$ whereas the price for separating this materials from the PCB is not included. It seems interesting that 67% of the material price comes from the gold contend (0.98\$) in the PCB which makes up 0.065% of the material weight. The second most valuable material in the PCB is Silver with 13% (0.20\$) of the material price. The third most valuable material is palladium with 0.099\$. They all are precious metals and recycling of these materials is physically possible.

Implementation and experiments

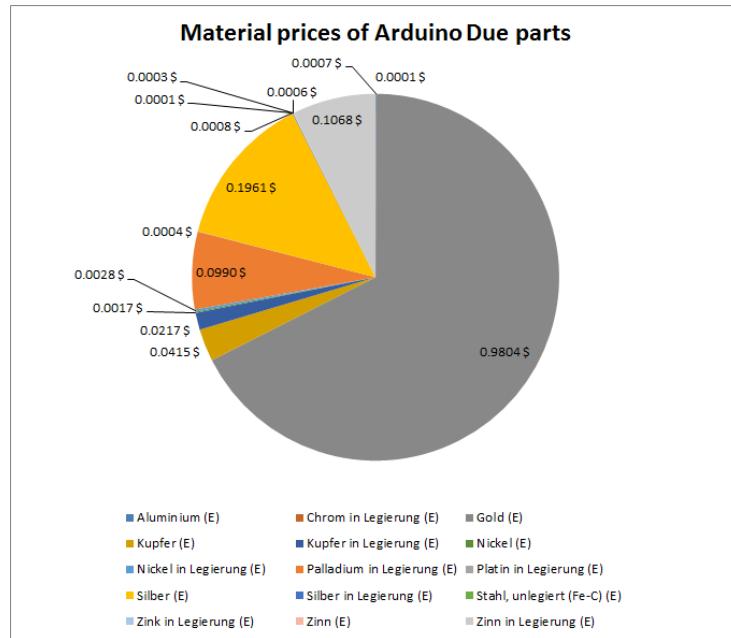


Figure 73: Estimated material prices of Arduino Due components

One of the most valuable materials is gold which is included in components or used as protective coating on electric connectors. The estimated amount of gold distributed over the components of the Arduino Due board is shown Figure 74. The amount of gold in the Arduino Due board was added up to 19.49mg, whereas just four different components (Ceramic Capacitors MLCC 0603, IC TSSOP 8, IC TQFP 32 and IC TQFP 100) contain the whole gold. With the possibility to identify and remove these components for gold recycling the concentration of gold can increase from 0.065% in the PCB to 2.76% gold in the removed electronic components. The increase of precious metal concentration improves the recycling rate and reduces recycling costs.

Implementation and experiments

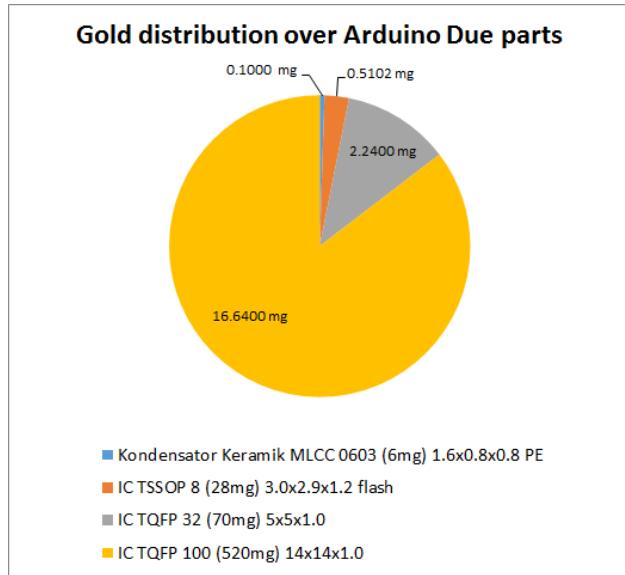


Figure 74: Estimated Gold distribution over Arduino Due parts

Another valuable and recyclable precious metal is palladium. The estimated palladium distribution over the components of the Arduino Due is shown in Figure 75. The 4.19mg of palladium in the PCB are distributed over four different components, Kondensator Keramik MLCC 0603, IC TSSOP 8, Widerstand Dickfilm Flat Chip 0402 and Widerstand Dickfilm Flat Chip 0603. The estimated concentration of palladium in the PCB is about 0.013% which increases by removing just the palladium containing components to 2.52%.

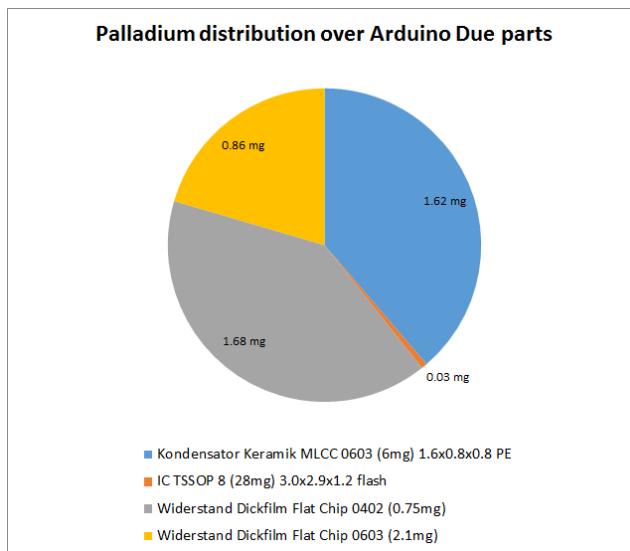


Figure 75: Estimated Palladium distribution over Arduino Due components

Implementation and experiments

The reuse of components is the most ecologically recycling way for electronic components. Electronic components can be unsoldered from PCBs and reused in other electronic applications. Due to the high price fluctuation between electronic components and the high cost of unsoldering and testing electronic components for reuse, the component prices are a strong indicator for reusability. The estimated prices of the Arduino Due component with a price greater than 0.1€ are shown in Figure 76, the prices from the rest of the electronic components are added up in the rest. The component prices are determined according to the reinstate value based on the distributors. The estimated component prices per unit for buying 1000 pieces are listed in Appendix H.

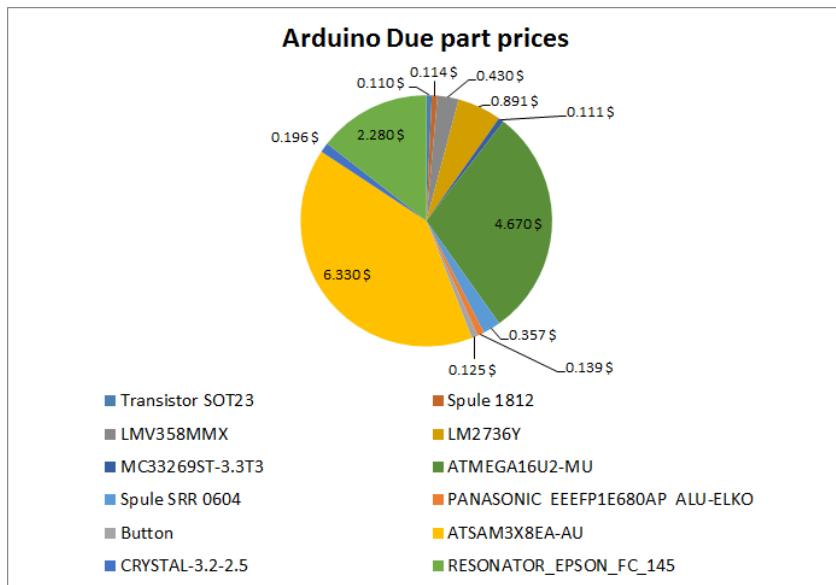


Figure 76: Estimated Arduino Due component prices

The most valuable component of the Arduino Due board is the Atmel ATSAM3X8EA-AU with a price of around 6.30\$. According to the estimated price of 2.30\$ for reusing an electronic component which was determined by the AutDem project (Automated disassembly of PWBs) three electronic components could be reused (Griese, et al., 2002).

6. Discussion and future work

The electronic component recognition process is based on the images acquired with the image acquisition setup specified in chapter 5.1.1. The detection and classification of electronic component packages and the optical character recognition process can be improved by selecting a telocentric objective to prevent optical barrel distortions. Industry cameras of higher image acquisition quality and an advanced lighting system can improve the recognition system.

The practical implementation of an electronic component recognition system for PCB recycling requires a fast and reliable recognition process. This can be archived by using an assembly line where removed PCBs from WEEE scrap are placed. The image acquisition can be done by line scan cameras. The images are analyzed to detect valuable components for reuse and recycling. The component removal process for reuse is done automatically or manually by non-destructive removal of the component. For recycling the electronic components can be destroyed by the removing process and can be done by milling, punching, heating and picking or alternative removal processes.

6.1 Electronic component detection

The electronic component detection is not specified in detail in this work. Alternative component detection approach based on 3D PCB models and laser triangulation are discussed further.

6.1.1 Electronic component detection based on 3D model

Several approaches for electronic component detection were examined in chapter 3.2 whereas a detailed evaluation was not applied. One further approach is the electronic component detection based on 3D PCB models. The Project “Integrierte Prozesskette für die Instandhaltung elektronischer Komponenten” (INPIKO) shows that a segmentation of electronic components based on 3D PCB models provides good results. In this approach a plane segmentation algorithm searches for the PCB surface and crops all voxels whose height is greater than the height of PCB support material. All voxels with a small Euclidian distance between each other (Euclidian cluster) are combined to an electronic component. The result for the electronic component segmentation of the Arduino Due board is shown in Figure 77. The segmented

Discussion and future work

components can be used to determine the centroid of the component bounding box and use the coordinates as inputs for the 2D component classification.

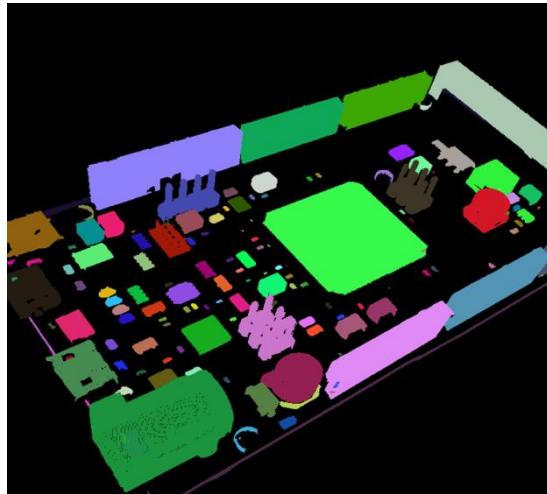


Figure 77: 3D model based component detection

6.1.2 Electronic component detection based on Height map from laser triangulation

A further approach for electronic component detection and segmentation is based on laser triangulation. The height map reconstruction from PCBs with laser triangulation was examined in (Koch, et al., 2013).

A laser triangulation is based on angle dependent projection displacement on surfaces. An optical sensor detects the position of a projected laser spot. The laser line is orthogonal to the direction in which the PCB is moved through the system. The principle of laser triangulation is shown in Figure 78. The line projected by the laser generates an angle dependent displacement $\lambda(x)$ at different heights at every position x of the slide, which is detected by the sensor. With the displacement λ and the known angle γ the height $h(x)$ can be calculated by

$$h(x) = \frac{\lambda(x)}{\tan(\gamma)} \quad (95)$$

Discussion and future work

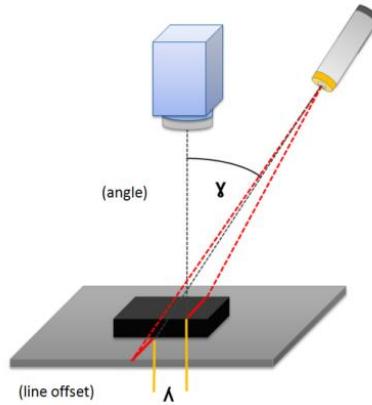


Figure 78: Principle of laser triangulation (Torsten Koch, 2013)

Advanced line detection algorithms and post processing, of the height map, leads to an average height error below 1 mm (Koch, et al., 2013), which is sufficient for reasonable electronic component detection and segmentation.

7. Conclusion

The goal of this thesis was to develop a system for PCB component recognition for material recycling. The system analyses PCB waste according to valuable electronic components for reuse or recycling. It is based on a 2D image sensor which acquires images from the PCB boards and its components. The advantage to use just 2D image sensors is, that cost-effective solutions for recyclers can be constructed. In this approach the consumer camera Samsung EX2F and four table lamps were used to acquire the images. A system with assembly line, modern industrial line-sensor cameras and a professional lighting system can improve the electronic component recognition results.

One of the main steps in electronic component recognition is the detection of components on the PCB. An approach based on template matching was analyzed and showed that a 2D cross-correlation with the average component image can determine potential component positions and decrease the search space of electronic components. The second component detection approach was based on the PCB surface recognition by color. The algorithm tries to segment the PCB surface by color to determine the PCB surface area where no component is located and therefore reduces the component search space. Alternatives based on 3D PCB models or laser triangulation were discussed. A detailed analysis of the detection rate was not specified and has to be investigated in further works.

After the detection of components a data fusion model was applied for electronic component classification. The one- vs. rest classification approach was used to determine the class of detected electronic components.

7.1 Data fusion model for electronic component recognition

A data fusion model was created which consists of three fusion levels (feature-level, classifier-level, decision-level). The feature level fusion is the first fusion-level which is based on the extracted features. The extracted features are extracted from four different feature domains.

The first feature domain is based on the frequency spectrum and contains much information about the existence and distances between solder joints (pitch). The intensity values of the

Conclusion

grayscale image are height at the positions of solder joints compared to the intensity values between the joints. This feature domain is particularly important for components with many equidistant solder joints (QFP100) or component with large solder joints (SOT223). The second feature domain is based on the color distribution of the component image. Especially components with significant colors (Tantalum capacitor, Quartz) can be classified according to these features. The third features domain is based on significant color segment. A region growing approach is used to find regions with equal color and uses the position, dimension and color as features. Components with significant color segments (Ceramic capacitor) can be well classified according to these features. The fourth feature domain is based on the edges in a component image. Components with significant edges at solder joints or light reflections at the rounded edges of electrolyte capacitors can be well classified according to this feature. Principal components (PCs) from a subset of edge images are determined to project the images along the PCs and reconstruct them. The reconstruction error of the edge image is used as a feature. The experimental results show that all feature domains are useful whereas the importance of individual features depends highly on the structure of the recognized component.

The features level fusion selects in each of the four feature domains the most important features. The feature selection is based on fisher score and random forest feature selection approach. Additionally the most important features from all four domains are selected to form a fifth feature set. The feature fusion was built modular, that new feature domains can be easily added to build a more stable recognition system with higher accuracy rates. It shows that feature selection works well and high classification rates can be archived with a small number of features.

The fusion on classifier-level is based on the selected features in the feature-level fusion process. Five classifiers were trained with the selected features from the four domains and additionally a classifier for the selected features from all four domains. The Random forest classifier, Linear support vector machine classifier (Linear-SVM) and support vector machines with radial basis function (RBF-SVM) where examined. The results in chapter 5.4 shows that all

Conclusion

three classifiers reached accuracy rates of more than 96% with the most important features from all feature domains. The random forest classifier seems to be the best classifier for the electronic component classification. He reached a true positive rate averaged over all 15 components of 99.3% and a true negative rate of 98.6%. The second best classifier is the support vector machine with RBF kernel and reaches a true positive rate of 99.1% and a true negative rate of 99.2%. The third best is the linear support vector machine with a true positive rate of 96.8% and a true negative rate of 96.3%. It shows that most of the data samples are linear separable so that also linear classifiers produce good results. It should be noted that the feature selection for all classifiers was done based on the random forest feature selection. Different feature selection approaches can lead to different results.

The last fusion level is the decision-level fusion step where the outputs from the classifier fusion level from all component classes are used to determine the class of a component. Additionally the component class can also be an unknown component class if the component is not in the recognition database and no classifier is trained for that component. A dempster-shafer fusion approach is used to combine the information from all classifiers. The test result shows that 95.0% of the components are classified correctly based on 14 component classes and one unknown component class. As compared to other electronic component recognition results, a fusion approach based on range image, color image and high resolution image reached 82% recognition rate with 19 component classes (Dop, 1999).

7.2 PCB material composition model estimation

To estimate the recycling potential of electronic components, the material composition was estimated based on a representative ILCD-composition model. Each component in the recognition database consists of an ILCD model which contains the material composition. If a component of the PCB board was classified as a component in the database, the corresponding ILCD model is added to the PCB model. The composition models were extracted from the GaBi Extension database XI: Electronics from PE INTERNATIONAL.

The Arduino Due board was modeled by the ILCD models and it shows that precious metals like gold, silver or palladium are distributed over a small number of components. If the components

Conclusion

are rejected, the material concentration greatly increases and can increase the recycling rate of the precious metals. Rare materials or special metals like tantalum, which are not recycled in today's recycling companies, because of their physical properties and the low economic material price become economically attractive for recyclers.

7.3 Electronic part name assignment for electronic part reuse

An optical character recognition approach was used to identify the electronic component names for potential reuse. The OCR system was developed to handle integrated circuits (ICs) with white (bright) characters on black (dark) background, with a character height of minimum 1.0 mm. Therefore an OCR system with the OCR engines Cognex Vision Pro OCRMax and Tesseract were tested. The results show that both OCR engines have almost equal recognition rates on character level (80%).

On word level the component names were verified based on the electronic component database Octopart. With this component name verification, Tesseract reached a recognition rates on word level of 42% and OCRMax of 60%.

The true part assignment rate is the accuracy rate that a component can be assigned to a component in the Octopart database. The true part assignment rate (TAR) of the OCR engine Tesseract is 35% which is lower compared to the part assignment rate for the OCRMax engine which is 52%.

The proportion of the component which can be reused can be calculated based on the detection rate, classification rate and the part assignment rate (TPA). The detection rate could not be specified in this work, therefore a rate of 90.0% is defined which seems to be realistic based on the results from related works. The classification rate is based on the decision fusion outcome and was examined in chapter 5.5. A classification rate of 95.0% was estimated based on 14 component classes and one unknown component class. A part assignment rate (TPA) of 52% can be reached with the OCRMax engine. The resulting reusability rate can be estimated as follows:

$$R_{reuse} = P_{detected} * P_{classified} * P_{TAR} = 0.9 * 0.95 * 0.52 = 44.5 \% \quad (96)$$

Conclusion

It shows that around 45% of the IC components on a PCB board can be reused.

An important information, for potential recyclers is the economic value of reusable electronic components, because the component price of many components is very low, compared to the costs for resold and test an electronic component. Therefore the reusability-valuable rate $R_{reuse,valueable}$ was estimated. It is the rate of electronic components which can be reused and have a reinstatement value of minimum 2.30 €. The critical price rate $R_{price,critical} = 0.17$ was estimated in chapter 5.6.4 and is the rate of a components which price is higher than 2.30€.

$$R_{reuse,valueable} = P_{detect} * P_{classifie} * P_{TAR} * P_{price,critical} = 0.9 * 0.95 * 0.52 * 0.17 \quad (97)$$
$$= 0.076 = 7.6 \%$$

The reusability-valuable rate seems to be low but can be improved by using an advanced image acquisition system, improved OCR schema and different electronic component databases than the Octopart database. Alternatives or additional databases are the TME API (www.developers.tme.eu/en/) or Ciiva (www.ciiva.com) electronic component database.

Note that the reusability-valuable rate $R_{reuse,valueable}$ was estimated based on a small number of electronic components. It is based on the OCR rate from the OCRMax engine and the electronic part name verification with the Octopart database. The rate is based on electronic components with white characters on black background, primarily Integrated circuits (ICs).

7.4 Application inclusion in the PCB recycling process chain

The WEEE recycling chain in chapter 1.1 was improved with the inclusion of the electronic component recognition application for PCBs. The improved WEEE recycling chain is shown in Appendix J. The process chain consists of three steps, the collection of WEEE, the preprocessing and the recovery and disposal. Influenced by the component recognition system are manly the preprocessing step and the recovery and disposal step.

Conclusion

In the preprocessing step, which is seen in Figure 79 the manual dismantling has to be changed. In many todays recycling chains the manual dismantling of WEEE is done to remove hazard substances. In some recycling companies the whole electronic device is shredded including PCBs which are damaged or destroyed by this process. In the improved system the electronic devices which include PCBs must be opened to remove the PCB. The PCBs are placed on an assembly line and an automatic optical inspection system (AOI) based on an electronic component database determines height valuable components. The components are examined from two points of view.

The first is the reuse of the components which is profitable if the value of the component reaches a certain threshold and can be tested successfully. The electronic components are manually or automatically unsolded and depending on the capability tested. After successfully testing the components are prepared for reuse.

The second point of view is the recycling of valuable materials from the electronic components. Therefore the components are analyzed, based on an LCI model database, according to the contend of precious metals, hazard substances or rare materials. The components are automatically loosened from the PCB whereas a destruction of the component is acceptable and simplified the loosen process. The removed components are separated according to valuable or hazard substances which increase the concentration.

Conclusion

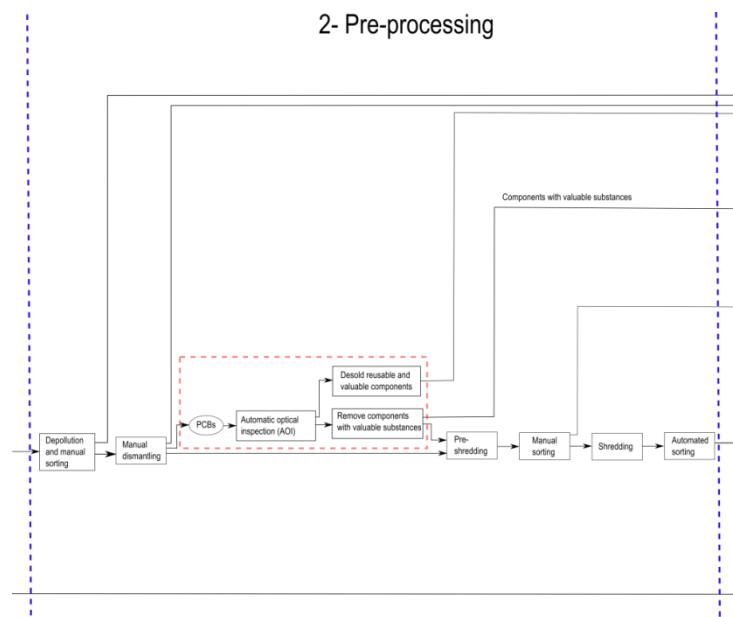


Figure 79: Improved pre-processing step in PCB recycling process chain

In the recovery and disposal step, the components which are tested successfully are prepared for reuse. The components are cleaned and packed in for sale. Unfortunately the market for reused electronic components from consumer electronics is still a very small market today. The B2C (Business-to-Consumer) environment for electronic components is a small market with a few shops for private hobbyist compared to the B2B (Business-to-Business) environment. Therefore the quality and reliability of reused electronic components has to be specified, which is challenging by a small price per unit and a large diversity of electronic components.

Components with valuable substances which were removed and collected according to their substances consist of a much higher concentration of special substances and decrease the recovery costs. The recycling of metals like tantalum which is mostly lost in today's recycling chains can profitably be recycled. The proportion of precious metal which ends up in the shredder in today's recycling processes can successfully be recovered. The process chain of the recovery and disposal step is shown in Figure 80.

Conclusion

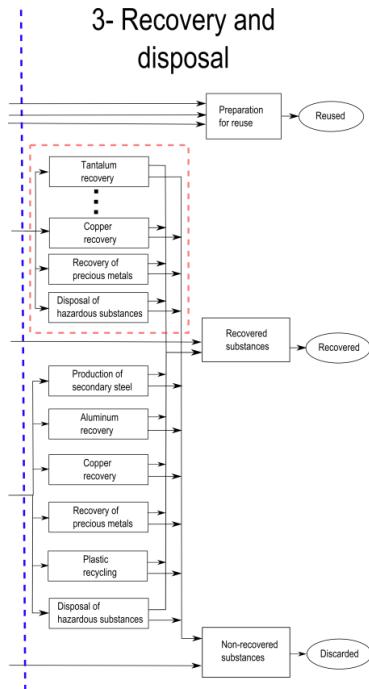


Figure 80: Improved recover and disposal step in PCB recycling process chain

The outlined recycling process chain is just a rough recycling model which has to be adjusted according to the goal of the recycling company. Recyclers which are only focused on the recycling of special metals can use the component detection application to detect specially these substances.

Bibliography

Bibliography

(SAIC) Scientific Applications International Corporation [et al.] Life-cycle Assessment: Principles and Practice [Book]. - [s.l.] : National Risk Management Research Laboratory, Office of Research and Development, U.S. Environmental Protection Agency, 2006.

Arduino ArduinoBoardDue: arduino.cc [Online] // arduino.cc. - 2014. - 12 16, 2014. - <http://arduino.cc/en/pmwiki.php?n>Main/ArduinoBoardDue>.

Article Original {Algorithm of locating PCB components based on colour distribution of solder joints} [Journal]. - 2011. - pp. 601-614. - DOI: 10.1007/s00170-010-2850-9.

Bakst J.S. [et al.] Guidelines for Assessing the Quality of Life-cycle Inventory Analysis [Book]. - [s.l.] : United States Environmental Protection Agency, Solid Waste and Emergency Response, 1995.

Blaes Patrick and Young Chris {Mobile IC Package Recognition} [Journal]. - pp. 1-6.

Breiman Leo Random Forests [Online]. - 2014. - https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm.

Burges Christopher J. C. A tutorial on support vector machines for pattern recognition [Journal] // Data Mining and Knowledge Discovery. - 1998. - Vol. 2. - pp. 121-167.

Castanedo Federico {A review of data fusion techniques.} [Journal] // TheScientificWorldJournal. - #jan# 2013. - Vol. 2013. - p. 704504. - ISSN: 1537-744X DOI: 10.1155/2013/704504.

Chancerel Perrine [et al.] {Assessment of Precious Metal Flows During Preprocessing of Waste Electrical and Electronic Equipment} [Journal] // Journal of Industrial Ecology. - #oct# 2009. - 5 : Vol. 13. - pp. 791-810. - ISSN: 10881980 DOI: 10.1111/j.1530-9290.2009.00171.x.

Chancerel Perrine [et al.] {Data availability and the need for research to localize, quantify and recycle critical metals in information technology, telecommunication and consumer equipment.} [Journal] // Waste management \& research : the journal of the International Solid

Bibliography

Wastes and Public Cleansing Association, ISWA. - #oct# 2013. - 10 Suppl : Vol. 31. - pp. 3-16. - ISSN: 1399-3070 DOI: 10.1177/0734242X13499814.

Chen Yi-wei and Lin Chih-jen {Combining SVMs with Various Feature Selection Strategies} [Journal]. - 1. - pp. 1-10.

Cheng H D [et al.] {Color image segmentation: advances and prospects} [Journal]. - 2001. - September 2000 : Vol. 34. - ISBN: 1435797205.

Commission European and Ies Centre {The International Reference Life Cycle Data System (ILCD) Handbook (online version)} [Book]. - 2012. - ISBN: 9789279216404 DOI: 10.2788/85727.

Cui Jirang and Forssberg Eric {Mechanical recycling of waste electric and electronic equipment: a review} [Journal]. - 2003. - Vol. 99. - pp. 243-263. - DOI: 10.1016/S0304-3894(03)00061-X.

Cutler Adele {Random Forests Leo Breiman and Adele Cutler} [Journal]. - 2014. - pp. 1-24.

deCampos T E, Feris F S and Cesar-Jr R M Improved Face x Non-Face Discrimination Using Fourier Descriptors Through Feature Selection [Conference] // SIBGRAPI . - Gramado, Brazil : [s.n.], 2000 .

Directive 2002/96/EC Directive 2002/96/EC of the European Parliament and of the Council of 27 January 2003 on waste electrical and electronic equipment (WEEE) - Joint declaration of the European Parliament, the Council and the Commission relating to Article 9 // Directive 2002/96/EC of the European Parliament and of the Council of 27 January 2003 on waste electrical and electronic equipment (WEEE) - Joint declaration of the European Parliament, the Council and the Commission relating to Article 9. - 2002.

Dong Jiang [et al.] {Advances in Multi-Sensor Data Fusion: Algorithms and Applications} [Journal]. - 2009. - 1. - pp. 7771-7784. - DOI: 10.3390/s91007771.

Dop Eric van Multi-sensor object recognition: The case of electronics recycling [Journal]. - 1999.

Duda Richard O, Hart Peter E and Stork David G Pattern classification [Book]. - [s.l.] : John Wiley \& Sons, 2012.

Bibliography

Duda Richard O. Pattern Classification [Book]. - [s.l.] : John Wiley and Sons, Inc., 2001.

Gonzalez Rafael C. and Woods Richard E. Digital Image Processing (3rd Edition) [Book]. - Upper Saddle River, NJ, USA : Prentice-Hall, Inc., 2006. - ISBN: 013168728X.

Griese Hansjorg [et al.] {Quality Assured Disassembly of Electronic Components for Reuse} [Journal]. - 2002. - pp. 299-305. - ISBN: 078037214X.

Gu Quanquan, Li Zhenhui and Han Jiawei {Generalized Fisher Score for Feature Selection} [Journal].

Guyon Isabelle {An Introduction to Variable and Feature Selection} [Journal]. - 2003. - Vol. 3. - pp. 1157-1182.

Heliński Marcin, Kmiecik Mirosz and Parkoła Tomasz Report on the comparison of Tesseract and ABBYY FineReader OCR engines // Report on the comparison of Tesseract and ABBYY FineReader OCR engines. - Poznań : PCSS, 2000.

Hsu Chih-wei, Chang Chih-chung and Lin Chih-jen {A Practical Guide to Support Vector Classification} [Journal]. - 2010. - 1 : Vol. 1. - pp. 1-16.

Huisman Jaco {QWERTY and Eco-Efficiency analysis on cellular phone treatment in Sweden} [Journal]. - 2004. - April.

IPK Fraunhofer INSTANDHALTUNG ELEKTRONISCHER KOMPONENTEN // INSTANDHALTUNG ELEKTRONISCHER KOMPONENTEN. - 2013.

Journal Arpn {ARPN Journal of Science and Technology:: Understanding Color Models: A Review} [Journal]. - 2012. - 3 : Vol. 2. - pp. 265-275.

Kay Rakowsky Uwe {Fundamentals of the Dempster-Shafer theory and its applications to system safety and reliability modelling} [Journal]. - 2007.

Koch Torsten, Breier Matthias and Li Wei {Heightmap generation for printed circuit boards (PCB) using laser triangulation for pre-processing optimization in industrial recycling}

Bibliography

applications} [Journal] // IEEE International Conference on Industrial Informatics (INDIN). - 2013. - pp. 48-53. - ISBN: 9781479907526 ISSN: 19354576 DOI: 10.1109/INDIN.2013.6622856.

Kohlas Jurg, Monney Paul-andre and Functions Belief {Theory of Evidence - A Survey of its Mathematical Foundations , Applications and Computational Aspects Key Words 2 . The Theory of Evidence - Di ering Approaches and Interpretations} [Journal]. - 21. - pp. 1-29.

ksdensity, mathworks mathworks [Online] // mathworks.de/help/stats/ksdensity. - 2014. - 11 10, 2014. - <http://www.mathworks.de/help/stats/ksdensity.html>.

Leiton Leiton: leiton-tools-gewichtsberechnung [Online] // Leiton. - 2014. - 11 25, 2014. - <http://www.leiton.de/leiton-tools-gewichtsberechnung.html>.

Lewis J P (Industrial Light \& Magic) {Fast Normalized Cross-Correlation Template Matching by Cross-} [Journal] // Vision Interface. - 1995. - 1 : Vol. 1995. - pp. 1-7. - ISSN: 15258955 DOI: 10.1007/s00034-009-9130-7.

Li Wei [et al.] {T}ext recognition for information retrieval in images of printed circuit boards [Book]. - Piscataway, NJ : IEEE, 2014. - pp. 3487-3493. - 1 USB-Stick. - ISBN: 978-1-4799-4033-2.

Li Wei, Esders Bernhard and Breier Matthias {SMD segmentation for automated PCB recycling} [Journal] // IEEE International Conference on Industrial Informatics (INDIN). - 2013. - pp. 65-70. - ISBN: 9781479907526 ISSN: 19354576 DOI: 10.1109/INDIN.2013.6622859.

Luo Bing [et al.] {SMT Components Model Inspection Based on Characters Image Matching and Verification} [Journal] // 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing. - [s.l.] : ieee, #aug# 2013. - pp. 1438-1441. - ISBN: 978-0-7695-5046-6 DOI: 10.1109/GreenCom-iThings-CPSCom.2013.252.

Luo Zhimin {An Automatic Chip Character Checking System for Circuit Board Quality Control} [Journal]. - 2014. - pp. 3-6. - ISBN: 0780379063.

Bibliography

Malagón-Borja Luis and Fuentes Olac {Object detection using image reconstruction with PCA} [Journal] // Image and Vision Computing. - #jan# 2009. - 1-2 : Vol. 27. - pp. 2-9. - ISSN: 02628856 DOI: 10.1016/j.imavis.2007.03.004.

Moreno-seco Francisco {Comparison of classifier fusion methods for classification in pattern recognition tasks} [Journal]. - 2014.

octopart Octopart // Octopart. - 2014.

Otsu Nobuyuki {A} {T}hreshold {S}election {M}ethod from {G}ray-Level {H}istograms [Journal] // IEEE Transactions on Systems, Man and Cybernetics. - 1979. - 1 : Vol. 9. - pp. 62-66. - DOI: 10.1109/TSMC.1979.4310076.

Petrou Maria and Bosdogianni Panagiota Image Processing : The Fundamentals [Book]. - [s.l.] : Wiley, 1999.

Pruhs Levenshtein Distance // Levenshtein Distance. - 2015.

Puchheim Stemmer Imaging Das Handbuch der Bildverarbeitung [Book]. - [s.l.] : Stemmer Imaging, 2010. - ISBN: 9783000300615.

Qubo Cao and Meng Han {Parameter selection in SVM with RBF kernel function} [Journal].

Rakowsky Kay Uwe Fundamentals of the Dempster-Shafer theory and its application to system safety and reliability modelling [Journal]. - 2007.

Rokach Lior {Ensemble-based classifiers} [Journal] // Artificial Intelligence Review. - #nov# 2009. - 1-2 : Vol. 33. - pp. 1-39. - ISSN: 0269-2821 DOI: 10.1007/s10462-009-9124-7.

Ruta Dymitr and Gabrys Bogdan {An Overview of Classifier Fusion Methods} [Journal]. - 2000. - Vol. 7. - pp. 1-10.

Scheideanstalt Scheideanstalt // Scheideanstalt. - 2015.

Steinberg Alan N [et al.] {Revisions to the JDL Data Fusion Model} [Journal].

Bibliography

Support_vector_machine // Support_vector_machine. - 2015.

Tantalumrecycling Tantalumrecycling // Tantalumrecycling. - 2015.

Tarnawski W {Colour image segmentation algorithm in vectoral approach for automated optical inspection in electronics} [Journal]. - 2003. - 3 : Vol. 11. - pp. 197-202.

Tianshou L. A highlight processing technology for SMT solder joint gray image [Journal]. - 2012.

Torsten Koch Matthias Breier, Wei Li Heighmap generation for printes curcuit boards (PCB) using laser triangulation for pre-processing optimization in industrial recycling applications [Journal]. - 2013.

van Dop Erik R. and Regtien Paul P.L. Multi-sensor recognition of electronic components [Journal] // Machine Vision and Applications. - [s.l.] : Springer-Verlag, 2001. - 5 : Vol. 12. - pp. 213-222. - ISSN: 0932-8092 DOI: 10.1007/s001380050141.

Verma Om Prakash [et al.] {A Simple Single Seeded Region Growing Algorithm for Color Image Segmentation using Adaptive Thresholding} [Journal] // 2011 International Conference on Communication Systems and Network Technologies. - [s.l.] : ieee, #jun# 2011. - pp. 500-503. - ISBN: 978-1-4577-0543-4 DOI: 10.1109/CSNT.2011.107.

VisionPro COGNEX COGNEX [Online] // www.cognex.com. - 11 11, 2014. - <http://www.cognex.com/ocr-optical-character-recognition.aspx>.

Wikipedia Outline of object recognition --- Wikipedia{,} The Free Encyclopedia // Outline of object recognition --- Wikipedia{,} The Free Encyclopedia. - 2015. - [Online; accessed 3-March-2015].

Wikipedia Region growing --- Wikipedia{,} The Free Encyclopedia // Region growing --- Wikipedia{ } The Free Encyclopedia. - 2014. - [Online; accessed 3-March-2015].

Wikipedia-Levenshtein Levenshtein distance --- Wikipedia{,} The Free Encyclopedia // Levenshtein distance --- Wikipedia{,} The Free Encyclopedia. - 2015. - [Online; accessed 3-March-2015].

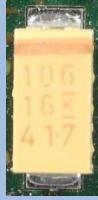
Bibliography

Wikipedia-RANSAC RANSAC --- Wikipedia{,} The Free Encyclopedia // RANSAC --- Wikipedia{,} The Free Encyclopedia. - 2015. - [Online; accessed 3-March-2015].

Wikipedia-SVM Support vector machine --- Wikipedia{,} The Free Encyclopedia // Support vector machine --- Wikipedia{,} The Free Encyclopedia. - 2015. - [Online; accessed 3-March-2015].

Wolf Marc-andree and Kusche Oliver {The International Reference Life Cycle Data System (ILCD) Format – Basic Concepts and Implementation of Life Cycle Impact Assessment (LCIA) Method Data Sets} [Journal]. - 2011. - Ilcd. - ISBN: 9783844004519.

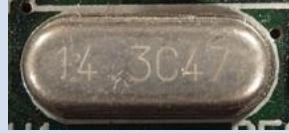
Appendix A Recognition database

Component name and description	Component image
<p style="text-align: center;">Tantalum capacitor</p> <ul style="list-style-type: none"> - Package: EIA Code 2412 - Color: yellow/orange - Tantalum capacitor with solid electrolyte polarity markings 	
<p style="text-align: center;">SMD Aluminum electrolytic capacitor</p> <ul style="list-style-type: none"> - Diameter: 6.5 mm 	
<p style="text-align: center;">QFP100</p> <ul style="list-style-type: none"> - Package: QFP100 - Dimension: 23.4 mm x 17.4 mm 	
<p style="text-align: center;">SMD Resistor Network array 1206, 4 Resistors</p> <ul style="list-style-type: none"> - Long Side Terminals - Four resistors 	
<p style="text-align: center;">SMD Transistor SOT23-3</p> <ul style="list-style-type: none"> - Package: SOT23-3 - Dimension: 3.0 mm x 2.6 mm 	
<p style="text-align: center;">DIP14</p> <ul style="list-style-type: none"> - Package: DIP14 - Dimension: 19.5 mm x 7.6 mm 	
<p style="text-align: center;">DIP16</p> <ul style="list-style-type: none"> - Package: DIP14 	

Appendices

<ul style="list-style-type: none">- Dimension: 19.5 mm x 7.6 mm	
<p>SMD Resistor 1206</p> <ul style="list-style-type: none">- Imperial code: 1206 (3216 metric)- Dimension: 3.2 mm x 1.6 mm	
<p>SOIC-8</p> <ul style="list-style-type: none">- Package: SOIC8- Dimension: 5.0 mm x 6.2 mm	
<p>Ceramic capacitor 1210</p> <ul style="list-style-type: none">- Imperial code: 1210 (3225 metric)- Dimension: 3.2 mm x 2.5 mm- Color: brown/orange	
<p>SOT223-3</p> <ul style="list-style-type: none">- Package: SOT223-3- Dimension: 6.5 mm x 7.0 mm	
<p>SMD Resistor 0806</p> <ul style="list-style-type: none">- Imperial code: 0806 (2012 metric)- Dimension: 2.0 mm x 1.2 mm	
<p>TO 263</p> <ul style="list-style-type: none">- Imperial code: 0806 (2012 metric)- Dimension: 10.1 mm x 15.0 mm	

Appendices

<p>Quartz HC-49/S</p> <ul style="list-style-type: none">- Package: HC-49/S-3- Dimension: 4.7 mm x 11.0 mm	
<p>PCI</p> <ul style="list-style-type: none">- 32-bit PCI slot- Dimension: 9.0 mm x 85.0 mm	

Appendix B Random forest classification results

		Frequency features	Color features	Segment features	PCA reconstruction feature	Features selection from all feature sets
Tantalum capacitor	True positive	59/59 (100%)	58/59 (98.3%)	52/52 (100%)	45/52 (86.5%)	59/59 (100%)
	True negative	57/59 (96.6%)	58/59 (98.3%)	43/47 (89.6%)	45/48 (93.8%)	59/59 (100%)
SMD Aluminum electrolytic capacitor	True positive	108/112 (96.4%)	108/112 (96.4%)	92/94 (97.9%)	87/94 (96.9%)	110/112 (98.2%)
	True negative	109/112 (97.3%)	101/112 (90.2%)	70/96 (72.9%)	93/96 (96.9%)	112/112 (100%)
QFP100	True positive	78/79 (98.7%)	77/79 (97.5%)	61/65 (93.8%)	57/65 (87.7%)	79/79 (100%)
	True negative	75/79 (94.9%)	75/79 (94.9%)	62/69 (89.9%)	67/69 (97.1%)	79/79 (100%)
SMD Resistor Network array 1206, 4 Resistors	True positive	261/266 (98.1%)	258/266 (97.0%)	225/255 (100%)	222/225 (98.7%)	265/266 (99.6%)
	True negative	265/266 (99.6%)	231/266 (86.8%)	188/227 (82.8%)	224/227 (98.7%)	264/266 (99.2%)
SMD Transistor SOT23-3	True positive	258/262 (98.5%)	255/262 (97.3%)	217/223 (97.7%)	207/223 (93.2%)	261/262 (99.6%)
	True negative	259/262 (98.9%)	239/262 (91.2%)	179/223 (80.3%)	215/224 (96.4%)	258/262 (98.5%)
DIP14	True positive	111/114 (97.4%)	104/114 (91.2%)	93/99 (93.9%)	95/99 (96.0%)	112/114 (98.2%)
	True negative	109/114 (95.6%)	98/114 (86.0%)	88/95 (92.6%)	93/95 (97.9%)	113/114 (99.1%)
DIP16	True positive	65/72 (90.3%)	69/72 (95.8%)	53/57 (93.0%)	47/57 (82.5%)	71/72 (98.6%)
	True negative	70/72 (97.2%)	63/72 (87.5%)	56/65 (86.2%)	61/65 (93.8%)	71/72 (98.6%)
SMD Resistor 1206	True positive	264/266 (99.2%)	256/266 (96.2%)	218/226 (96.5%)	219/226 (96.9%)	265/266 (99.6%)
	True negative	262/266 (98.5%)	237/266 (89.1%)	192/226 (85.0%)	223/226 (98.7%)	265/266 (99.6%)
SOIC-8	True positive	103/106 (97.2%)	102/106 (96.2%)	82/88 (93.2%)	86/88 (97.7%)	103/106 (97.2%)

Appendices

		101/106 (95.3%)	93/106 (87.7%)	78/92 (84.8%)	92/92 (100%)	104/106 (98.1%)
Ceramic capacitor 1210	True positive	34/42 (81.0%)	42/42 (100%)	35/36 (97.2%)	29/38 (80.6%)	42/42 (100%)
	True negative	29/42 (69.0%)	39/42 (92.9%)	28/35 (80.0%)	25/35 (71.4%)	39/42 (92.9%)
SOT223-3	True positive	126/126 (100%)	116/126 (92.1%)	99/105 (94.3%)	105/105 (100%)	126/126 (100%)
	True negative	137/137 (100%)	107/137 (78.1%)	84/117 (71.8%)	116/117 (99.1%)	137/137 (100%)
SMD Resistor 0806	True positive	289/308 (93.8%)	296/308 (96.1%)	257/258 (99.6%)	241/258 (93.4%)	308/308 (100%)
	True negative	285/308 (92.5%)	276/308 (89.6%)	230/266 (86.6%)	255/266 (95.9%)	299/308 (97.1%)
TO263	True positive	35/36 (97.2%)	31/36 (86.1%)	25/29 (86.2%)	28/29 (96.6%)	34/36 (94.4%)
	True negative	36/36 (100%)	30/36 (83.3%)	28/32 (87.5%)	32/32 (100%)	36/36 (100%)
Quartz HC-49/S	True positive	43/46 (93.5%)	44/46 (95.7%)	32/27 (86.5%)	28/37 (75.7%)	46/46 (100%)
	True negative	45/46 (97.8%)	42/46 (91.3%)	33/36 (91.7%)	34/36 (94.4%)	43/46 (93.5%)
32-bit-PCI slot	True positive	77/77 (100%)	77/77 (100%)	60/63 (95.2%)	63/63 (100%)	77/77 (100%)
	True negative	76/7 (98.7%)	71/77 (92.2%)	63/68 (92.6%)	66/68 (97.1%)	76/77 (98.7%)

Appendix C Linear-SVM classification results

		Frequency features	Color features	Segment features	PCA reconstruction feature	Features selection from all feature sets
Tantalum capacitor	True positive	59/59 (100%)	59/59 (100%)	52/52 (100%)	48/52 (92.3%)	59/59 (100%)
	True negative	58/59 (98.3%)	58/59 (98.3%)	42/48 (87.5%)	44/48 (91.7%)	59/59 (100)
SMD Aluminum electrolytic capacitor	True positive	106/112 (94.6%)	107/112 (95.5%)	91/94 (96.8%)	89/94 (94.7%)	110/112 (98.2%)
	True negative	107/112 (95.5%)	98/112 (87.5%)	68/96 (70.8%)	94/96 (97.9%)	109/112 (97.3%)
QFP100	True positive	79/79 (100%)	71/79 (89.9%)	64/65 (98.5%)	58/66 (89.2%)	79/79 (100%)
	True negative	77/79 (97.5%)	74/79 (93.7%)	66/69 (95.7%)	67/69 (97.1%)	78/79 (98.7%)
SMD Resistor Network array 1206, 4 Resistors	True positive	207/266 (77.8%)	239/266 (89.8%)	221/233 (94.8%)	215/233 (92.3%)	222/266 (83.5%)
	True negative	204/266 (76.7%)	177/266 (66.5%)	156/219 (71.2)	179/219 (81.7%)	218/266 (82.0%)
SMD Transistor SOT23-3	True positive	257/262 (98.1%)	254/262 (96.9%)	215/222 (96.8%)	209/222 (94.1%)	259/262 (98.9%)
	True negative	258/262 (98.5%)	233/268 (88.9%)	189/223 (84.8%)	214/223 (96.0%)	260/262 (99.2%)
DIP14	True positive	109/114 (95.6%)	106/114 (93.0%)	89/99 (90.0%)	96/99 (97.0%)	110/114 (96.5%)
	True negative	111/114 (97.4%)	104/114 (91.2%)	85/95 (89.5%)	93/95 (97.9%)	113/114 (99.1%)
DIP16	True positive	63/72 (87.5%)	68/72 (94.4%)	48/57 (84.2%)	50/57 (87.7%)	69/72 (95.8%)
	True negative	69/72 (95.8%)	67/72 (93.0%)	58/65 (89.2%)	61/65 (93.8%)	71/72 (98.6%)
SMD Resistor 1206	True positive	261/266 (98.2%)	251/266 (94.4%)	212/226 (93.8%)	220/226 (97.3%)	265/266 (99.6%)
	True negative	260/266 (97.7%)	242/266 (91.0%)	197/226 (87.2%)	223/226 (98.7%)	266/266 (100%)
SOIC-8	True positive	103/106 (97.2%)	103/106 (97.2%)	76/88 (86.4%)	86/88 (97.7%)	106/106 (100%)
	True	103/106	90/106	74/96	92/92	1037106

Appendices

	negative	(97.2%)	(84.9%)	(80.4%)	(100%)	(97.2%)
Ceramic capacitor 1210	True positive	30/42 (71.4%)	41/42 (97.6%)	35/36 (97.2%)	25/36 (69.4%)	42/42 (100%)
	True negative	36/42 (85.7%)	35/42 (83.3%)	30/35 (85.7%)	28/35 (80.0%)	40/42 (95.2%)
SOT223-3	True positive	126/126 (100%)	113/126 (89.7%)	97/105 (92.4%)	105/105 (100%)	126/126 (100%)
	True negative	137/137 (100%)	113/137 (82.5%)	91/117 (77.8%)	116/117 (99.1%)	137/137 (100%)
SMD Resistor 0806	True positive	283/308 (91.9%)	290/308 (94.2%)	252/258 (97.7%)	244/258 (94.6%)	304/308 (98.7%)
	True negative	293/308 (95.1%)	269/308 (87.3%)	242/266 (91.0%)	253/266 (95.1%)	296/308 (96.1%)
TO263	True positive	35/36 (100%)	33/36 (91.7%)	26/29 (89.7%)	28/29 (96.55%)	34/36 (94.4%)
	True negative	36/36 (97.2%)	30/36 (83.3%)	26/32 (81.2%)	32/32 (100%)	36/36 (100%)
Quartz HC-49/S	True positive	42/46 (91.3%)	46/46 (100%)	30/37 (81.1%)	29/38 (78.4%)	46/46 (100%)
	True negative	45/46 (97.8%)	42/46 (91.3%)	35/36 (97.2%)	33/36 (91.7%)	46/46 (100%)
32-bit-PCI slot	True positive	77/77 (100%)	77/77 (100%)	61/62 (96.8%)	62/63 (98.4%)	77/77 (100%)
	True negative	77/77 (100%)	73/77 (94.8%)	60/68 (88.2%)	66/68 (97.1%)	77/77 (100%)

Appendix D RBF-SVM classification results

		Frequency features	Color features	Segment features	PCA reconstruction feature	Features selection from all features sets
Tantalum capacitor	True positive	59/59 (100%)	59/59 (100%)	52/52 (100%)	47/52 (90.4%)	59/59 (100%)
	True negative	59/59 (100%)	59/59 (100%)	47/48 (97.9%)	41/48 (85.4%)	59/59 (100%)
SMD Aluminum electrolytic capacitor	True positive	107/112 (95.5%)	107/112 (95.5%)	88/94 (93.6%)	88/94 (93.6%)	110/112 (98.2%)
	True negative	111/112 (99.1%)	106/112 (94.6%)	83/96 (86.5%)	90/96 (93.8%)	111/112 (99.1%)
QFP100	True positive	75/79 (94.9%)	76/79 (96.2%)	62/65 (95.4%)	59/65 (90.8%)	79/79 (100%)
	True negative	77/79 (97.5%)	75/79 (94.9%)	65/69 (94.2%)	59/69 (85.5%)	77/79 (97.5%)
SMD Resistor Network array 1206, 4 Resistors	True positive	264/266 (99.2%)	260/266 (97.7%)	211/225 (93.8%)	219/225 (97.3%)	265/266 (99.6%)
	True negative	266/266 (8100%)	255/266 (95.9%)	204/227 (89.9%)	223/227 (98.2%)	265/266 (99.6%)
SMD Transistor SOT23-3	True positive	257/262 (98.1%)	257/262 (98.1%)	212/222 (95.5%)	205/222 (92.3%)	262/262 (100%)
	True negative	258/262 (98.5%)	250/262 (95.4%)	201/223 (90.1%)	207/223 (92.8%)	262/262 (100%)
DIP14	True positive	111/114 (97.4%)	104/114 (91.1%)	91/99 (91.9%)	95/99 (96.0%)	112/114 (98.2%)
	True negative	112/114 (98.2%)	105/114 (89.5%)	85/95 (89.5%)	91/95 (95.8%)	112/114 (98.2%)
DIP16	True positive	69/72 (95.8%)	67/72 (93.0%)	51/57 (89.5)	50/57 (87.7%)	71/72 (98.6%)
	True negative	70/72 (97.2%)	67/72 (93.0%)	59/65 (90.8%)	52/65 (80.0%)	72/72 (100%)
SMD Resistor 1206	True positive	260/266 (97.7%)	260/266 (97.7%)	215/226 (95.1%)	222/226 (98.2%)	261/266 (98.1%)
	True negative	266/266 (100%)	258/266 (97.0%)	208/226 (92.0%)	220/226 (97.3%)	266/266 (100%)
SOIC-8	True positive	104/106 (98.1%)	105/106 (99.1%)	83/88 (94.3%)	87/88 (98.9%)	104/106 (98.1%)

Appendices

	True negative	105/106 (99.1%)	96/106 (90.6%)	76/92 (82.6%)	92/92 (100%)	106/106 (100%)
Ceramic capacitor 1210	True positive	36/42 (85.7%)	41/42 (97.6%)	35/36 (97.2%)	24/36 (66.6%)	41/42 (97.6%)
	True negative	34/42 (81.0%)	38/42 (90.5%)	34/35 (97.1%)	24/35 (68.6%)	41/42 (97.6%)
SOT223-3	True positive	125/126 (99.2%)	126/126 (100%)	99/105 (94.3%)	105/105 (100%)	126/126 (100%)
	True negative	137/137 (100%)	127/137 (92.7%)	110/117 (94.0%)	116/117 (99.1%)	137/137 (100%)
SMD Resistor 0806	True positive	294/308 (95.4%)	302/308 (98.0)	249/258 (96.5%)	242/258 (93.8%)	306/308 (99.4%)
	True negative	306/308 (99.3%)	290/308 (94.2%)	252/266 (94.7%)	247/266 (92.9%)	303/308 (98.4%)
TO263	True positive	36/36 (100%)	32/36 (88.9%)	26/29 (88.9%)	28/29 (96.6%)	34/36 (94.4%)
	True negative	36/36 (100%)	30/36 (83.3)	29/32 (90.6%)	31/32 (96.9%)	35/36 (97.2%)
Quartz HC-49/S	True positive	36/46 (78.3%)	46/46 (100%)	31/37 (83.8%)	27/37 (73.0%)	46/46 (100%)
	True negative	43/46 (93.5%)	45/46 (97.8%)	35/36 (97.2%)	29/36 (80.6%)	45/46 (97.8%)
32-bit-PCI slot	True positive	76/77 (98.7%)	76/77 (98.7%)	60/63 (95.2%)	62/63 (98.4%)	77/77 (100%)
	True negative	76/77 (98.7%)	71/77 (92.2%)	63/68 (92.6%)	67/68 (98.5%)	76/77 (98.7%)

Appendix E Decision level fusion results

True class/ Predicted class		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SMD Resistor 0806	1	39	0	0	0	0	0	0	0	0	0	0	0	0	0	2
Resistor Network array 1206	2	0	65	1	0	0	0	0	0	0	0	0	0	0	0	1
Resistor 1206	3	1	0	20	0	0	0	0	0	0	0	0	0	0	0	0
Transistor SOT23-3	4	0	0	2	53	0	0	0	0	0	0	0	0	0	0	1
Ceramic capacitor 1206	5	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0
Tantalum capacitor	6	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0
SOIC-8	7	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0
Aluminum electrolytic capacitor	8	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0
Quartz HC-49/S	9	0	0	0	0	1	0	0	0	6	0	0	0	0	0	0
SOT223-3	10	0	0	0	0	3	0	0	0	0	23	0	0	0	0	2
TO263	11	0	0	0	0	0	0	0	0	0	0	77	0	0	0	0
DIP16	12	0	0	0	0	0	0	0	0	0	0	0	13	0	0	4
DIP14	13	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0
QFP100	14	0	0	0	0	0	0	0	0	0	0	0	0	0	16	1
Unknown	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	67

Appendix F Basis weight determination (PCB mounted)

Length [cm]	Width [cm]	Weight [g]	Area [cm ²]	Basis weight [$\frac{g}{cm^2}$]
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

Appendix G Arduino Due component replacement model

Arduino Due component package	GaBi component replacement model	Number of components	Large component deviation
SMC_B	Kondensator Keramik MLCC 0603 (6mg) 1.6x0.8x0.8 PE	9	No
C0402	Kondensator Keramik MLCC 01005 (0,054 mg) 0,4x0,2x0,22	32	No
C0603	Kondensator Keramik MLCC 0603 (6mg) 1.6x0.8x0.8 PE	1	No
SMB	Transistor signal SOT23 3 leads (10mg) 1.4x3x1	1	Yes
MINIMELF	Diode MELF (130mg) D2.6x5.2	2	No
DO220AAL	Diode power DO214_219 (93mg) 4.3x3.6x2.3	1	No
SMD_1575SW	Schalter Tact (242mg) 6.2x6.3x1.8	1	Yes
L1812	Spule Multilayer Chip 1812 (108mg) 4.5x3.2x1.5	2	No
MSOP08	IC TSSOP 8 (28mg) 3.0x2.9x1.2 flash	1	Yes
SOT23-6	Transistor signal SOT23 3 leads (10mg) 1.4x3x1	1	Yes
SOT223	Transistor signal SOT223 3 leads (110mg) 3.8x7.65x2.3	1	No
MLF32	IC TQFP 32 (70mg) 5x5x1.0	1	Yes
SC70-5	IC TSSOP 8 (28mg) 3.0x2.9x1.2 flash	1	Yes
R0402	Widerstand Dickfilm Flat Chip 0402 (0.75mg)	18	No
CHIP-LED0805	LED SMD low-efficiency max 50mA (35mg) without Au 3.2x2.8x1.9	6	No
SRR0604	Spule Miniatur gewickelt SDR1006 (1.16g) D9.8x5.8	1	No
PANASONIC_D	Kondensator Al-Elko SMD (300mg) D6.3x5.4	2	No
SOT23	Transistor signal SOT23 3 leads (10mg) 1.4x3x1	1	No
R0603	Widerstand Dickfilm Flat Chip	3	No

Appendices

	0603 (2.1mg)		
TS42	Schalter Tact (242mg) 6.2x6.3x1.8	1	Yes
CAT16	4 x Widerstand Dickfilm Flat Chip 1206 (8.9mg)	4	Yes
SOT-23	Transistor signal SOT23 3 leads (10mg) 1.4x3x1	3	No
LQFP144	IC TQFP 100 (520mg) 14x14x1.0	1	Yes
CRYSTAL-3.2-2.5	0.5 x Quartz Crystal (500mg) 11.05x4.65x2.5	2	Yes
RESONATOR_EPSON_FC_145	1 x Quartz Crystal (500mg) 11.05x4.65x2.5	1	Yes
CT/CN0603	Widerstand Dickfilm Flat Chip 0603 (2.1mg)	5	No
PINHD-2x3	1.5 x Widerstand Dickfilm Flat Chip 0402 (0.75mg)	2	Yes
PINHD-1x8	2 x Widerstand Dickfilm Flat Chip 0402 (0.75mg)	5	Yes
PINHD-1x8	2 x Widerstand Dickfilm Flat Chip 0402 (0.75mg)	1	Yes
PINHD-1x10	2.5 x Widerstand Dickfilm Flat Chip 0402 (0.75mg)	1	Yes
PINHEAD_2X05_127	1.5 x Widerstand Dickfilm Flat Chip 0402 (0.75mg)	1	Yes
PINHD-2X18	9 x Widerstand Dickfilm Flat Chip 0402 (0.75mg)	1	Yes
PINHEAD_2X07_127	3.5 x Widerstand Dickfilm Flat Chip 0402 (0.75mg)	1	Yes
0805	Spule 0805	5	Yes
J0402	Kondensator Keramik MLCC 01005 (0,054 mg) 0,4x0,2x0,22	1	Yes
MCR-AB1-S-RA-SMT	Stecker, für Netzwerkkabel, ab Werk	1	Yes
POWERSUPPLY_DC-21MM	Stecker, für Netzwerkkabel, ab Werk	1	Yes
CON2_USB_MICRO_B_AT	4 x Widerstand Dickfilm Flat Chip 0402 (0.75mg)	1	Yes
FR4 glass epoxy	Leiterplatte 2-Lagen starr FR4		No
Solder SnAg3.5	Lotpaste SnAg		No

Appendix H Arduino Due estimated part prices

Arduino Due part	Price	Source
Kondensator MLCC 0603	0.016	http://de.farnell.com/yageo-phycomp/cc0603jrnpoabn101/kond-mlcc-c0g-np0-100pf-200v-0603/dp/1284111
Kondensator MLCC 01005	0.030	http://de.farnell.com/tdk/c0402x5r0j103k020bc/ceramic-capacitor-0-01uf-6-3v/dp/2354042
Kondensator MLCC 0603	0.016	http://de.farnell.com/yageo-phycomp/cc0603jrnpoabn101/kond-mlcc-c0g-np0-100pf-200v-0603/dp/1284111
Transistor SOT23	0.110	http://de.farnell.com/diodes-inc/b220-13-f/schottky-diode-20v-2a-smb/dp/1843749
CD1206-S01575	0.033	http://de.farnell.com/bourns/cd1206-s01575/schaltdiode-100v-150ma-1206/dp/2211947
Diode power	0.083	http://de.farnell.com/vishay/mss1p3l-m3-89a/schottky-diode-1a-30v-microsmp/dp/1815644
Spule 1812	0.114	http://de.farnell.com/multicomp/mcft000197/spule-ferrit-1812-1uh/dp/1711925
LMV358MMX	0.430	http://de.farnell.com/texas-instruments/lmv358mmx-nopb/ic-op-amp-1mhz-1v-us-soic-8/dp/1496055?ost=LMV358MMX
LM2736Y	0.891	http://de.farnell.com/texas-instruments/lm2736ymk/reg-buck-750ma-smd-sot23-6-2736/dp/1312554
MC33269ST-3.3T3	0.111	http://de.farnell.com/stmicroelectronics/ld1117s33tr/v-reg-ldo-3-3v-smd-1117-sot-223/dp/1202826
ATMEGA16U2-MU	4.670	http://de.farnell.com/atmel/atmega16u2-mur/ic-8bit-mcu-avr-mega-16mhz-nw/dp/2364798RL
74LVC1G125DCK	0.090	http://de.farnell.com/texas-instruments/sn74lvc1g125dckr/buffer-gate-single-smd-sc70-5/dp/1470771
Widerstand 0402	0.088	http://de.farnell.com/te-connectivity-amp/8-1879061-0/widerstand-0402-20k-0-1/dp/1863394
LED 0805	0.056	http://de.farnell.com/kingbright/kpt-2012sgc/led-0805-green-12mcd-568nm/dp/2099239
Spule SRR 0604	0.357	http://de.farnell.com/bourns/srr0604-100ml/leistungsinduktivit-t-10uh-20/dp/1929700
PANASONIC EEEFP1E680AP ALU-ELKO	0.139	http://de.farnell.com/panasonic-electronic-components/eeefp1e680ap/alu-elko-68uf-25v-smd/dp/1539487
BC847B	0.016	http://de.farnell.com/nxp/bc847b-215/transistor-npn-45v-sot-23/dp/1081232

Appendices

Widerstand 0603	0.022	http://de.farnell.com/multicomp/mc0063w0603121k/widerstand-0603-21k/dp/1170918
Button	0.125	http://de.farnell.com/c-k-components/ksem31j-lfs/taster-spst-0-05a-32vdc-smd/dp/2435311
Widerstandsnetzwerk	0.014	http://de.farnell.com/panasonic-electronic-components/exb38v103jv/widerstandsarray-konvex-0603x4/dp/2060107
ATSAM3X8EA-AU	6.330	http://de.farnell.com/atmel/atsam3x8ea-au/mcu-32bit-cortex-m3-84mhz-lqfp/dp/2318839?ost=ATSAM3X8EA-AU
CRYSTAL-3.2-2.5	0.196	http://de.farnell.com/fox-electronics/foxslf-120-20/quarz-12-0-mhz-20pf/dp/2063948
RESONATOR_EPSON_FC_145	2.280	http://de.farnell.com/tzc/7xz-32-768kbe-t/osc-32-768khz-3-2-x-2-5mm-cmos/dp/1892184
Varistor 0603	0.023	http://de.farnell.com/multicomp/mcvz0603m180agt/varistor-0603-14-vac/dp/1856931
Einbaubuchse Netzteil	0.770	http://de.farnell.com/lumberg/neb-21-r/einbaubuchse-leiterplatte-stift/dp/1217037
Micro USB Buchse	0.395	http://de.farnell.com/hirose-hrs/zx62d-b-5p8/steckverb-micro-usb-buchse-5pol/dp/2300437

Appendix I Material prices

Material name (GaBi)	Material name (Data source)	Price [\$/kg]	Data source	Year (annual mean)
Aluminium (E) [kg]	Aluminum metal	2.20	http://minerals.usgs.gov/minerals/pubs/commodity/aluminum/myb1-2012-alumi.pdf	2012
Aluminiumoxid (Al ₂ O ₃) (E) [kg]	-	0	-	-
Anorganische Flammschutzmittel (E) [kg]	-	0	-	-
Antimonoxid (Diantimontrioxid) (Sb ₂ O ₃) (E) [kg]	-	0	-	-
Blei (E) [kg]	Lead metal	2.40	http://minerals.usgs.gov/minerals/pubs/commodity/lead/myb1-2011-lead.pdf	2011
Blei in Legierung (E) [kg]	Lead metal	2.40	http://minerals.usgs.gov/minerals/pubs/commodity/lead/myb1-2011-lead.pdf	2011
Chrom in Legierung (E) [kg]	Chromium metal	13.30	http://minerals.usgs.gov/minerals/pubs/commodity/chromium/myb1-2012-chrom.pdf	2012
Cobalt in Legierung (E) [kg]	Cobalt (minimum of 99.8% cobalt)	29.30	http://minerals.usgs.gov/minerals/pubs/commodity/cobalt/myb1-2012-cobal.pdf	2012
Eisen in Legierung (E) [kg]	-	0	-	-
Elektrolyt (E) [kg]	-	0	-	-
Epoxidharz (EP) (E) [kg]	-	0	-	-
Ethylen-Propylen-Dien-Kautschuk (EPDM) (E) [kg]	-	0	-	-
Ferrite (E) [kg]	-	0	-	-
Glas (E) [kg]	-	0	-	-
Glasfasern (E) [kg]	-	0	-	-
Gold (E) [kg]	Gold metal	50562.00	http://minerals.usgs.gov/minerals/pubs/commodity/gold/myb1-2011-gold.pdf	2011
Gold in Legierung (E) [kg]	Gold metal	50562.00	http://minerals.usgs.gov/minerals/pubs/commodity/gold/myb1-2011-gold.pdf	2011
Kupfer (E) [kg]	Copper (London Metal Exchange,	7.30	http://minerals.usgs.gov/minerals/pubs/commodity/c	2013

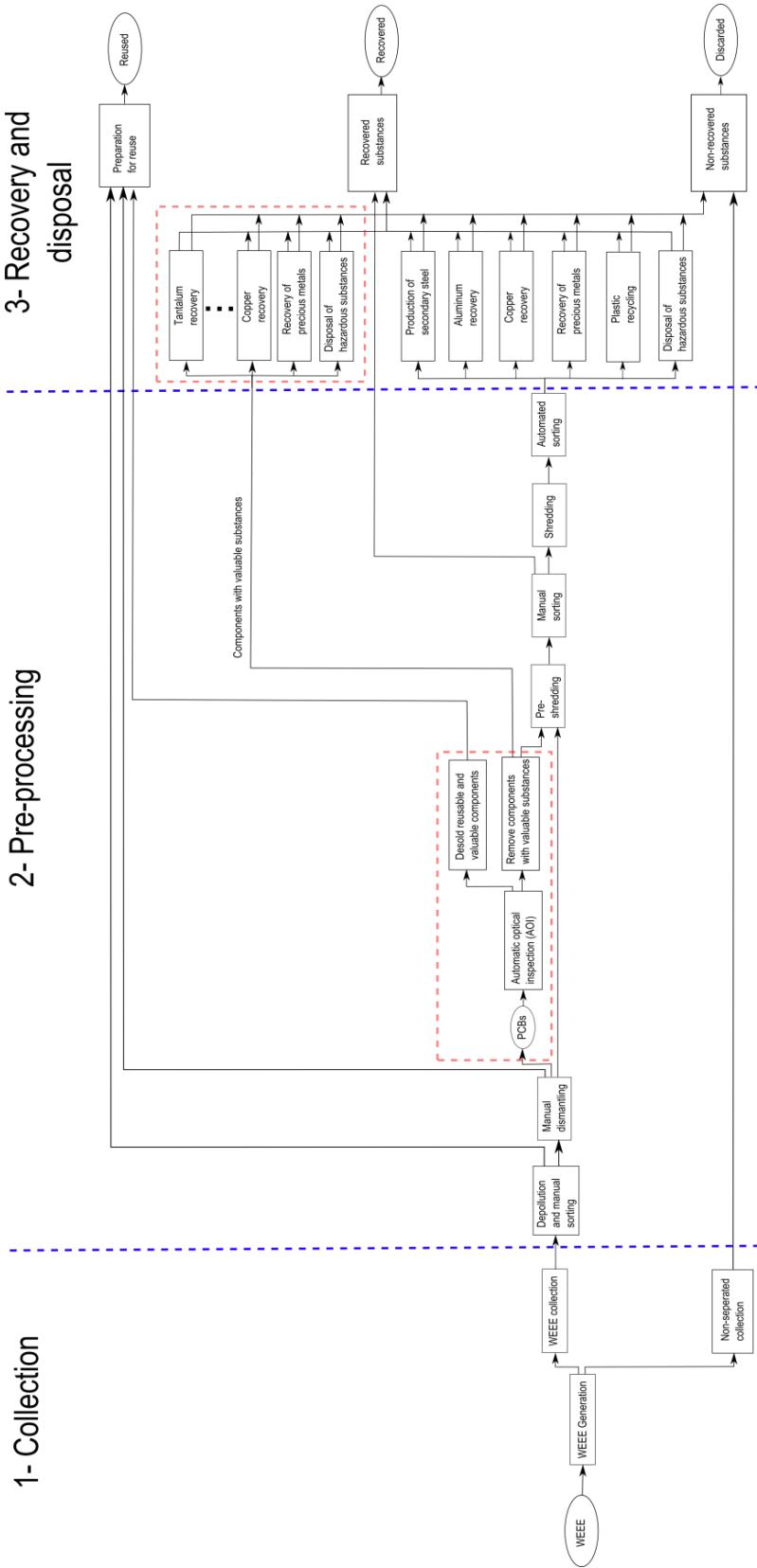
Appendices

	highgrade)		opper/mcs-2014-coppe.pdf	
Kupfer in Legierung (E) [kg]	Copper (London Metal Exchange, highgrade)	7.30	http://minerals.usgs.gov/minerals/pubs/commodity/copper/mcs-2014-coppe.pdf	2013
Nickel (E) [kg]	Nickel metal	22.90	http://minerals.usgs.gov/minerals/pubs/commodity/nickel/myb1-2011-nicke.pdf	2011
Nickel in Legierung (E) [kg]	Nickel metal	22.90	http://minerals.usgs.gov/minerals/pubs/commodity/nickel/myb1-2011-nicke.pdf	2011
Palladium in Legierung (E) [kg]	Palladium metal	23665.00	http://minerals.usgs.gov/minerals/pubs/commodity/platinum/mcs-2014-plati.pdf	2014
Papier (E) [kg]		0	-	
Platin in Legierung (E) [kg]	Platinum metal	48585.00	http://minerals.usgs.gov/minerals/pubs/commodity/platinum/mcs-2014-plati.pdf	2014
Polyamid 6 (PA6) (E) [kg]	-	0	-	-
Polyphenylensulfid (PPS) (E) [kg]	-	0	-	-
Polystyrol (PS) (E) [kg]	-	0	-	-
Polytetrafluorethylen (PTFE) (E) [kg]	-	0	-	-
Silber (E) [kg]	-	1133.76	http://minerals.usgs.gov/minerals/pubs/commodity/silver/myb1-2011-silve.pdf	2011
Silber in Legierung (E) [kg]	-	1133.76	http://minerals.usgs.gov/minerals/pubs/commodity/silver/myb1-2011-silve.pdf	2011
Silicium (E) [kg]	-	0	-	-
Siliciumdioxid (SiO2) (E) [kg]	-	0	-	-
Silikon (Si) (E) [kg]	-	0	-	-
Stahl, unlegiert (Fe-C) (E) [kg]	Steal (hot-rolled steel sheet)	0.63	http://minerals.usgs.gov/minerals/pubs/commodity/iron_and_steel/myb1-2012-feste.pdf	2012
Tetrabrombisphenol A (TBBA) (E) [kg]	-	0	-	-
Ungesättigte Polyester (UP) (E) [kg]	-	0	-	-
Zink (E) [kg]	Zinc (super high grade (SHG) zinc, 99.995% pure)	1.95	http://minerals.usgs.gov/minerals/pubs/commodity/zinc/myb1-2012-zinc.pdf	2012
Zink in Legierung (E) [kg]	Zinc (super high grade (SHG) zinc,	1.95	http://minerals.usgs.gov/minerals/pubs/commodity/zinc/myb1-2012-zinc.pdf	2012

Appendices

Zinn (E) [kg]	99.995% pure)		nc/myb1-2012-zinc.pdf	
Zinn in Legierung (E) [kg]	Tin metal	21.1	http://minerals.usgs.gov/minerals/pubs/commodity/tin/myb1-2012-tin.pdf	2012
Zinn in Legierung (E) [kg]	Tin metal	21.1	http://minerals.usgs.gov/minerals/pubs/commodity/tin/myb1-2012-tin.pdf	2012

Appendix J Improved WEEE recycling chain



Eidesstattliche Erklärung

Ich erkläre hiermit, dass ich diese Masterarbeit selbstständig ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst habe. Alle den benutzten Quellen wörtlich oder sinngemäß entnommenen Stellen sind als solche einzeln kenntlich gemacht.

Ort, Datum

Unterschrift Bernhard Föllmer