

# PCB part recognition for material recycling

---

BY

Bernhard Föllmer

Department of Machine Tools and Factory Management

Chair of Industrial Information Technology

Submitted in partial fulfillment of the requirements

for the degree of

Master of Science in

Physical Engineering Science

at the

Technical University of Berlin

13.10.2014

## **Abstract**

Sadsad

# Table of Contents

## Table of Contents

Abstract.....	2
Table of Contents.....	3
List of figures.....	8
List of tables .....	11
List of abbreviations.....	13
1. Introduction .....	14
1.1    Background.....	14
1.2    Problem formulation.....	14
1.3    Purpose.....	17
2.    Background Theories and related works .....	18
2.1    Feature extraction algorithms.....	18
2.1.1    Single seed region growing approach for color images.....	18
2.1.2    k-means clustering .....	21
2.1.3    Normalized cross correlation for 2-D pattern matching .....	22
2.1.4    Image reconstruction with PCA .....	23
2.2    Feature selection.....	24
2.2.1    Fisher score .....	25
2.2.2    Random forest feature selection .....	26
2.3    Object Classification .....	27
2.3.1    Random forest classifier .....	28
2.3.2    Support vector machine classifier .....	30

2.4	Data fusion model .....	33
	Data fusion with Dempster-Shafer theory.....	34
2.5	Optical character recognition of IC markings from electronic PCB scrap.....	36
2.5.1	Levenshtein distance .....	37
2.5.2	Octopart database for part-name verification .....	37
2.6	Life Cycle Inventory (LCI) analysis .....	38
2.6.1	Categorization of WEEE and PCB waste .....	39
2.6.2	Recycling potential of electronic PCB waste.....	40
2.6.3	Reuse potential of electronic parts from PCB waste.....	40
2.6.4	International Reference Life cycle Data System (ILCD) format# .....	41
3.	Methods for electronic component recognition .....	42
3.1	Image preprocessing .....	42
3.1.1	Image rotation correction.....	42
3.1.2	Scaling determination based on scaling symbol.....	45
3.1.3	Image resolution for feature extraction .....	50
3.2	Electronic component detection.....	51
3.2.1	PCB board segmentation .....	52
3.2.2	Color based PCB surface detection.....	53
3.2.3	Electronic component detection based on normalized 2-D cross-correlation .....	58
3.3	Feature extraction.....	60
3.3.1	A priori knowledge generation .....	61
3.3.2	Fourier coefficients based feature extraction .....	62
3.3.3	Histogram based feature extraction.....	64
3.3.4	Segment based feature extraction .....	65

3.3.5	PCA reconstruction error based feature extraction .....	67
3.4	Feature selection based on Fisher score and Random forest.....	70
3.5	Classification.....	71
3.5.1	Random forest classifier .....	72
3.5.2	Support vector machines.....	72
3.6	Data fusion model .....	73
3.6.1	Feature level fusion .....	73
3.6.2	Classifier level fusion.....	74
3.6.3	Decision level fusion with Dempster-Shafer theory .....	75
3.7	Optical character recognition of electronic component marking .....	80
3.7.1	Optical character recognition difficulties .....	80
3.7.2	Optical character recognition flow chart.....	81
3.7.3	Optical character recognition evaluation scheme.....	88
4.	Life-cycle inventory analyses of printed circuit boards .....	92
4.1	Printed circuit board region classification based on electronic part recognition results	
	92	
4.2	Printed circuit board LCI model .....	93
4.2.1	ILCD interface for automatic generation of LCI-models from PCBs <b>Error! Bookmark not defined.</b>	
	<b>Error! Bookmark not defined.</b>	
4.2.2	PCB flow diagram .....	<b>Error! Bookmark not defined.</b>
4.2.3	Data collection plan and data collection .....	95
4.2.4	Evaluation and results.....	97
5.	Implementation and experiments .....	98
5.1	Implementation.....	98

5.2	Dataset creation .....	98
5.2.1	Image acquisition .....	101
5.2.2	Dataset composition .....	102
5.3	PCB surface detection results .....	103
5.4	Feature selection results .....	104
5.4.1	Fourier features .....	105
5.4.2	Color features .....	106
5.4.3	Segment features.....	106
5.4.4	PCA reconstruction feature .....	107
5.4.5	Dependence of classification accuracy from number of selected features .....	108
5.5	Classification results.....	108
5.5.1	Random forest classifier results.....	108
5.5.2	Support vector machine classifier results.....	109
5.5.3	Multiclass classification result .....	109
5.6	Decision level fusion results .....	109
5.7	Optical character recognition results.....	110
5.7.1	Optical character recognition dataset and limits .....	110
5.7.2	Optical character recognition accuracy result on character level, word level, label level and part level.....	111
5.7.3	Octopart based part name assignment .....	112
5.7.4	Octopart based part price assignment .....	115
5.8	Life-cycle inventory analyses results.....	117
5.8.1	GaBi-Software and LCI data availability of electronic components .....	117
5.8.2	Recycling and reuse potential of electronic components .....	118

5.8.3	Arduino Due board LCI-model .....	118
6.	Discussion and future work .....	126
6.1	Electronic component detection.....	126
6.1.1	Electronic component detection based on 3D model .....	126
6.1.2	Electronic component detection based on Height map from laser triangulation	127
7.	Conclusion.....	128
7.1	Application inclusion in a PCB recycling process chain.....	129
Appendix A	Recognition database components.....	133
Appendix B	Most important selected features .....	136
Appendix C	Random forest classification results .....	137
Appendix D	Linear-SVM classification results.....	139
Appendix E	RBF-SVM classification results .....	141
Appendix F	Decision level fusion results.....	143
Appendix G	Basis weight determination (PCB mounted).....	144
Appendix H	Arduino Due component replacement model .....	145
Appendix I	Metal prices .....	148

## List of figures

Figure 1: Simplified recycling chain for WEEE .....	15
Figure 2: Mass balance of the preprocessing of 1,000 kg of input WEEE (Chancerell 2009).....	16
Figure 3: Connection between belief, disbelief, plausibility and doubt (Rakowsky 2007) .....	35
Figure 4: Image rotation correction process .....	43
Figure 5: Image rotated by 3.0 degree .....	44
Figure 6: Canny edge image of the rotated image .....	44
Figure 7: Shifted DFT of the rotated image (logarithmic representation) .....	44
Figure 8: Summed amplitude over angle (invariants by 90 degree) .....	45
Figure 9: Scale symbol.....	46
Figure 10: Scale symbol placed on the board .....	46
Figure 11: Scaling determination process.....	47
Figure 12: Value channel (brightness) of HSV color image.....	49
Figure 13: Cosine transform filtered image .....	49
Figure 14: Otsu thresholding .....	49
Figure 15: Blobs of the scaling symbol .....	49
Figure 16: Image resolution .....	51
Figure 17: PCB board segmentation process flow .....	52
Figure 18: Acquired PCB image .....	53
Figure 19: Otsu segmentation .....	53
Figure 20: Morphological eroded image with 10x10 kernel .....	53
Figure 21: Segmented PCB board image .....	53
Figure 22: PCB surface segmentation process flow.....	54
Figure 23: Original image .....	55
Figure 24: First 200 image segments based on region growing approach.....	55
Figure 25: PCB surface cluster pyramid .....	56
Figure 26: Image template for DIP14 component (RGB color space) .....	58
Figure 27: Spatial image resolution for 2D-cross correlation.....	59
Figure 28: Determined potential component positions for DIP14 component .....	60

Figure 29: Seed point grid (30 seed points) .....	62
Figure 30: DIP14 package with equidistant solder joints .....	63
Figure 31: Tantalum capacitor in RGB color model (left) and HSV color model (right) .....	65
Figure 32: Normalized histogram of hue channel (tantalum capacitor) .....	65
Figure 33: Normalized histogram of saturation channel (tantalum capacitor).....	65
Figure 34: Normalized histogram of value channel (tantalum capacitor).....	65
Figure 35: 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) .....	68
Figure 36: PCA feature construction process .....	69
Figure 37: Feature selection process chain .....	70
Figure 38: Data fusion model.....	73
Figure 39: Difficulties of IC marking recognition .....	81
Figure 40: IC marking recognition flow chart .....	85
Figure 41: OCR evaluation on character level.....	88
Figure 42: OCR evaluation on word level without Octopart .....	89
Figure 43: OCR evaluation on word level with Octopart .....	89
Figure 44: OCR evaluation on label level with Octopart.....	90
Figure 45: OCR evaluation on part level with Octopart.....	91
Figure 46: PCB flow diagram for LCI-model .....	94
Figure 47: PCB flow diagram for composition model .....	95
Figure 48: PCB regions .....	93
Figure 49: Component border definition.....	100
Figure 50: Database section.....	101
Figure 51: Image acquisition system.....	101
Figure 52: original PCB image .....	104
Figure 53: Sum of RBF-kernel SVM scores $wx, y$ (grayvalues are scaled between -20 and 20) 104	104
Figure 54: A comparison of different feature selection approaches.....	105

Figure 55: Resistor network 1206 and the most significant real part elementary image.....	106
Figure 56: Most important segment and seed point from ceramic capacitor .....	107
Figure 57: 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).....	108
Figure 58: Arduino Due board .....	119
Figure 59: Estimated material composition of Arduino Due parts [kg].....	121
Figure 60: Estimated material prices of Arduino Due parts .....	122
Figure 61: Estimated Gold distribution over Arduino Due parts.....	123
Figure 62: estimated Palladium distribution over Arduino Due parts.....	124
Figure 63: Estimated Arduino Due part prices.....	125
Figure 64: 3D model based component detection .....	126
Figure 65: Principle of laser triangulation (Torsten Koch 2013) .....	127
Figure 66: Improved PCB recycling process chain .....	132
Figure 67: Improved pre-processing step in PCB recycling process chain .....	130
Figure 68: Improved recover and disposal step in PCB recycling process chain .....	131

## List of tables

Table 1: Feature extraction algorithm based resolution parameter .....	50
Table 2: Dataset approaches for non-part images .....	71
Table 3: Component properties.....	99
Table 4: Dataset composition .....	102
Table 5: Confusion matrix of the predicted PCB surface training data .....	103
Table 6: Confusion matrix of the predicted PCB surface test data.....	103
Table 7: Random forest classification results .....	109
Table 8: OCR accuracy results.....	111
Table 9: Confusion matrix of the manual labeled words (word-level) verified with Octopart database.....	112
Table 10: Confusion matrix of the manual labeled labels (label-level) verified with Octopart database.....	113
Table 11: Accuracy rate of part assignment with manual labeled parts on word level verified with Octopart database (part-level) .....	113
Table 12: Accuracy rate of part assignment with manual labeled parts on label level verified with Octopart database (part-level) .....	113
Table 13: Confusion matrix of the Tesseract recognized words (word-level) verified with Octopart database .....	113
Table 14: Confusion matrix of the Tesseract recognized labels (label-level) verified with Octopart database .....	114
Table 15: Accuracy rate of part assignment with Tesseract OCR engine on word level verified with Octopart database (part-level) .....	114
Table 16: Accuracy rate of part assignment with Tesseract OCR engine on label level verified with Octopart database (part-level) .....	114
Table 17: Confusion matrix of the OCRMax recognized words (word-level) verified with Octopart database .....	114
Table 18: Confusion matrix of the OCRMax recognized labels (label-level) verified with Octopart database.....	115

Table 19: Accuracy rate of part assignment with OCRMax OCR engine on word level verified with Octopart database (part-level) .....	115
Table 20: Accuracy rate of part assignment with OCRMax OCR engine on label level verified with Octopart database (part-level) .....	115
Table 21: Arduino Due parts of the LCI model .....	119

## **List of abbreviations**

DFT

Discrete fourier transform, 41

FFT

Fast fourier transform, 42

LoG

Laplacian of Gausson, 102

## 1. Introduction

Ewt

### 1.1 Background

Ads

- Industrial PCB recycling process chain
- INPIKO erklären

### 1.2 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 od 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

Huismann et al. 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 makeup as well as the original equipment's manufacturing processes. 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 word 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 around 9% (Chancerell 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.

### - Reuse???

A simplified recycling chain for WEEE is shown in Figure 1. The recycling chain consist of three steps. The first step is the Collection of WEEE which is out of focus of 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

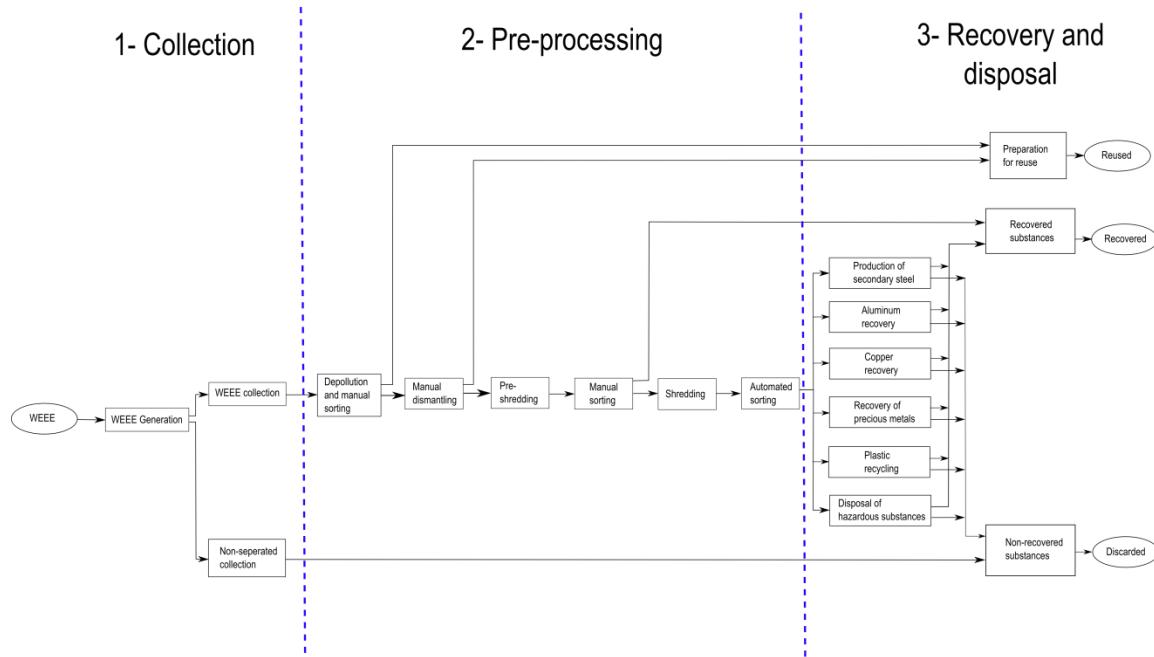
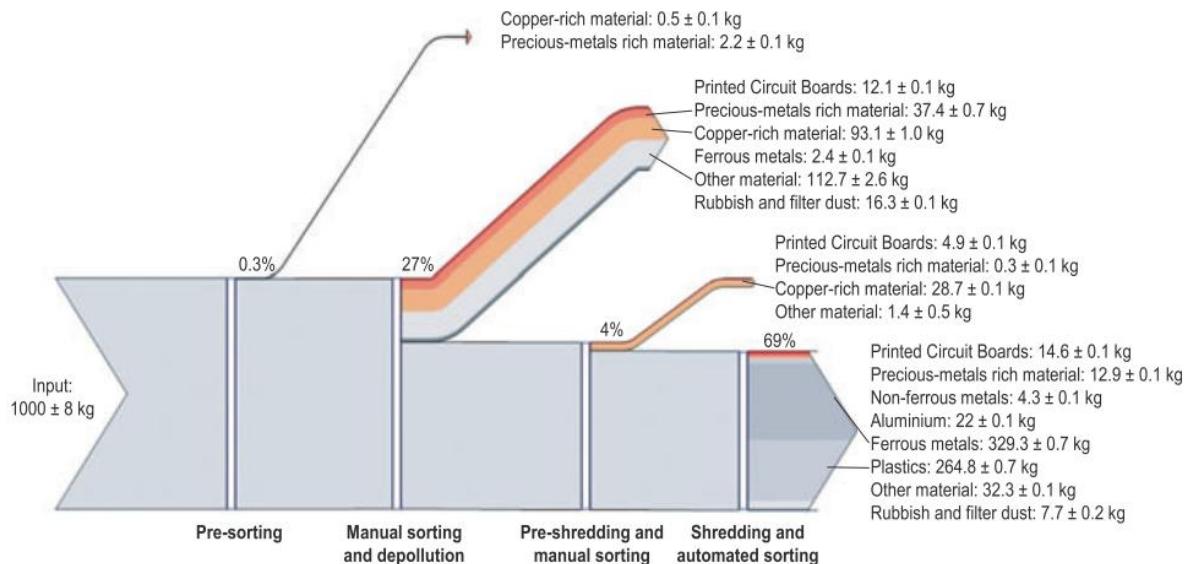


Figure 1: Simplified recycling chain for WEEE

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

- **PCB anteil (9%)?**



**Figure 2: Mass balance of the preprocessing of 1,000 kg of input WEEE (Chancerell 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 (Chancerell 2009). 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 (Chancerell 2009). Characterization of the waste stream is of paramount importance for developing a cost-effective and environmentally friendly recycling system (Jirang Cui 2003).

### 1.3 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. The automatic characterization of PCB waste 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 for the electronic component recognition system.

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 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 shown in chapter 0.

- continue

- Prozesskette optimieren

## 2. Background Theories and related works

### 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 2001).

#### 2.1.1 Single seed region growing approach for color images

For background segmentation and feature extraction from electronic part 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 (3.3.4 Segment based feature extraction, 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 were linear scaled between 0 and 1.

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

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

$$D_S = (f(x + 1, x + j, 2) - PG_{mean}(2))^2 \quad (3)$$

$$D_V = (f(x + 1, x + j, 3) - PG_{mean}(3))^2 \quad (4)$$

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

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

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

If the distance is 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 pixels to the region exceed the threshold, the region growing stops and the segmented region is determined as a segment of the image (O. Verma 2011). The pseudo code of the single seed region growing approach is shown in Code 1.

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  $\mu_1, \mu_2, \dots, \mu_k$  were 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:

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

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

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

The approximations of  $\hat{P}(w_i|x_k, \hat{\Theta})$  and the recomputations of the means are repeated until the approximations do not change compared to the previous iteration step (Duda 2001). The pseudo code is shown in Code 2.

#### PSEUDOCODE:

```

k_Means_Clustering(samples)
begin initialize n, k,  $\mu_1, \mu_2, \dots, \mu_k$ 
    do classify n samples according to nearest  $\mu_i$ 
        recompute  $\mu_i$ 
    until no change in  $\mu_i$ 
    return  $\mu_1, \mu_2, \dots, \mu_k$ 
end

```

[Code 2\\_ k-means clustering pseudo code \(Duda 2001\)](#)

### 2.1.3 Normalized cross correlation for 2-D 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 2-D 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  $\gamma$  is calculated between the image  $f$  and the template  $t$  whereas the template  $e$  is shifted by  $u$  steps in the  $x$  direction and  $v$  steps in the  $y$  direction. The normalized cross correlation value  $\gamma$  is calculated as follow:

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

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 templated  $t(x - u, y - v) - \bar{t}$ . 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 (Kai Briechle, Uwe D. Hanebeck 2001).

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  $m$  part 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  $\mu$  and the covariance matrix  $\mathbf{C}$  are computed for all vectors according to (14) and (15).

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

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

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  $\lambda_i$  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  $\mathbf{v}_i$  into the eigenspace is given by

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

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 whereas  $\mathbf{v}'_i$  is the reconstructed image of the image  $\mathbf{v}_i$ .

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

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

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

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

## 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 build a better prediction. Many techniques were 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:

- 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, 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 too 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 (Guyen 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 ration of the between class variance to the within-class variance (Guyen 2003). Suppose we have a set of d-dimensional samples  $x_1, \dots, x_n$ ,  $n_k$  is the number of samples in the subset  $D_k$  labeled  $\omega_k$  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  $\sigma^j$  is the standard deviation and  $\mu^j$  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  $x_i$ .

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

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

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

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

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 (Q. Gu, Z. Li, J. Han 2012). In this approach the fisher score is only used in the two stage feature selection and not applied alone for feature selection (see chapter 3.4).

### 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 not used to construct the  $k$ -th tree (oob-samples). Each sample that was left out to construct the tree is predicted by the  $k$ -th tree and compared to the true class of the sample. This is done with all trees of the random forest and the error over all trees and out-of-bag-samples are summed and divided by the number of out-of-bag-samples (Breiman, [www.stat.berkeley.edu](http://www.stat.berkeley.edu) 2014).

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

- Redundante features
- Plot tantalum importance

- Missing values (median, proximities)

## 2.3 Object Classification

Object recognition 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 approach 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 classifier 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 (Object recognition, wikipedia.org 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 are referred to classes or labels. The classification of pattern is based on classification models (classifiers) which are learning the reclassified patterns of the training set. An algorithm which constructs the model is called inducer and an instance of an inducer for a

specific training set is called a classifier. The idea behind an ensemble classifier is to weight several individual weak classifiers and combine them to form a strong inducer. It is well known that ensemble methods can improve the prediction performance (Rokach 2010).

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

- [Cart](#)
- [Breiman](#)
- [How to grow trees](#)

### *[Random forest training](#)*

Given a training set  $X = x_1, \dots, x_n$  with response  $y_1, \dots, y_n$ , bagging repeatedly selects bootstrap samples of the training set and fits trees to the samples. For each tree in the random forest classifier selects k random training samples  $X_b, Y_b$  (bootstrap samples) from the training set and trains the bagging trees  $f_b$  on  $X_b$  and  $Y_b$ . 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 (Breiman, Random Forests 2001). In random forests at each candidate split a random subset of features is selected. Typically for a dataset with p features  $\sqrt{p}$  features are used in each split (Random forest 2014).

- [How to grow trees](#)

### *[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  $\hat{f}_b$  the trained tree (Random forest 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, www.stat.berkeley.edu 2014).

- overestimate

### **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 (S. v. Wikipedia 2014). 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 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 (53) holds are placed on the hyperplane  $H_1: \mathbf{x}_i \cdot \mathbf{w} + b = 1$  and the point for which the equality (62) 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  $\alpha$  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  $\mathbf{w}$ ,  $b$ , and simultaneously require that the derivatives of  $L_p$  with respect to all the  $\alpha_i$  vanish, all 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 1997). 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:

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

This can be substituted in **Error! Reference source not found.** 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 1997)

### *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 polynomial kernels (C. Hsu, C.Chang, C. Lin 2003).

### ***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  $\gamma$  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.

The grid search method for the nonlinear RBF kernel SVM is taking  $m$  values in  $C$  and  $n$  values in  $\gamma$  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 (C. Qubo 2014).

## **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). Data fusion is a multidisciplinary area and difficult to classify. 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 (A.Steinberg 1999).

The Data fusion classification model which is manly used in image processing and used in this approach is the classification 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 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” (Dop 1999). 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) (Dop 1999).

## **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:

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

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:

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

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

$$doubt(B) = 1 - bel(B) \quad (37)$$

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 (Rakowsky 2007).

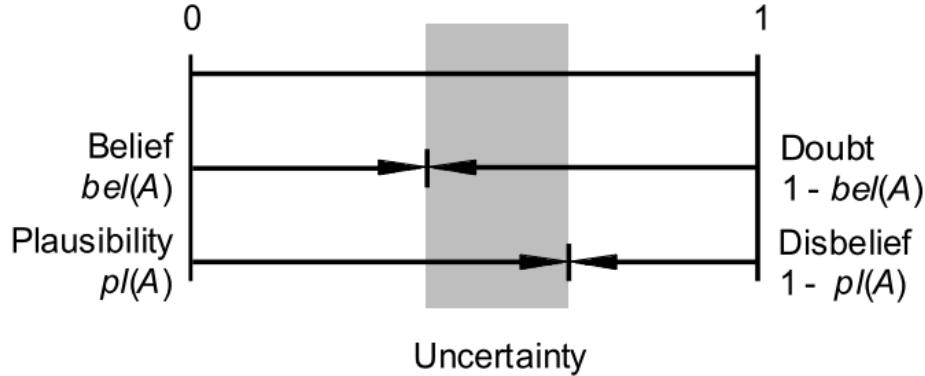


Figure 3: Connection between belief, disbelief, plausibility and doubt (Rakowsky 2007)

### 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  $m_1, \dots, m_s$ .

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 worldwide PCB production increases and quality control becomes more and more important. Very tiny and so many chaps are very difficult to check by humans. Therefore many OCR engines were developed to recognize character strings on chips or other electronic components. Most of the IC-marking recognition engines are developed for the inspection of chips and electronic components for assembly (Wang 2010). Information about the size and position of characters is well known. Many applications for IC marking inspection are developed Optical character verification (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 chips of 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. Unknown position, font or size of the characters, make it more difficult to recognize characters. Just a few publications deal with the task of IC marking recognition from electronic PCB waste (Wei Li 2014). An important measurement 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 different the strings are (people.cs.pitt.edu 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\_distance 2015).

### 2.5.2 Octopart database for part-name verification

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

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

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

or equal the distance threshold  $distance_{label, \text{oct}, \text{thresh}} = 2$  is assigned to the part. 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 (Curran 2006). An LCI is the basis of an Life cycle impact assessment (LCA) to evaluate comparative environmental impacts or potential improvements. With respect of reuse and recycling an LCI can assist organizations in comparing products or processes and considering environmental factors in material recycling. The “Guidelines for Assessing the Quality of Life Cycle Inventory Analysis” (Lynda Wynn, Eugene Lee 1995) provides a framework for performing an inventory analysis. Four steps are defined for making a life cycle inventory:

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

### 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 became a European law in 2003 (Parliament 2002). The 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 telecommunications equipment, Consumer equipment, Medical devices, Monitoring and control instruments and automatic dispensers are of particular importance (Chancerell 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)

Valuable PCB component can be categorized as gold connectors, mobile phone PCBs, CPI 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 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 prevents for landfill of hazardous materials in 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.

### **2.6.3 Reuse potential of electronic parts from PCB waste**

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 resoldered 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 are necessary to increase the potential of reusing electronic components.

### **2.6.4 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 (European Commission 2011). 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 modelling that are linked together via typed links called global references (Marc-Andree Wolf 2011). These types of data set concepts are:

- Process: Modelling 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 references 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

lop

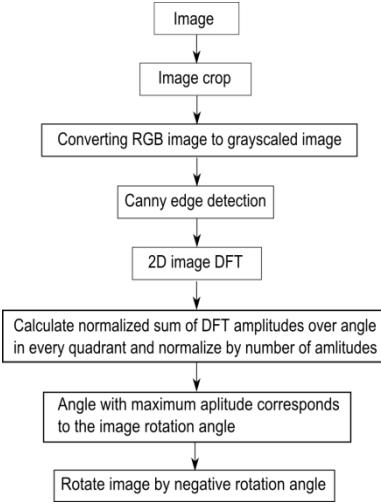
- preprocessing
- Detection (big search space, surface detection, segmentation)
- Classification (feature extraction, feature selection, classification, undefined class)
- OCR
- verification

### 3.1 Image preprocessing

- Scale invariante
- Rotation invariant

#### 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 parts are horizontal or vertical aligned. The whole process is based on the assumption that Conductor tracks and electronic parts are mostly horizontal or vertical aligned and there structure and borders producing more horizontal and vertical edges than edges with different orientations. The rotation angle estimation is based on the rotation property of a discrete Fourier transform. The DFT of an image rotated by an angle  $\Theta$  is the DFT of the unrotated image, rotated by the same angle  $\Theta$ . The rotation property of a DFT is derived in (Maria Petrou, Costas Petrou 2010) and therefore omitted here. The image rotation correction process is shown in Figure 4.



**Figure 4: Image rotation correction process**

At first the Image is cropped to a squared image [2000 x 2000] to reduce the process runtime. The RGB image is converted to grayscaled image and canny edge detection is applied. Afterward a 2D DFT is computed from the edge image. To estimate the rotation angle, the amplitude of the shifted 2D FFT image is summed up over discretized angles and normalized by number of amplitudes per angle step. The discretization is done in steps of 0.25 degree from 0 to 360 degree which results in a discretization error of 0.125 degree. The maximum of the normalized sum of amplitudes over the angle corresponds to the image rotation angle. With this process the rotation angle can be estimated with invariants of 90 degree image rotation. An example of a rotated image by 3 degree, the edge image and amplitude of discrete Fourier transform is shown in Figure 5, Figure 6 and Figure 7. The accuracy of the angle estimation was not investigated in detail but inaccuracy could not be determined by eye.

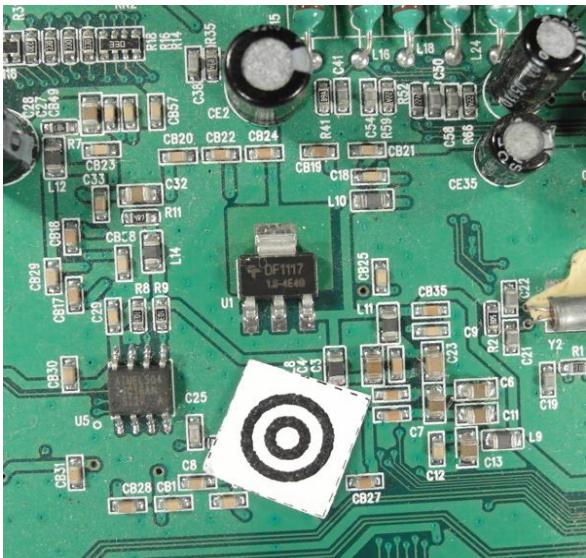


Figure 5: Image rotated by 3.0 degree

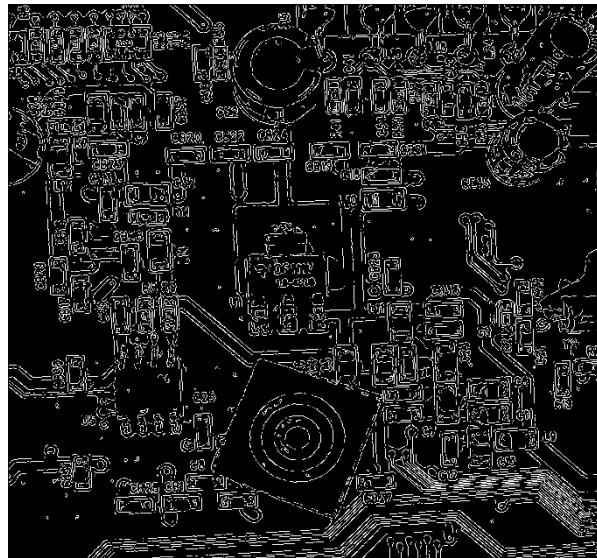


Figure 6: Canny edge image of the rotated image

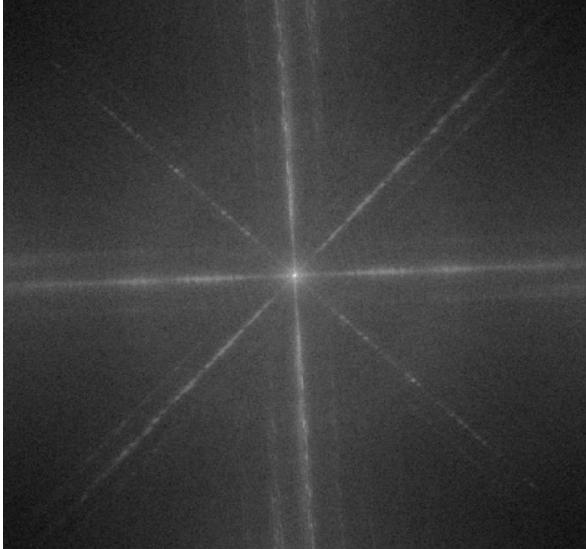


Figure 7: Shifted DFT of the rotated image (logarithmic representation)

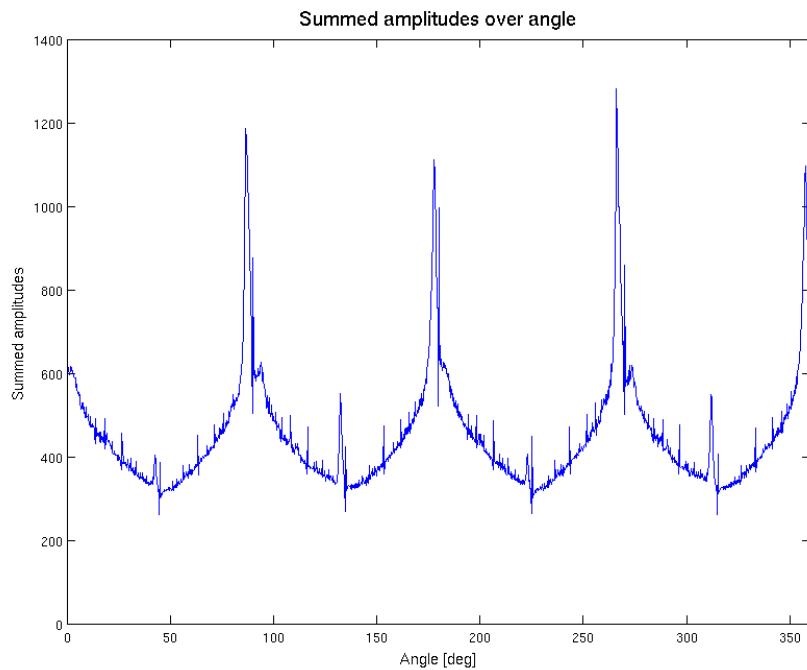


Figure 8: Summed amplitude over angle (invariants by 90 degree)

- Linien werden auf Punkte abgebildet

### 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.

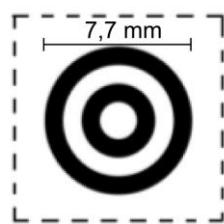


Figure 9: Scale symbol

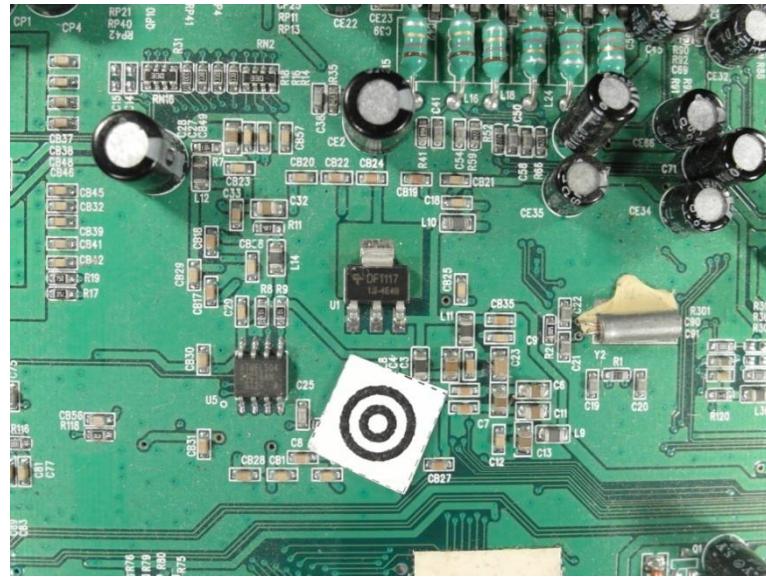
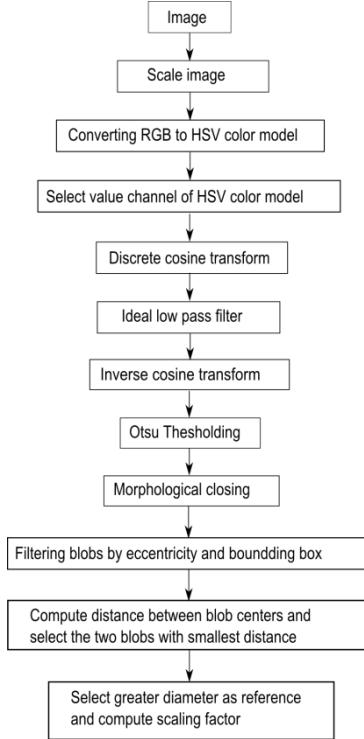


Figure 10: Scale symbol placed on the board

The scaling symbol is shown in Figure 9. The whole scaling determination process is shown in Figure 11.



**Figure 11: Scaling determination process**

At first the image is converted from the RGB color model to the HSV color model and the brightness channel (value channel) is used to make a discrete cosine transform. The discrete cosine transform is frequently used in image compression such as the JPEG format. The discrete cosine transform is similar to the discrete Fourier transform but uses only cosine functions as kernels. The discrete cosine transform is shown in Equation (43) and (44) (Rafael C.Gonzalez 2008).

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

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

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

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

$$\text{Maintained blobs} = \{ \text{blobs}, \text{eccentricity}_{min} < \text{eccentricity} \wedge \text{diameter}_{min} < \text{diameter} < \text{diameter}_{max} \} \quad (46)$$

- eccentricity min angeben

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

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

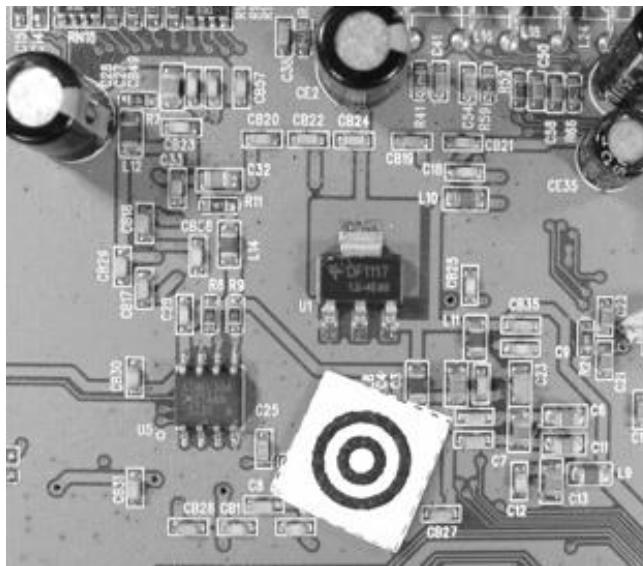


Figure 12: Value channel (brightness) of HSV color image

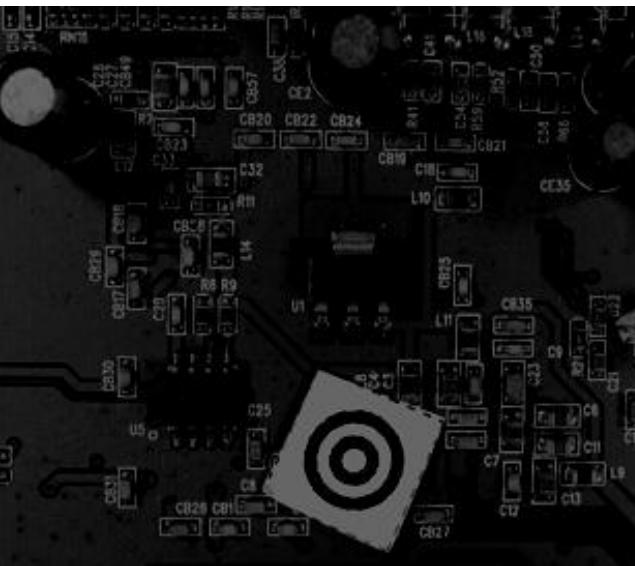


Figure 13: Cosine transform filtered image



Figure 14: Otsu thresholding



Figure 15: 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 trade off between a high image resolution on one hand and memory usage and runtime on the other side must be found. In this approach the image resolution depends on the size of the component. Smaller components require a higher resolution than larger ones because there images contain more details.

- entropy

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

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

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

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	18	0.005	2

The area and algorithm dependent resolution is plotted in Figure 16.

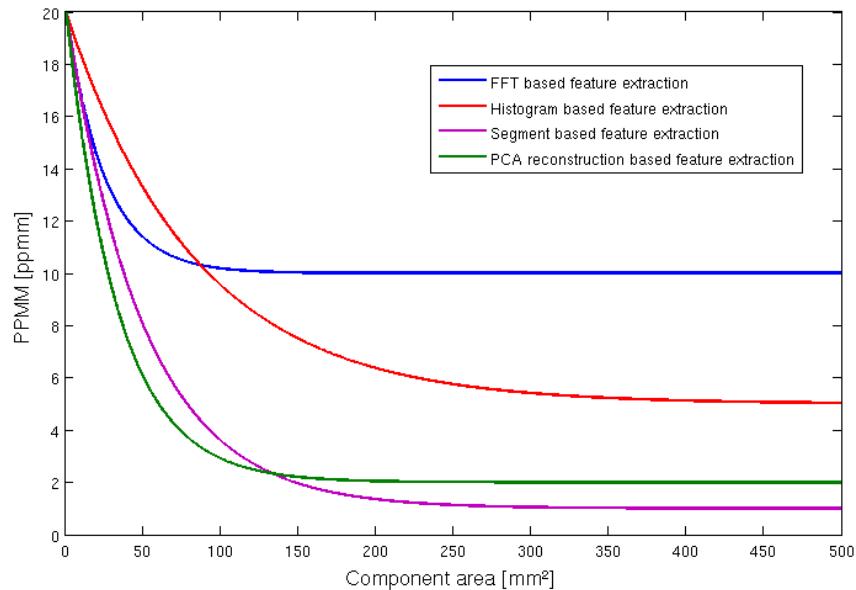


Figure 16: Resolution dependency from component area and feature extraction algorithm

### 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 2-D 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 search area for electronic components. In this approach the images of the PCB board where acquired where the PCB boards where placed on a white sheet what results in a white/bright background. In this approach the process flow shown in Figure 17 is applied.

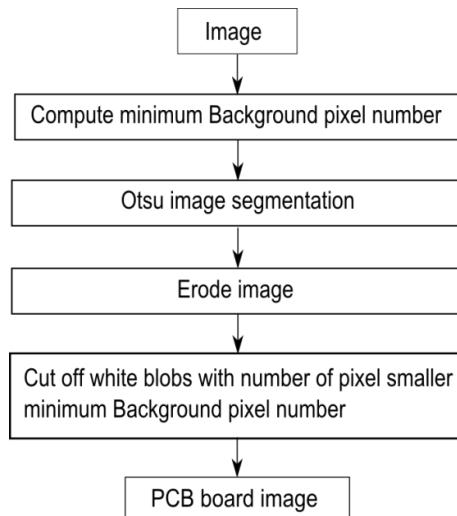


Figure 17: PCB board segmentation process flow

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.

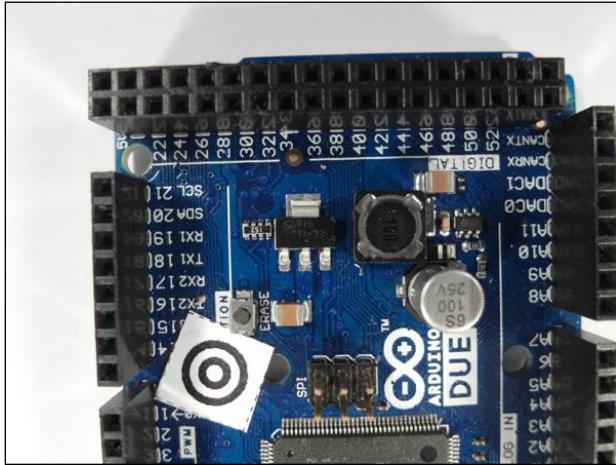


Figure 18: Acquired PCB image

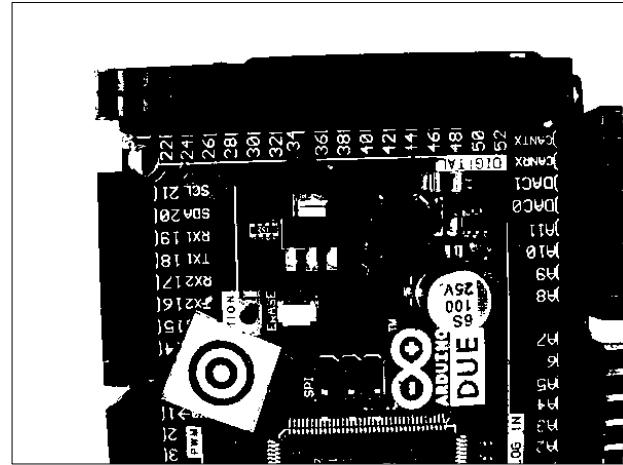


Figure 19: Otsu segmentation

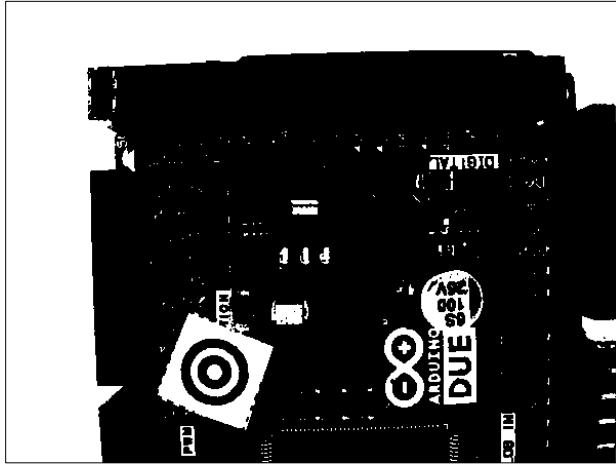


Figure 20: Morphological eroded image with 10x10 kernel

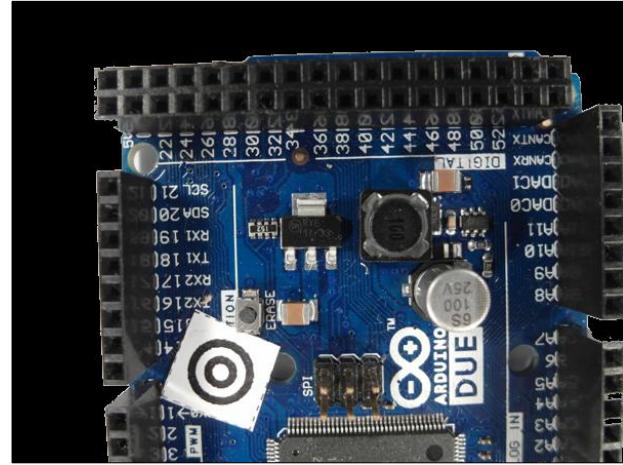


Figure 21: Segmented PCB board image

### 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.
- The number of surface pixel cluster is high compared to other pixel clusters based on the mostly large surface area compared to individual components
- The surface pixels form mostly large 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 22.

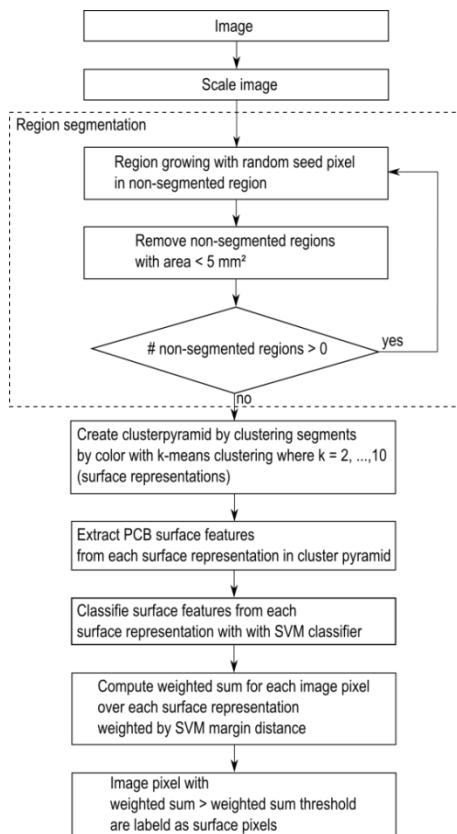


Figure 22: PCB surface segmentation process flow

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 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<sup>2</sup> 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 24.

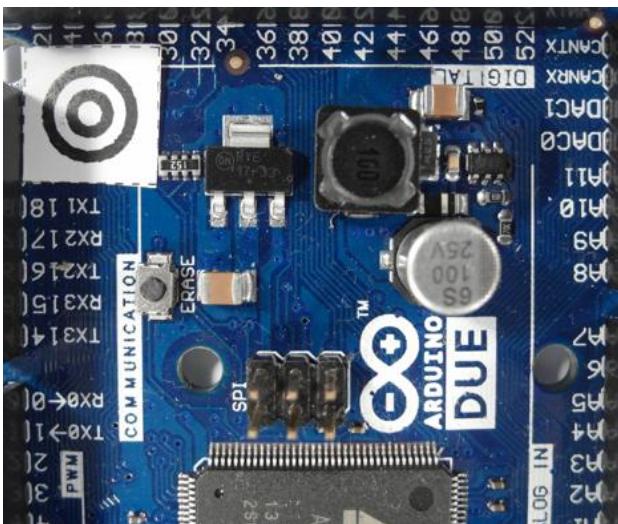


Figure 23: Original image

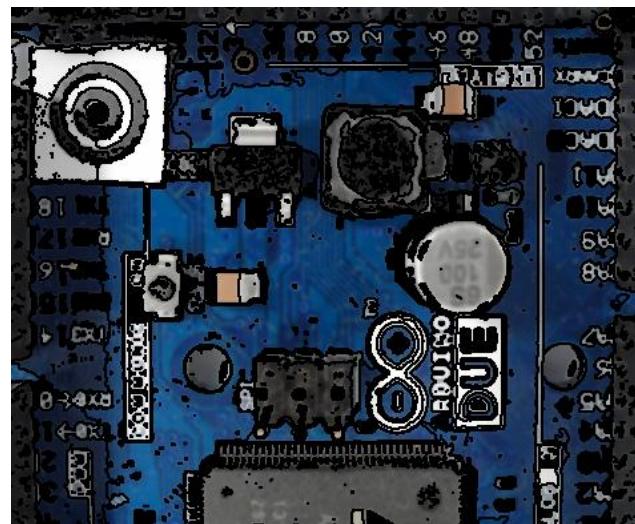


Figure 24: First 200 image segments based on region growing approach

After region growing, the segments are clustered based on their color. A cluster pyramid is drawn up where the number of clusters increases by one on each level of the cluster

pyramid. The k-means clustering algorithm is used with randomly selected initial set of k means. The k-means clustering algorithm is specified in chapter 2.1.2. The maximum number of cluster levels of the pyramid is set to ten ( $k_{max} = 10$ ) and is shown in Figure 25.

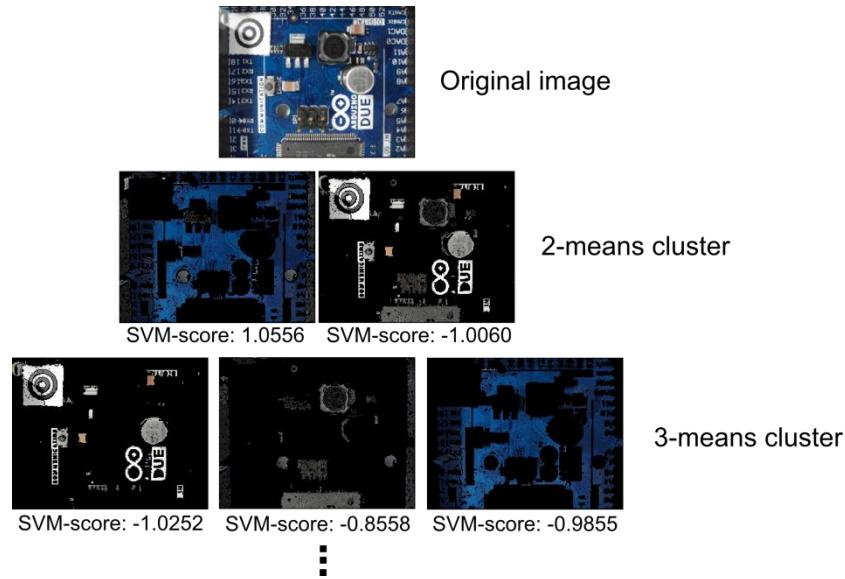


Figure 25: PCB surface cluster pyramid

After creating the cluster pyramid for all 54 surface representations ( $2 + 3 + 4 + \dots + 10 = 54$ ) the surface features are extracted. The nine surface features 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 f the same pyramid level (1-Feature)
- Number of surface representation edge pixels normalized by the maximum number of surface representation edge a surface representation 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 RBF-SVM parameter were estimated with a grid search method and 3-fold cross validation on the surface training set.

To train the RBF-Kernel SVM, each surface representation in the cluster pyramid of 77 images where 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 where labeled as PCB surface and 3250 clusters where 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 where treaded 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 treaded 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 2-D 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 templates in an image and is used in many object detection approaches. A detailed description about pattern matching with normalized 2-D 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 18.



Figure 26: Image template for DIP14 component (RGB color space)

In determining the image resolution a trade off between the computation time and spatial image resolution has to be made. 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 27.

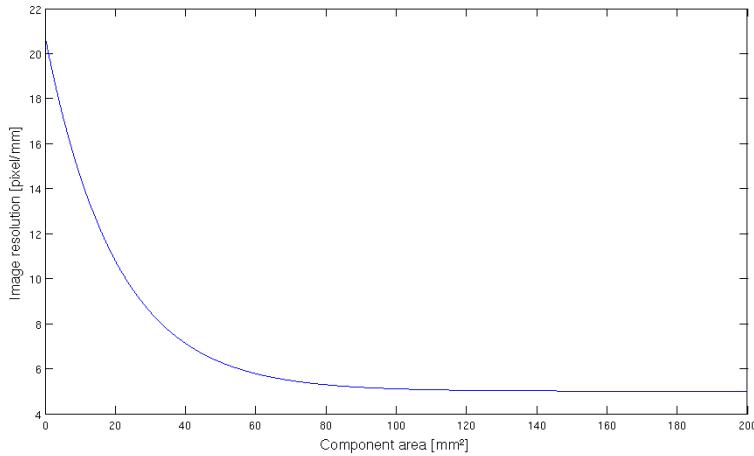


Figure 27: Spatial image resolution for 2D-cross correlation

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 spaces. The average correlation values over all three color spaces are determined and filtered by a 2D Gaussian kernel to get a score map  $p(x, y)$ . The Gaussian kernel has a size of 5x5 pixel and  $\sigma = 1.5$ . Scores  $p(x, y)$  greater than a 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.

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

An image and its determined potential component positions for the DIP14 component is shown in Figure 28. It can be seen that the two DIP14 parts in the image were recognized but also a false positive component position was selected.

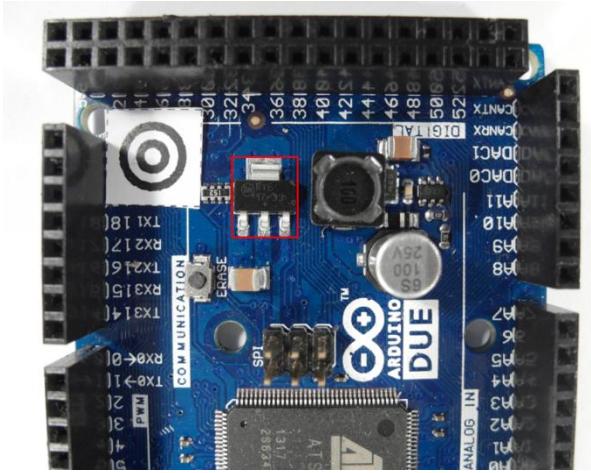


Figure 28: SOT223 transistor

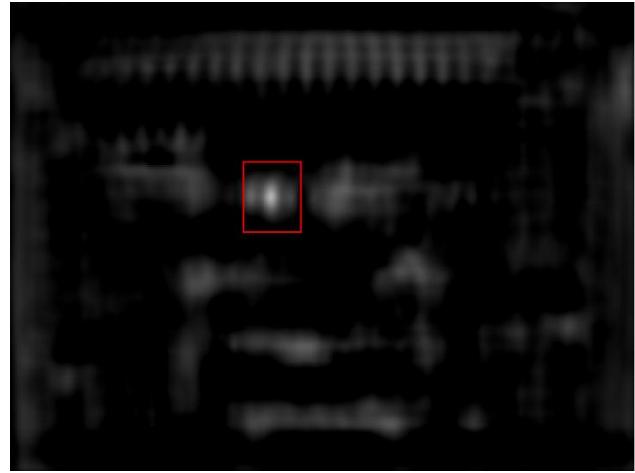


Figure 29: Determined potential component positions for SOT223 transistor

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

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.

The feature extraction algorithms are specified below.

### **3.3.1 A priori knowledge generation**

The extraction of representative features depends in two of the feature extraction algorithm 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 positions 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, 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 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 30.

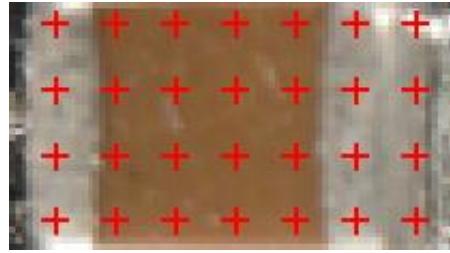


Figure 30: Seed point grid (30 seed points)

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, 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 features, specified in chapter 3.3.5, and is used to train the classifier. The projection matrix  $P_{ep}$  and the mean  $\mu_{ep}$  are computed from the a priori subset of component images and the projection matrix  $P_{en}$  and mean  $\mu_{en}$  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 periodical infinite signal can be decomposed **in ....**

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

The idea to use Fourier coefficients as features comes from the representation of solder joints by most electronic component images. Many computer vision systems for solder joint detection,

localization and segmentation have been developed. Specular reflections of solder joint depending on small changes in viewing direction and different shape and size of the solder joints make it difficult to create a stable recognition system (Tianshou 2012). Many electronic components consist of several equidistant arranged solder joints. An example is the widely used DIP14 package seen in Figure 31. Since the solder joints appear in the grayscaled image as bright equidistant spots they should be representative frequencies in the 2D Fourier spectrum with the period around the solder joint distance.



Figure 31: 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  $M \times N$  (Rafael C.Gonzalez 2008). The Fourier coefficients are in general complex numbers consisting of real and imaginary part. The real part represents the cosine and the imaginary the sinus proportion of the signal. The  $M \times N$  image consists of  $M \times N$  Fourier coefficients which produces  $2 \times M \times N$  frequency features which is a large number of features that can be used. To increase execution time of the classifier and decrease recognition rate, a subset of low frequency features is extracted. Further research shows that spatial frequencies with lower frequency represent global information about the shape such as general orientation and proportion. **The visual information is represented**

Since the solder joints are the main focus for frequency feature, the solder joint distance of electronic components is used as a measure of minimal frequency period. In our feature

extraction all Fourier coefficient (real and imaginary part) with a frequency under the cutoff frequency are used as features.

$$f_{cutoff} = \frac{1}{T_{cutoff}} = \frac{1}{0.50 \text{ mm}} = 2 \text{ mm}^{-1} \quad (55)$$

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

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

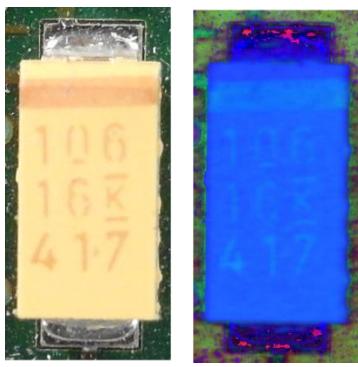
### Abtasttheorem (+1)

Another interesting feature extraction based on wavelets could analyze frequencies and their temporal occurrence which could improve the classification results. A view on that topic was done in the prospective section **Error! Reference source not found..**

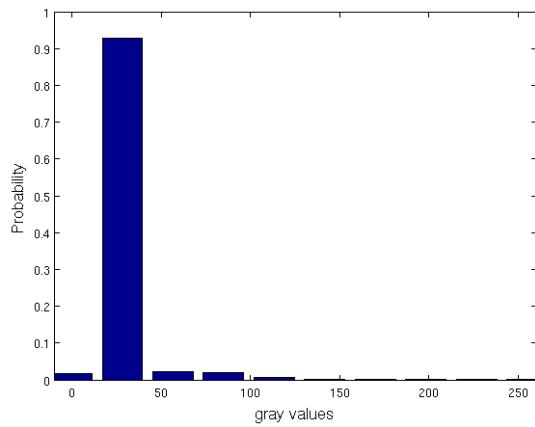
- Energie in niedrigen Frequenzen -> hohe Information (paper)

### 3.3.3 Histogram based feature extraction

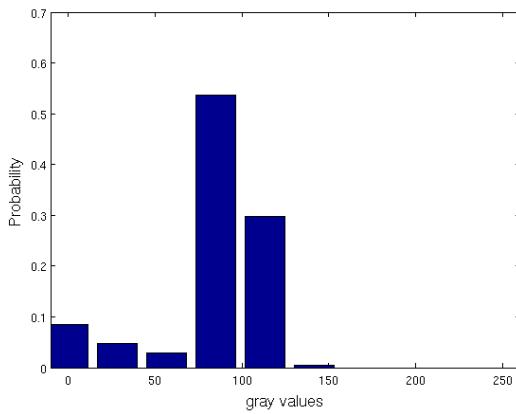
Color image segmentation algorithms for automated optical inspection in electronics have already been investigated (Tarnawski 2003). Electronic components varying in color, such as several tantalum capacitors, ICs or SMD electrolyte capacitors. To find representative features the color model has to be defined. In this system, the HSV (hue-saturation-value) color model was used because the channels are not that strongly correlated such as in the RGB color model and relatively stable against illumination changes or shadows (H. Cheng, H. Jiang, Y. Sun, Jingli Wang 2000), (Noor. A. Ibraheem, Mokhtar M. Hasan, Refiqul Z. Khan, Pramod K. Mishra 2012). Histogram based features are features which depend on the probability distribution of the pixels over the color values. In the histogram based feature extraction 10 equidistant bins are defined in each color channel (hue-saturation-value) and the pixel distributions are determined and normalized by the number of pixels. The values correspond to the probability density function of the gray value. All ten bin values are used as features that result in 30 color features. The histogram of a tantalum capacitor is seen in Figure 32, Figure 33, Figure 34 and Figure 35.



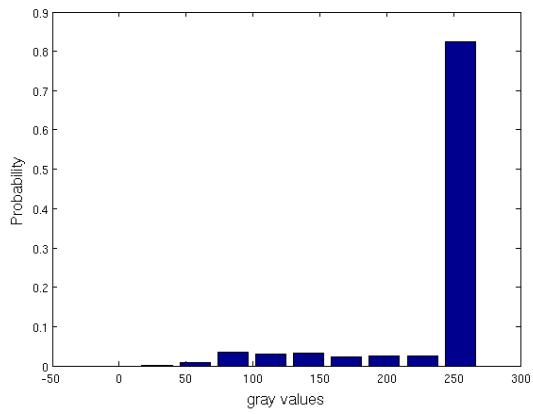
**Figure 32:** Tantalum capacitor in RGB color model (left) and HSV color model (right)



**Figure 33:** Normalized histogram of hue channel (tantalum capacitor)



**Figure 34:** Normalized histogram of saturation channel (tantalum capacitor)



**Figure 35:** 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 striking color regions. One approach to extract information about spatial proximity

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

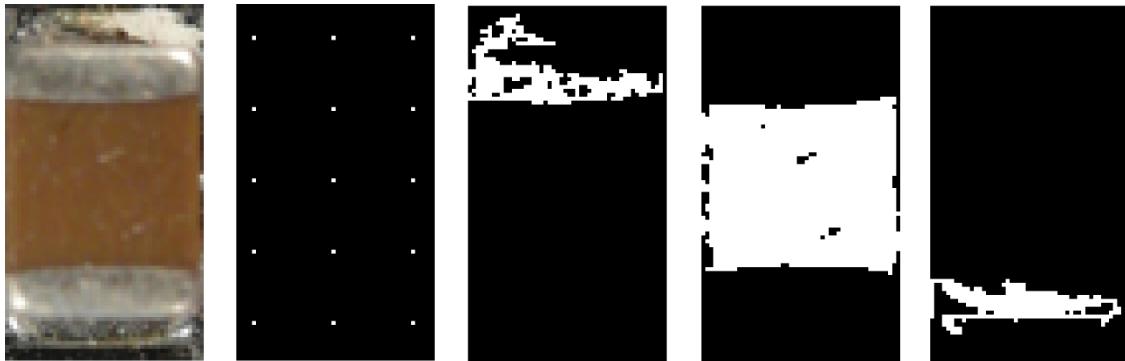
In this feature extraction algorithm, the seed points are uniformly distributed over the part image. The region growing and feature extraction of the segments is done in HSV color space. The distance between the seed points depends on the size of the component which is based on the assumption that smaller components consist of smaller color regions than big components. The equation for the distances is specified in (57).

$$A_{component} = length_{component} * width_{component} [\text{mm}^2] \quad (57)$$

$$\Delta_{seed} = 0.02 * A_{component} + 0.5 [\text{mm}] \quad (58)$$

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

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



- Abbildung korrigieren

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

- Border to seeds in picture
- Formeln region growing

### 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. (L. Malagón-Borja, Olac Fuentes 2007). A similar approach was used to extract a PCA reconstruction feature. In that system the PCA reconstruction is based on edge images of the parts. At first a subset of the training images of parts are used to find principal components which can compress optimally only the kind of images that were used to compute the principal components. The estimation of the principal components is specified in chapter 3.3.1.

- Übergang prüfen!!!

A set of PCs from a set of images from one component reconstruct the images of the same component better than other types of images. The fact can be observed in the images in Figure 36 and can be used to create a feature which represents the difference between the reconstruction error of the projection into the component PCs and the non-component PCs.

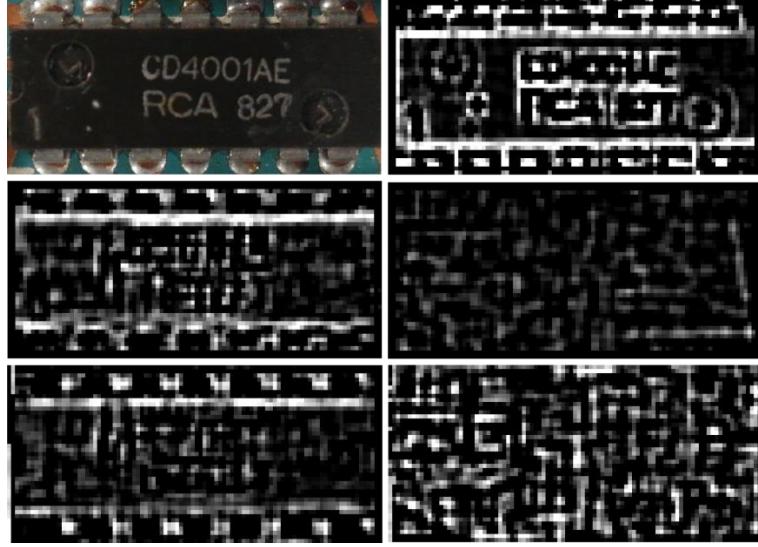


Figure 36: DIP14 (top, left), DIP14 edge image (top, right), DIP14 reconstruction with component PCs (middle, left), DIP14 reconstruction with non-component PCs (middle, right), unit matrix projection into component PCs (bottom, left), unit matrix projection into non-component PCs (bottom, right)

In this approach the component images and non-component images are scaled according to the size of the component. After the RGB images are converted to grayscaled images and the image intensity values are adjusted for contrast improvement. To obtain a feature that contains information about the edges the edge image was created by applying a Laplacian of Gaussian (LoG) filter. The projection matrix  $\mathbf{P}_{ep}, \boldsymbol{\mu}_{ep}, \mathbf{P}_{en}, \boldsymbol{\mu}_{en}$  are computed from the a priori subset specified in chapter 3.3.1. The reconstruction based on the component PC projection is computed by (59) and the reconstruction based on the non-component PC projection is computed by (60).

$$\mathbf{r}_{ep} = \mathbf{P}_{ep}^T \mathbf{P}_{ep} (\mathbf{e} - \boldsymbol{\mu}_{ep}) + \boldsymbol{\mu}_{ep} \quad (59)$$

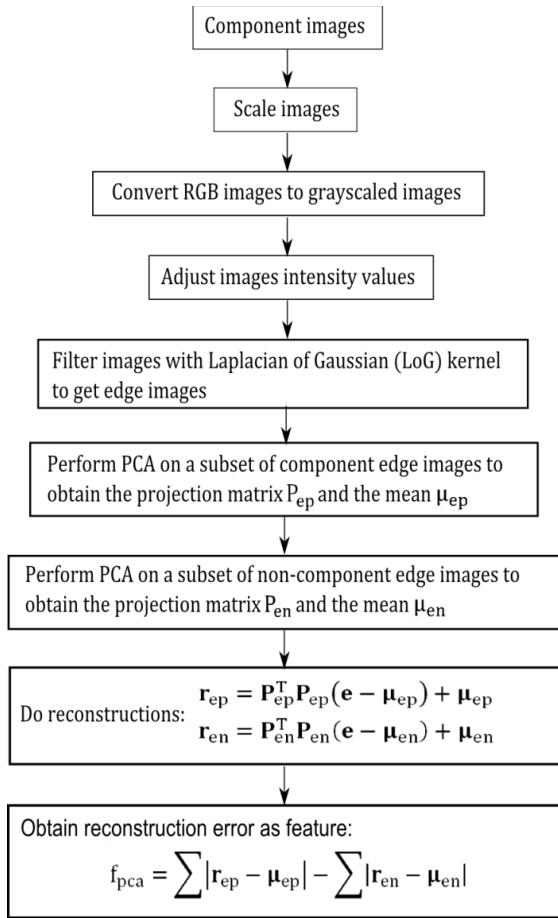
$$\mathbf{r}_{en} = \mathbf{P}_{en}^T \mathbf{P}_{en} (\mathbf{e} - \boldsymbol{\mu}_{en}) + \boldsymbol{\mu}_{en} \quad (60)$$

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

reconstruction error projected in the component PCs and the error projected in the non-component PCs shown in (61).

$$f_{pca} = \sum |r_{ep} - \mu_{ep}| - \sum |r_{en} - \mu_{en}| \quad (61)$$

The process is shown in Figure 37.



**Figure 37: PCA feature construction process**

Asdf

### 3.4 Feature selection based on Fisher score and Random forest

In practice random forest cannot handle a lot of features because it requires a lot of time to estimate the trees of the random forest and the accuracy decreases with a large number of features (Y. Chen, C.Lin 2003). This approach does feature selection in two steps. First the Fisher score is used to select a subset of features from the feature set with a large number of features. The features are selected by a fisher score threshold of 0.01. All features with a fisher score larger than the threshold value are selected for the second step. In the second step the random forest based feature selection from 2.2.2 is applied to select the most important features from step one. The process chain of the feature selection approach is shown in Figure 38.

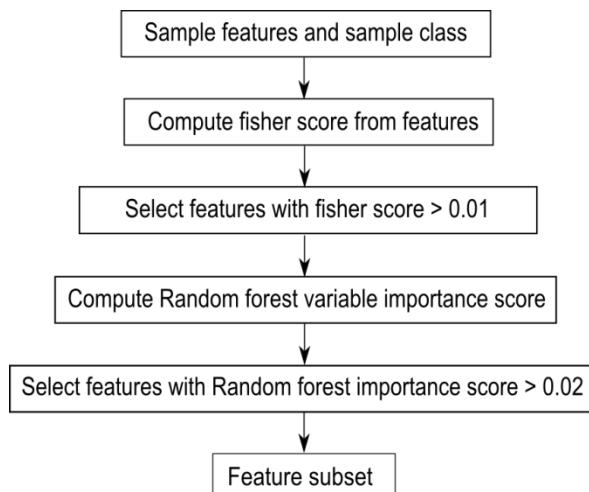


Figure 38: Feature selection process chain

### 3.5 Classification

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

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

Table 2: Dataset approaches for non-part images

Part images for DIP14	Non-part images for DIP14 (images from different parts)	Non-part images for DIP14 (images from arbitrary image section)

Both approaches use the same number of part images and non-part images and have advantages and disadvantages with respect to the representativeness of the data. If the non-part images consist of only images from different components, the variance of the non-part image set is smaller and the accuracy should be greater. On the other hand for the classifier it is

more difficult to handle non-part images from components which were not in the training set or images on which no part is seen.

### **3.5.1 Random forest classifier**

The used random forest parameters are as followed:

- Number of random forest trees: 100

### **3.5.2 Support vector machines**

### 3.6 Data fusion model

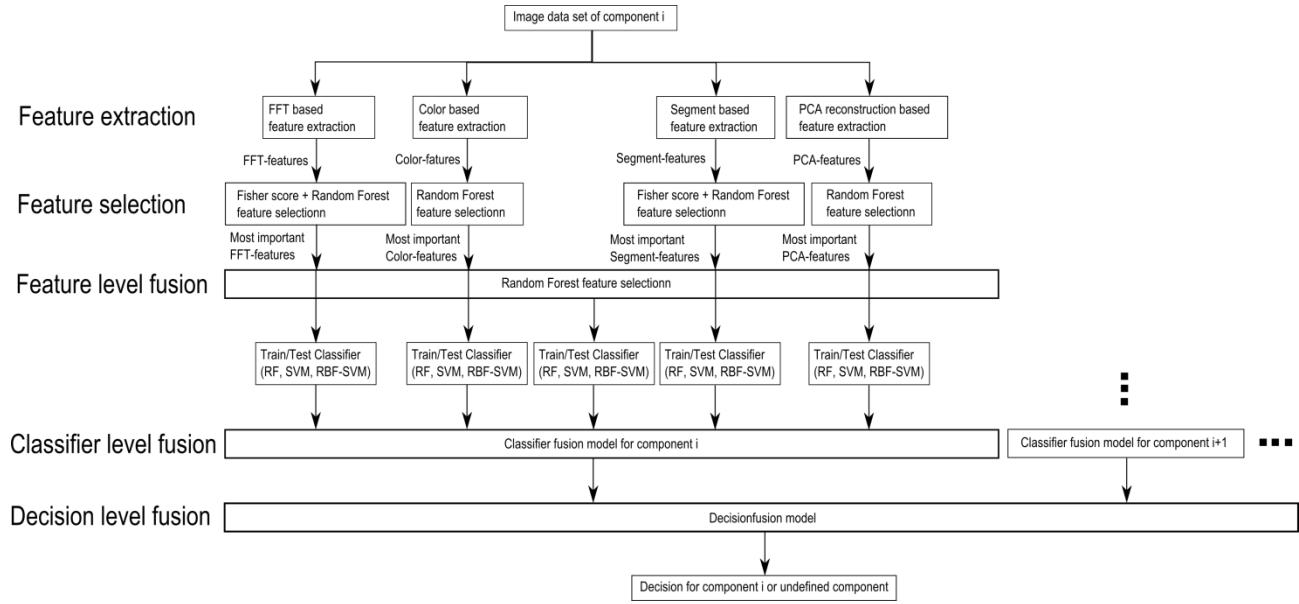


Figure 39: Data fusion model

#### 3.6.1 Feature level fusion

The inputs for the feature selection process are the extracted features in 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 PCA reconstruction error based features.

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

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 problems 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, [www.stat.berkeley.edu](http://www.stat.berkeley.edu) 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 39.

### 3.6.2 Classifier level fusion

The data fusion on classifier level (classifier level fusion) is performed to make the performance more robust against the difficulties that each individual classifier may have. Combining classifiers is one of the most widely explored methods in pattern recognition and it has been shown that these techniques can reduce error rate in classification tasks (Moreno-Seco n.d.). 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 all important features of all feature extraction algorithms. The largest groups of classifier fusion methods operate on classifiers which produce so-called soft outputs. The outputs are real values in range [0, 1] (D. Ruta, B. Gabrys 2000). The random forest classifier

outputs the number of votes for a class based on the number of trees. The number of votes can be normalized by the number of trees to get a soft output.

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

$$P_i = \sum_{k=1}^5 w_{i,k} * P_{i,k} \quad (62)$$

$$w_{i,k} = \frac{\sum_{j=1}^5 P_{i,j,test}}{P_{i,k,test}} \quad (63)$$

### 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  $P_i = 0$  corresponds to the situation where zero percent of the trees from the classifier  $i$  decided that the component is from class  $i$ . The value  $P_i = 1$  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 part is from component  $i$ . The outputs from all classifier fusion outputs are combined to make a final decision to which component the examined part belongs (Dong 2009).

The number of possible classes is  $n$  consist of the number of classifiers (equal the number of components)  $s$  and a class for unknown components ( $n = s + 1$ ). The basic probability assignment is made to each subset of the power set,

$$2^h = \{\emptyset, \{h_1\}, \dots, \{h_n\}, \{h_1, h_2\}, \dots, \{h_2, h_3\}, \{h_1, h_2, h_3\}, \dots, \{h_2, h_3, h_4\}, \dots, \Omega\} \quad (64)$$

whereas just the subset  $M$  contains masses unequal to zero:

$$M \subseteq 2^h \quad (65)$$

$$M = \{\{h_1\}, \{h_2, \dots, h_n\}, \{h_2\}, \{h_1, h_3, \dots, h_n\}, \dots, \{h_n\}, \{h_1, \dots, h_{n-1}\}\} \quad (66)$$

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 class one or class two, ..., or class  $n$ .

$$\{h_1, \dots, h_n\} = \{h_1 \cap \dots \cap h_n\} \quad (67)$$

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

$$m_i(\{h_j\}) = P_i \text{ for } i = j \quad (68)$$

$$m_i(\{h_1, \dots, h_n\} \setminus \{h_j\}) = 1 - P_i \text{ for } i \neq j \quad (69)$$

Where

$$i = 1, \dots, s \text{ and } j = 1, \dots, n \text{ and } n = s + 1 \quad (70)$$

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).

$$m_{1,\dots,s}(\{h_i\}) = K \cdot (1 - P_1) \cdot \dots \cdot (1 - P_{i-1}) \cdot P_i \cdot (1 - P_{i+1}) \cdot \dots \cdot (1 - P_s) \quad (71)$$

$$m_{1,\dots,s}(\{h_n\}) = K * (1 - P_1) \cdot \dots \cdot (1 - P_s) \quad (72)$$

The measure of contradiction  $K$  is calculated as follows:

$$K^{-1} = (1 - P_1) \cdot \dots \cdot (1 - P_s) \\ + \sum_{i=1}^s (1 - P_1) \cdot \dots \cdot (1 - P_{i-1}) \cdot P_i \cdot (1 - P_{i+1}) \cdot \dots \cdot (1 - P_s) \quad (73)$$

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

$$bel(\{h_j\}) = m(h_j) \quad (74)$$

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

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

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 (Jürg Kohlas 1995).

The component class with the maximum belief (component class 1) 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. An unknown 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  $P_1 = 0.9$  is the output of the classifier  $i = 1$  which gives a score that the component belongs to Component class 1. The output  $P_2 = 0.2$  is the output of the classifier  $i = 2$  which gives a score that the component belongs to Component class 2. The outputs from the classifier fusion level are shown in Table 3.

**Table 3: 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

$$2^h = \{\emptyset, \{h_1\}, \{h_2\}, \{h_u\}, \{h_1, h_2\}, \{h_2, h_u\}, \{h_1, h_u\}, \{h_1, h_2, h_3\}\} \quad (76)$$

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.

**Table 4: Basic probability assignments**

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

The measure of contradiction  $K$  is determined according to (73) as follows:

$$\begin{aligned}
 K^{-1} &= m_1(\{h_1\}) \cdot m_2(\{h_1 \cup h_u\}) + m_1(\{h_2 \cup h_u\}) \cdot m_2(\{h_2\}) \\
 &\quad + m_1(\{h_2 \cup h_u\}) \cdot m_2(\{h_2\}) = 0.9 \cdot 0.8 + 0.1 \cdot 0.2 + 0.8 \cdot 0.1 \\
 &= 0.82
 \end{aligned} \tag{77}$$

The combination of the assigned probabilities according to (71) and (72) is done as follows:

$$m_{1,2}(\{h_1\}) = K \cdot m_1(\{h_1\}) \cdot m_2(\{h_1 \cup h_u\}) = \frac{0.9 \cdot 0.8}{0.82} = 0.8780 \tag{78}$$

$$m_{1,2}(\{h_2\}) = K \cdot m_2(\{h_2\}) \cdot m_1(\{h_2 \cup h_u\}) = \frac{0.2 \cdot 0.1}{0.82} = 0.0244 \tag{79}$$

$$m_{1,2}(\{h_u\}) = K \cdot m_2(\{h_2\}) \cdot m_1(\{h_2 \cup h_u\}) = \frac{0.8 \cdot 0.1}{0.82} = 0.0976 \tag{80}$$

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

**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, check and passport or license plate recognition. The automatic inspection of IC markings is a field which mainly focuses on inspection and quality control of PCB assembly processes. Inappropriate placement of chips and surface mounted devices (SMDs) can automatically be detected and corrected (B. Luo, Y.Gao, Z.Sun 2013). This approach is focusing on the inspection of IC markings whereas makings of other component like capacitors or coils are out of focus because of their complexity.

#### **3.7.1 Optical character recognition difficulties**

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

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

Some difficulties about IC marking recognition from electronic scrap are shown in Figure 40.

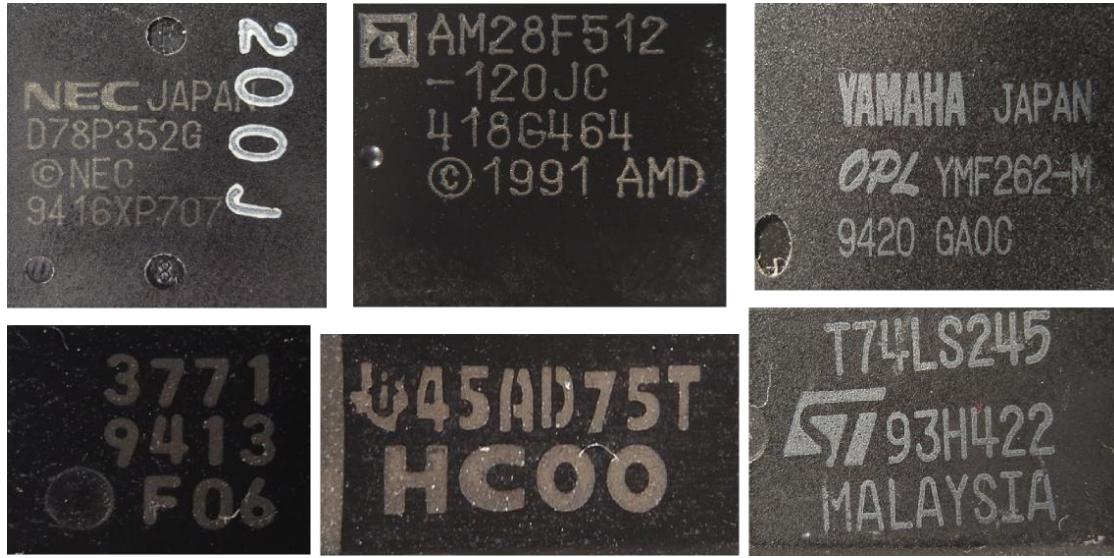


Figure 40: Difficulties of IC marking recognition

### 3.7.2 Optical character recognition flow chart

- Assumption made for the algorithm (character size, baselines,...)

The most important step of this OCR approach is the character recognition 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 (Patrick Blaes, Chris Young n.d.). 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 41 were carried out.

Each IC component has different requirements for the IC marking recognition algorithm what does it matter that the markings have different properties. Properties which have to be known for the algorithm and are stored in the component database are the region of interest (ROI) for

the IC marking and the subset of characters making up the marking. For the SMD resistor 1206 component for example the character subset could be {"0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "R"} because smaller character subsets increases the recognition rate. The marking recognition flow chart is shown in Figure 41.

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

To emphasize the characters of the markings a Laplacian of Gaussian (LoG) filter is applied. The LoG kernel is a rotationally symmetric filter which is mainly used for edge detection. The filter is composed of the second derivative (Laplace operator) of a Gaussian filter shown in equations (82) and (81). 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 1mm ( $h = \text{imagescale} [\text{pixel}]$ ) and in practice lies between 50 and 120 pixels. The standard deviation of the Gaussian is constant  $\sigma = 0.5$ .

- **Explane LoG operator**

$$G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (81)$$

$$\nabla G(x, y) = \frac{\partial^2 G(x, y)}{\partial x^2} + \frac{\partial^2 G(x, y)}{\partial y^2} \quad (82)$$

The next step is the blob segmentation which is done by Otsu's segmentation method (OTSU 1978). Otsu method is a segmentation process based on a global segmentation threshold which is computed by minimizing the intra-class variance (variance within classes). After segmentation step a morphologic closing operator is applied to reduce holes in the character blobs. The size of the rectangular closing kernel ( $h \times h$ ) changes linear with the image scale  $h = [0.05 * \text{imagescale}][\text{pixel}]$ .

After the segmentation process blobs that do not correspond to a character still exist. Therefore the areas of the blobs are estimated and blobs with an area smaller  $area_{min}$  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 computing the probability density estimate which is done with the Matlab function `ksdensity`. The function returns a probability density estimate for samples based on a normal kernel function and is evaluated at equally spaced points that cover the range of the data (`ksdensity`, mathworks 2014). In this approach 1000 equally spaced points from zero to one are used whereas the samples are normalized by the height of the image, the smoothing parameter  $\sigma$  is set to 0.025. All local maximums in the probability density function are potential lower character baselines.

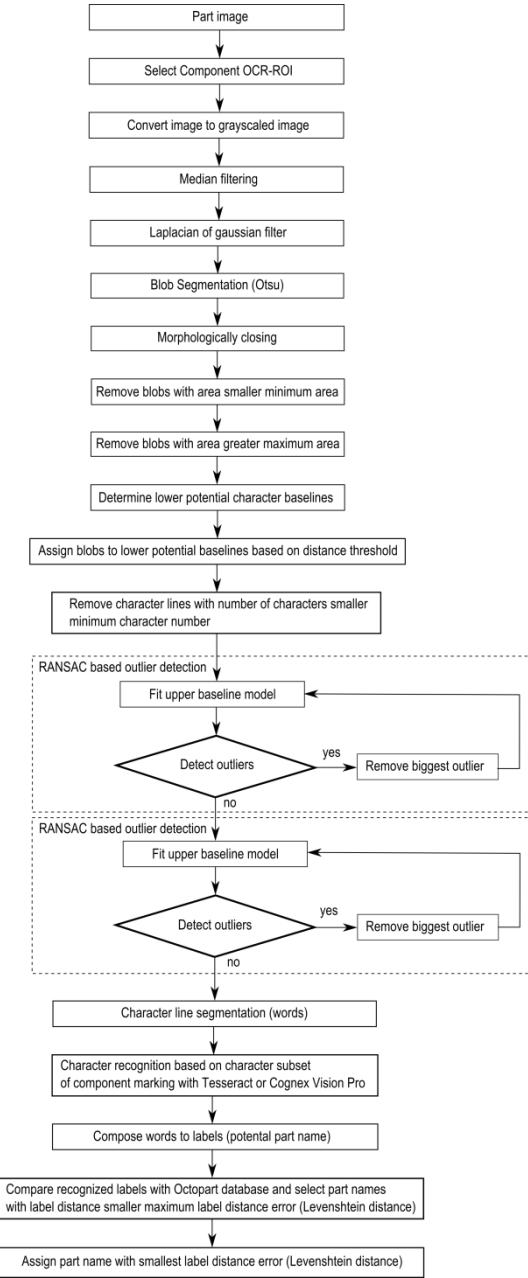
After demining potential character baselines the blobs are assigned to the baselines based on the distance threshold  $distance_{char, potential\ baseline} = 0.25mm$ . All characters that distance from baseline is shorter than  $distance_{char,baseline}$  are assigned to the baseline as potential characters 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 part names usually consist of more than two characters.

To remove blobs that correspond to a baseline but are no characters the RANSAC outlier detection approach is used to estimate baseline models and select all characters that fit the baseline model with a distance error from the baseline smaller  $distance_{char,baseline} = 0.1mm$ . This is done with the lower and upper baseline of the character lines.

- **RANSAC**

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

In the next step the characters which are assigned to baselines are segmented in character lines (words). These words are transferred as an image to the character recognition software Tesseract or 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 5.7.2. The settings and difficulties of the two OCR engines are mentioned in 0.



**Figure 41: 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 comparison between Tesseract 3.0.1 and FineReader10 Corporation Edition from ABBYY shows that there is no significant difference in accuracy between both software engines. The differences in accuracy depend on quality and font of the characters whereas each engine has its advantages and disadvantages (Helinski 2012).

For character recognition with Tesseract, the markings were decomposed in lines referring to the flow chart in Figure 41. The segmented binarized character line images were transferred to 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*

OCRMax<sup>TM</sup> is a font-trainable OCR and OCV (Optical character recognition and Optical character verification) tool from the Cognex VisionPro® software suite for image processing (VisionPro 2014). In this approach the OCR engine OCRMax<sup>TM</sup> 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 part 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/"

- Einstellungen beschreiben

The character level accuracy of both OCR engines depend 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:

$$character\ set = \{0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, /\} \quad (83)$$

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 42. 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.

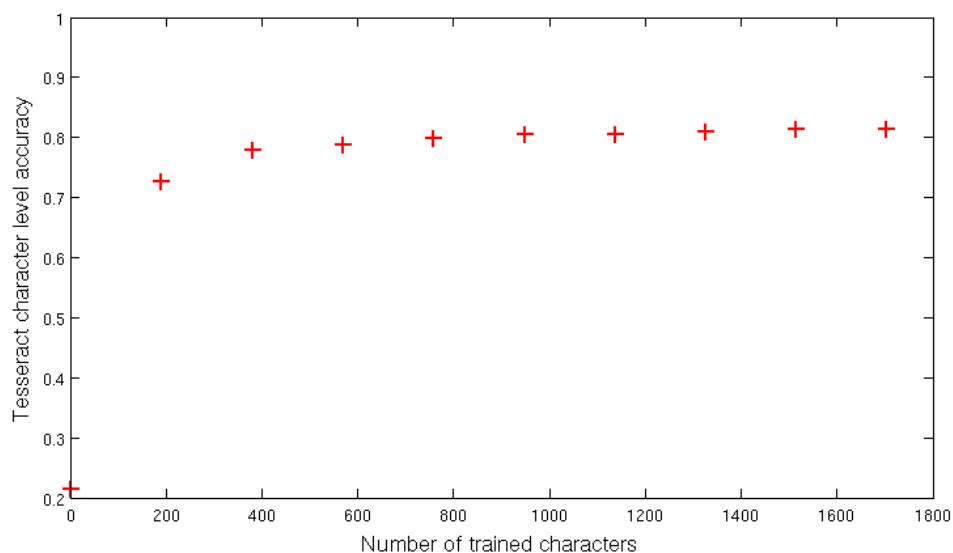


Figure 42: Dependency of Tesseract character recognition accuracy from number of characters

### 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. 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 evaluation on character level with Levenshtein distance is shown in Figure 43.

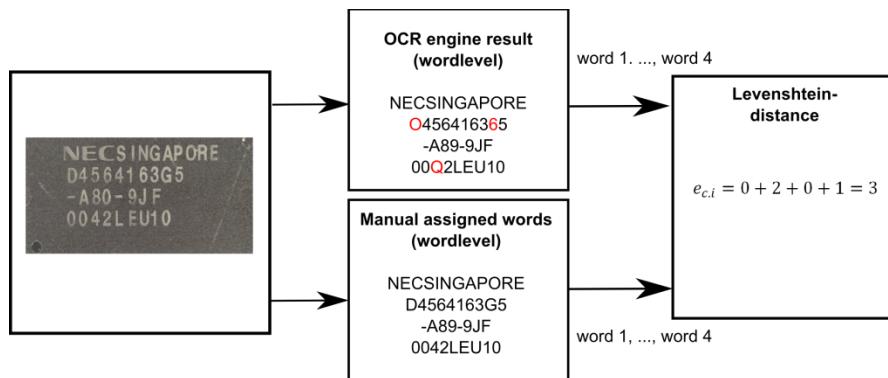


Figure 43: OCR evaluation on character level

#### 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 44.

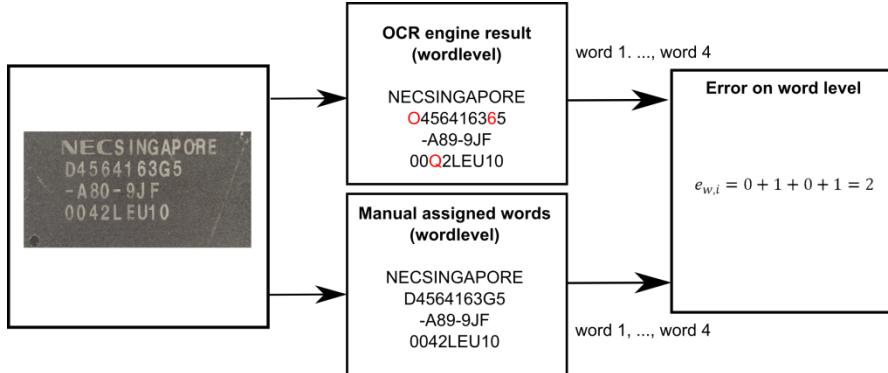


Figure 44: OCR evaluation on word level without Octopart

### *Word level evaluation with Octopart database*

The word level verification with Octopart database is shown in Figure 45. The difference between the OCR evaluations on word level without Octopart database is the assignment of one of the two classes (part-name, non-part-name) to the words. In Figure 45 the part-names are marked in black and the non-part names are marked in red.

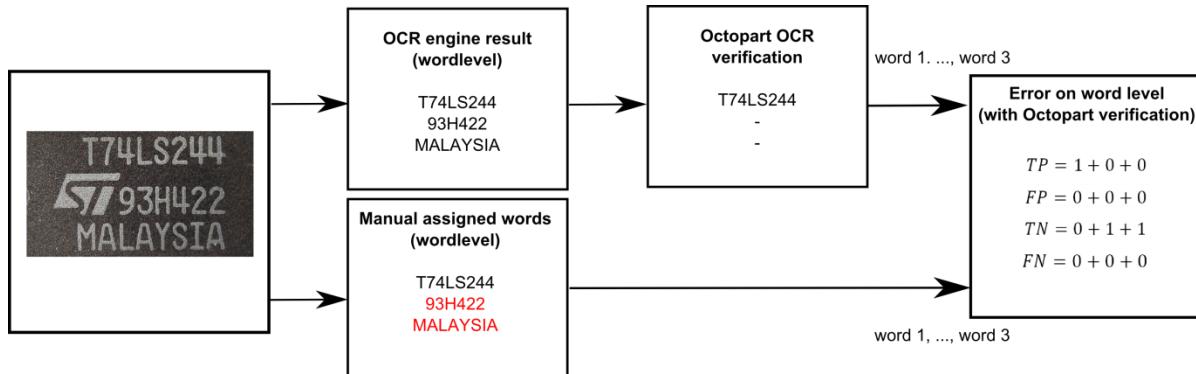


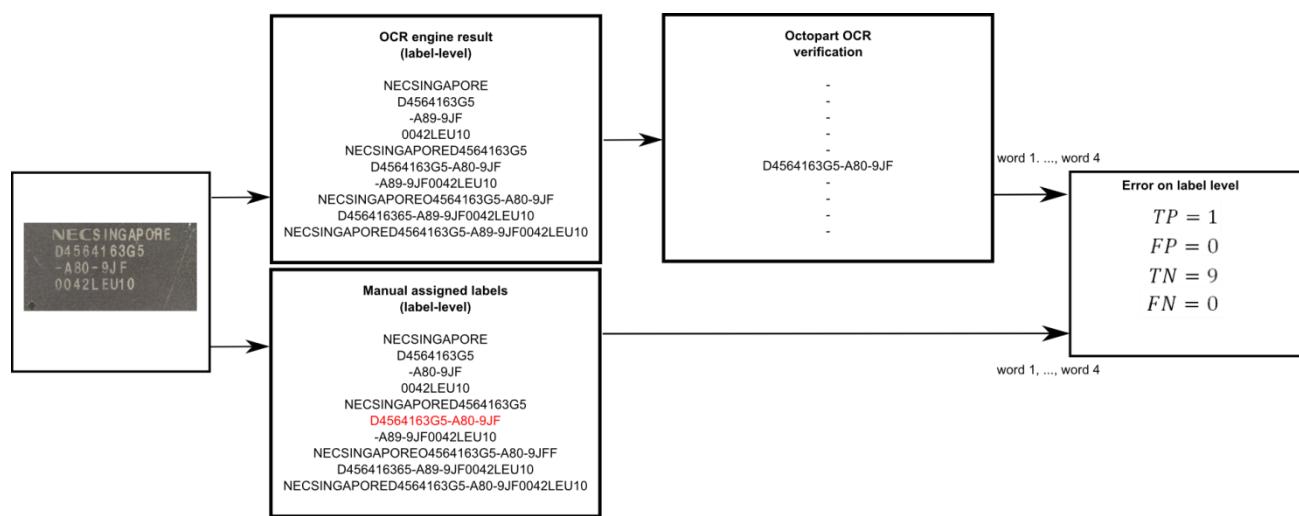
Figure 45: OCR evaluation on word level with Octopart

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

false negative rate (FN) increases by one. If a word was assigned as non-part-name and the Octopart database could not reference the word to a part in the database, the true negative rate (TN) increases by one. If a word was assigned as non-part-name and the Octopart database referenced the word to a part 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 part-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 part-name and the Octopart database response a part-name, the true positive rate (TP) increases by one. If a label was assigned as part-name but Octopart cannot match the word to a part in the database the false negative rate (FN) increases by one. If a word was assigned as non-part-name and the Octopart database could not reference the word to a part in the database, the true negative rate (TN) increases by one. If a word was assigned as non-part-name and the Octopart database referenced the word to a part in the database, the false positive rate (FP) increases by one. An example of OCR evaluation on label level is shown in Figure 46.



**Figure 46: OCR evaluation on label level with Octopart**

## Part level evaluation with Octopart database

The accuracy rate on part-level shows many parts were assigned correctly to a part in the Octopart database whereas the potential part-names were evaluated first at label level. If at least one label (potential part-name) was correctly assigned to a part 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 parts is shown in Figure 47.

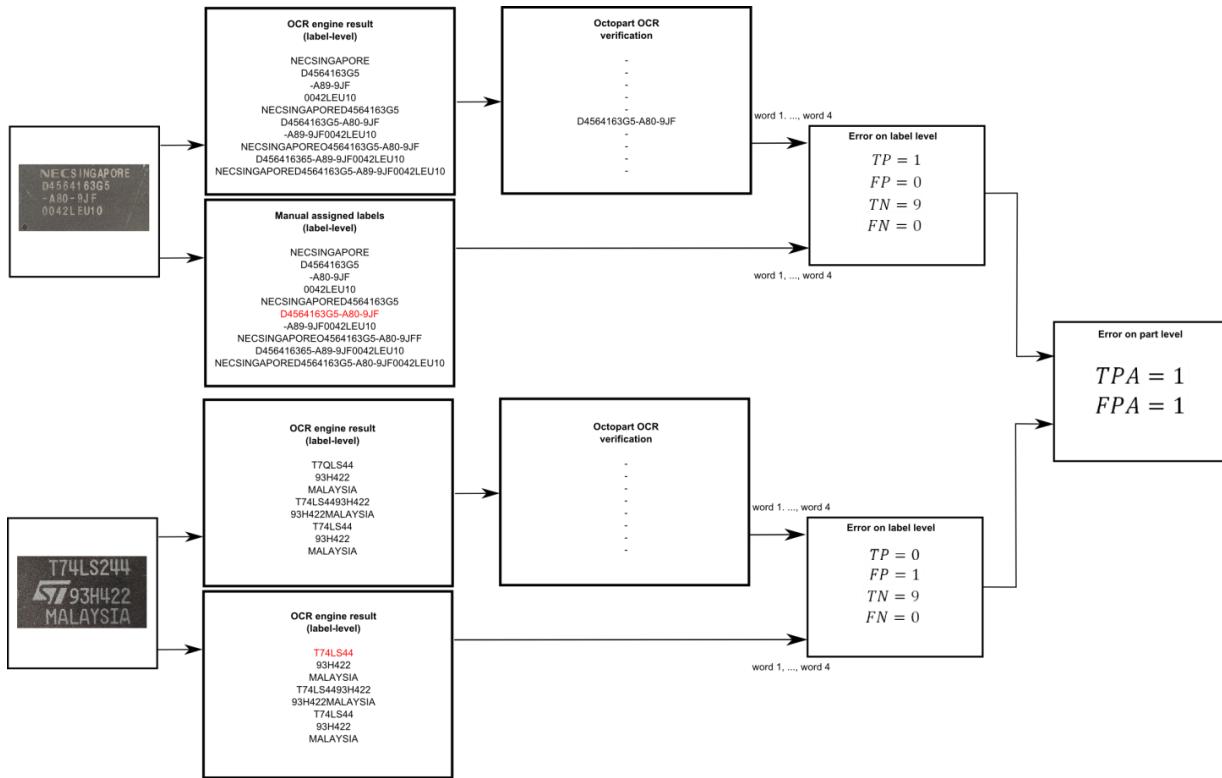


Figure 47: OCR evaluation on part level with Octopart

The OCR evaluation results on character level, word level and part level of the OCR system are shown in detail in chapter 5.7.2.

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

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

In this work an LCI-PCB-model and a PCB-composition-model are automatically created. 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) Detected and classified as unknown component or PCB areas which could not be recognized as PCB support material and therefore are treated as unknown components  $A_{PCB,mounted}$
- 3) Detected, classified as known electronic component (SOT223, Resistor network) but component marking could not be recognized or does not have of a component marking
- 4) Detected, classified and component marking was recognized

The four PCB regions for a sub image of the Arduino Due board are shown in Figure 49. 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.

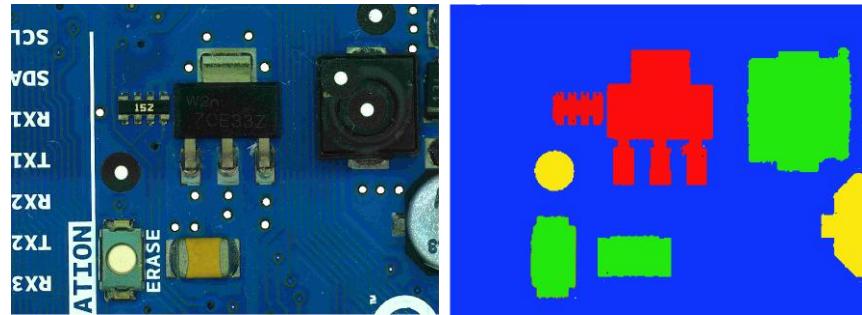


Figure 48: PCB model regions

## 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.

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 imports ILCD data like GaBi or OpenLCA, to analyze the PCB model. The LCI-model in this approach is a generalize 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 and raw material requirements, emissions, solid waste and other releases. The flow diagram for a generalize PCB model is shown in Figure 49. The PCB consists of the four different PCB regions which are modeled as follows:

- 1) Leiterplatte (FR4;2l;2s)
- 2) Printed wiring board HASL 2-layer (subtractive method)
- 3) ILCD component from database
- 4) ILCD component from database

Solder paste is additionally added to the PCB model.

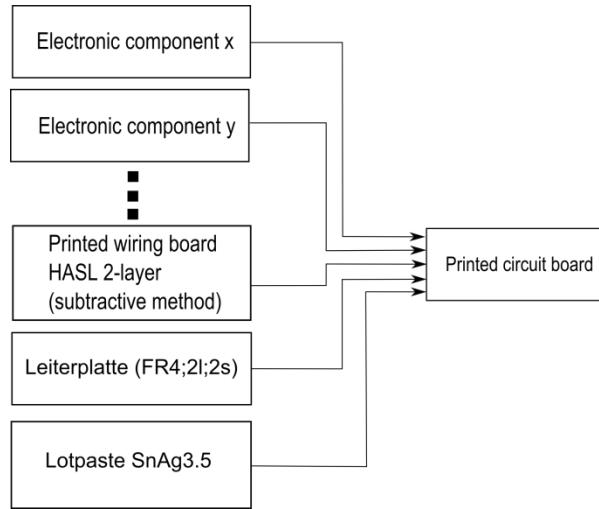


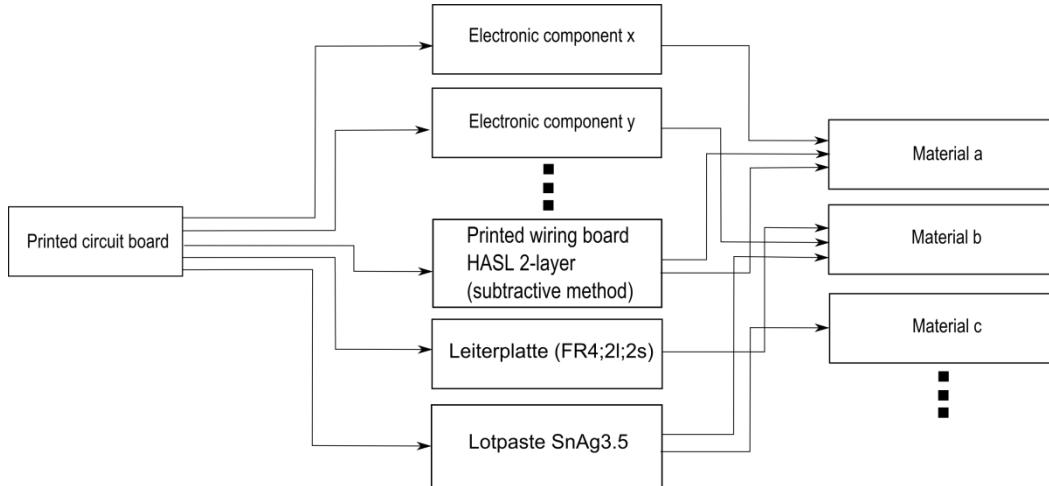
Figure 49: PCB flow diagram for LCI-model

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

Each flow owns flow properties where information are stored about the amount or the component composition materials. An additional flow property containing the buying price of the component was added to the flow properties, if the price could be estimated with the component marking recognition and the Octopart database. The estimation of the buying price can help recyclers to determine components which are valuable for 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 ...). 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 50. The flows in the figure between the PCB components and the materials are symbolic and depend on the content of the components.



**Figure 50: 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 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 where created of the flow properties. The new processes are models of the material composition of the component.

### 1.3 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: leiton-tools-gewichtsberechnung 2014).

$$N_{PCB,surface} = w_{PCB,surface} * A_{PCB,surface} \quad (84)$$

$A_{PCB,surface}$  – Area of PCB support material

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

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 (substractive 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 G.

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

$A_{PCB,mounted}$  – Area of PCB mounted components (unknown components)

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

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 (86).

$$N_{PCB,component,model} = N_{PCB,component} * \frac{m_{PCB,component}}{m_{GaBi,component}} \quad (86)$$

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

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

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

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

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

$$N_{PCB,solder} = w_{PCB,solder} * A_{PCB} \quad (87)$$

$A_{PCB}$  – Area of PCB

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

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

## 1.4 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 therefor is 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 parts containing hazard materials that can be specially treated.

The material composition model and the LCI-model for the Arduino Due board are specified as an example PCB in chapter 5.8.3.

## 5. Implementation and experiments

### 5.1 Implementation

### 5.2 Dataset creation

The dataset consist of 12 electronic components which were analyzed. The components are listed in **Error! Reference source not found.**. The component selection depends on the occurring frequency on the available printed circuit boards. It was taken care that also similar looking components were selected. Therefor the DIP14 component and DIP16 component which differ almost only by number and position of solder joints were selected. In addition the tantalum capacitors of different size but similar appearance were selected. For electronic component recognition, a machine learning application was used whereas multiple representation of the component must be created to analyze representative features. The component representations are taken from different parts of a component and different printed circuit boards to create a representative dataset. The available printed circuit boards are seen in .

To detect the edges of the part border, border pixels are also selected from the printed circuit board images as can be seen in Figure 51. Additional important information and properties of the component are listed in Table 6.

**Table 6: Component properties**

<b>Component properties</b>	<b>Description</b>
<b>Package properties</b>	
Component length	
Component width	
Component border size	
Package DOF	
<b>OCR properties</b>	
ROI for optical character recognition	
Subset of characters for optical character recognition	
Maximum and minimum number of OCR lines	
<b>Frequency features generation properties</b>	
Image scale for frequency feature generation	
Number of Fourier coefficient features	
Border cut information	
<b>Color histogram features</b>	
Image scale for histogram feature generation	
<b>Segment features</b>	
Image scale for histogram feature generation	
Number of initial seed points for region growing approach	
<b>PCA reconstruction features</b>	
Image scale for histogram feature generation	
Kernel size for LoG (Laplacian of	

Gaussian) edge detection	
Number of PCs	
<b>LCI properties</b>	
ILCD-model full aggregated model	
ILCD-model composition model	

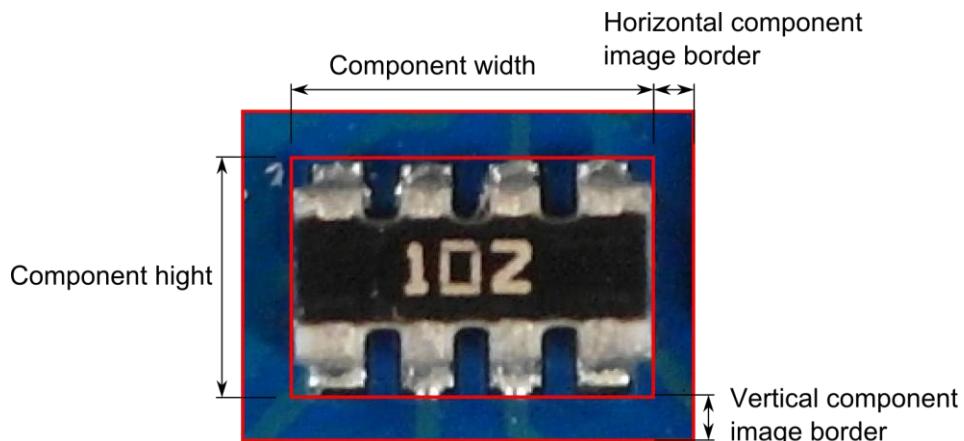


Figure 51: Component border definition

A section of the component database is shown in Figure 52.

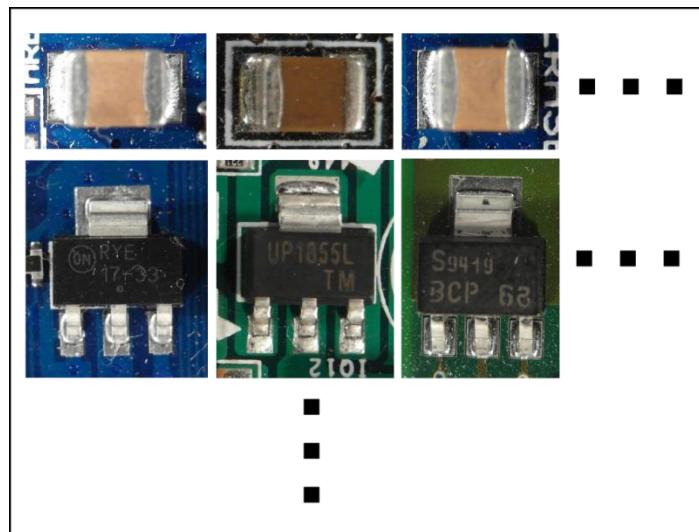


Figure 52: Database section

### 5.2.1 Image acquisition

The image acquisition was done with a Samsung EX2F camera and a working distance in a range from 20 mm to 120 mm through the Object. Autofocusing was used to get sharp images. The working distance was adapted to the size of the component in which the distance was decreased for smaller components and increased for bigger components. For illumination a bright-field incident illumination was selected because it generates a uniformly bright, well-contrasted image (Imaging 2012). The lighting sources consist of four DSL-1110 table lamps with diffusion film to generate a uniformly bright and diffuse illumination. The image acquisition system is seen in Figure 53.

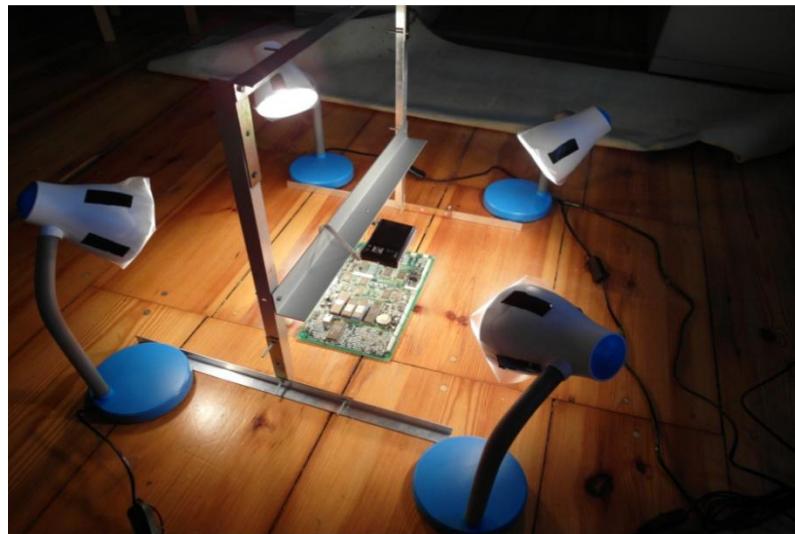


Figure 53: Image acquisition system

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

- Tiefenschärfe
- Telezentrisches Objektiv

### 5.2.2 Dataset composition

The dataset used in the experiments consist of 2000 parts from 12 component classes. The dataset composition is shown in Table 7.

**Table 7: Dataset composition**

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

TO263			
Quartz HC-49/S			

### 5.3 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 each image from the PCB surface detection test set were classified based on a 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 surface images and 1287 were non-surface images. The distinction between surface-image and non-surface image was determined manually and could not be determined clearly for each PCB surface representation and therefore influences the method in a negative way. 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 54 is shown in Figure 55. It can be seen that the PCB surface pixel have much higher score values than others.

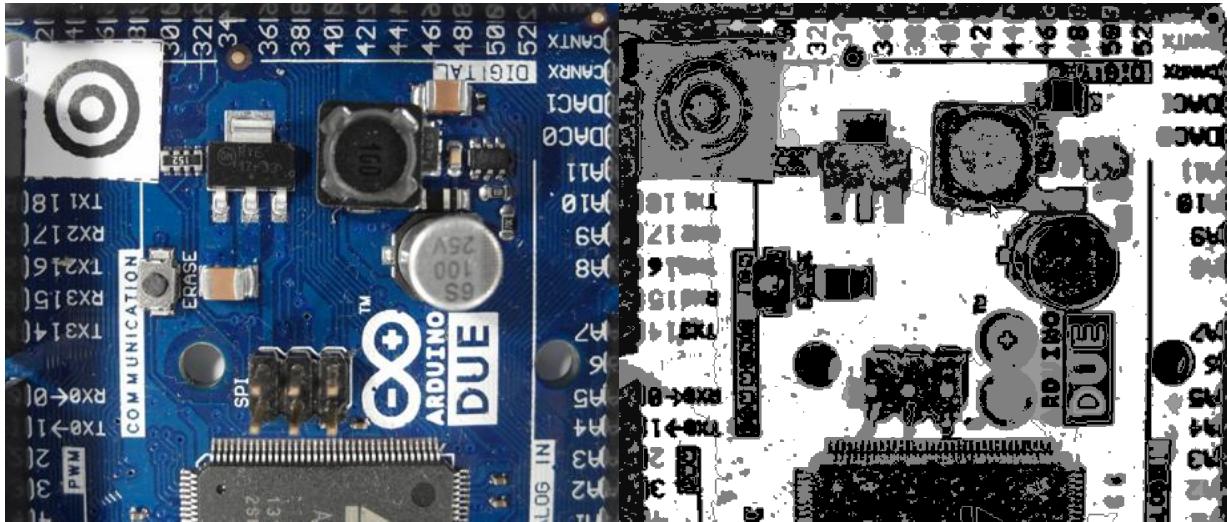


Figure 54: original PCB image

Figure 55: Sum of RBF-kernel SVM scores  $w(x,y)$   
(grayvalues are scaled between -20 and 20)

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

## 5.4 Feature selection results

The out-of-bag error depends on the number of random forest trees. The oob-error depending on the number of trees for 3136 FFT features extracted from the Resistor network 1206 component was computed. The red graph shows the out-of-bag error from the two step feature selection (FS+FR), the blue one the out-of-bag error from the random forest feature selection (RF) and the green one the out-of-bag error from fisher score (FS) feature selection with 235 selected features. The graphs show that the error rate of the FS+RF feature selection approach decreases faster and becomes smaller compared to the others whereas the oob-error does not show a big difference between the algorithm what indicates that the samples tend to be well linearly separable.

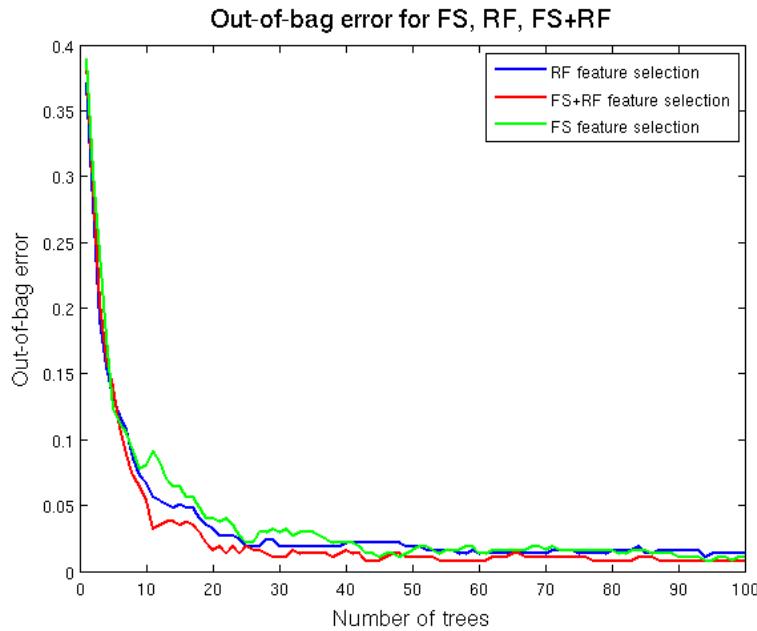


Figure 56: A comparison of different feature selection approaches

In this approach the feature selection algorithm based on Fisher score and Random forest described in **Error! Reference source not found.** was used to select a subset of important features for classification. The most important features depend on the component therefore feature selection was applied to each component dataset. A list of the most important features from all examined components is shown in Appendix A. The numbers in the table correspond to the feature numbers of the associated feature set. The assignment of the features numbers to the features is specified in .

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

#### 5.4.1 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 57.

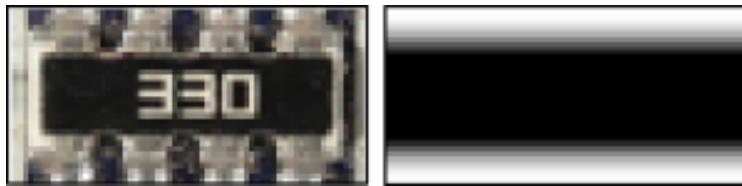


Figure 57: Resistor network 1206 and the most significant real part elementary image

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

#### 5.4.2 Color features

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

- Bild + erläuterung

#### 5.4.3 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  $y=1.7\text{mm}$ ,  $x=0.26\text{mm}$ . 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.

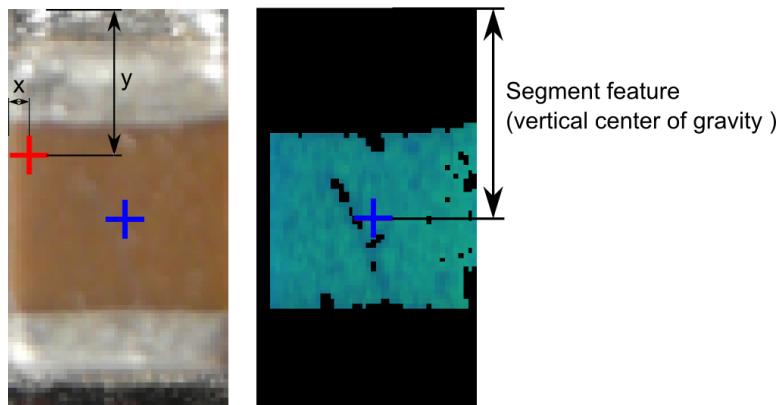


Figure 58: Most important segment and seed point from ceramic capacitor

#### 5.4.4 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 part. The rounded border reflects the light almost independent from the beam angle of the illumination. That forms a bright shiny circle that is striking in the Laplacian of Gaussian (LoG) filtered images and can be efficiently be compressed into the part image PCs. A LoG filtered edge image of the SMD Aluminum electrolytic capacitor and the unit matrix projection into the PCs is shown in Figure 59.

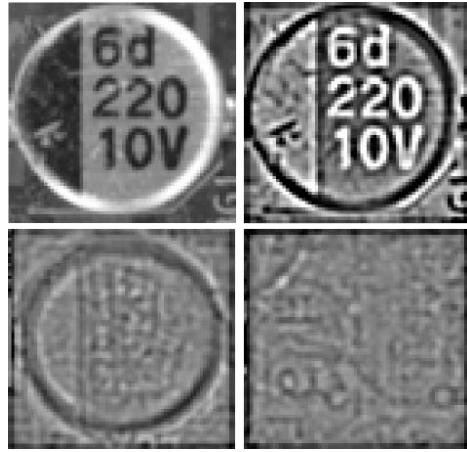


Figure 59: SMD Electrolyte capacitor (top, left), SMD Electrolyte capacitor edge image (top, right), unit matrix projection into component PCs (bottom, left), unit matrix projection into non-component PCs (bottom, right)

#### 5.4.5 Dependence of classification accuracy from number of selected features

### 5.5 Classification results

- bbox
- continue
- plot number of features
- plot number of trees

#### 5.5.1 Random forest classifier results

Five random forest classifiers were trained whereas the first to fourth are based on the four selected feature sets from Frequency features, Color features, Segment features and PCA-reconstruction features which were extracted from the four feature extraction algorithms described to chapter 2.1. The accuracy of the random forest classifier from the mean accuracies over all fifteen components is shown in Table 10. A detailed breakdown can be found in Appendix C.

**Table 10: Random forest classification results**

		Frequency features	Color features	Segment features	PCA reconstruction features	Features selection from all feature sets
<b>Average recognition accuracy of all Components</b>	True positive	1909/1971 (96.9%)	1918/1961 (97.6%)	1565/1655 (94.6%)	1560/1656 (94.2%)	1953/1971 (99.1%)
	True negative	1956/1982 (98.7%)	1872/1984 (94.4%)	1551/1735 (89.4%)	1589/1694 (93.8%)	1968/1982 (99.3%)

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

### 5.5.2 Support vector machine classifier results

Sdf

### 5.5.3 Multiclass classification result

Dfg

## 5.6 Decision level fusion results with Dempster-Shafer

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. To compare all components, the PCI-slot bounding box has to be cropped from an image of a Resistor 0806 at the same position. With this camera configuration it is not possible to make images with a large filed 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 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 samples from all other classes (including samples from undefined classes) 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 (68) 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.

## 5.7 Optical character recognition results

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

- continue

### 5.7.1 Optical character recognition dataset and limits

The optical character recognition dataset consists of 85 ICs which were acquired with an image resolution of 100 *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

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

### 5.7.2 Optical character recognition accuracy result on character level, word level, label level and part level

To evaluate the optical character recognition results the labeled component markings and the recognized marking with OCR software are compared on different accuracy levels (Helinski 2012).

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

$$A_{c,i} = 1 - \frac{e_{c,i}}{c_i} \quad (88)$$

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:

$$A_c = 1 - \frac{\sum_{i=1}^{85} e_{c,i}}{\sum_{i=1}^{85} c_i} \quad (89)$$

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

$$A_{w,i} = 1 - \frac{e_{w,i}}{w_i} \quad (90)$$

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:

$$A_w = 1 - \frac{\sum_{i=1}^{85} e_{w,i}}{\sum_{i=1}^{85} w_i} \quad (91)$$

The accuracy results of the OCR engines Tesseract and OCRMax on all accuracy levels is shown in Table 11.

**Table 11: 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%)

The label level accuracy was not studied because of the high number of non-part 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 part-name database. An investigation of the accuracy on label level with the Octopart database is done in chapter 5.7.3.

### 5.7.3 Octopart based part name assignment

The online electronic component database Octopart gives the opportunity to verify recognized part 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 and the results were analyzed according to the method in chapter **Error! Reference source not found.**. One of the classes “part-name” and “non-part-name” is assigned to each of the words and labels. The analyzed results were evaluated according to the assignment. The words/labels that are part-names and are assigned to the right part in the Octopart database are true positive labeled results. Words/labels that are non-part-names like manufacturer names, country of manufacture, production numbers etc. and which are not assigned to parts in the Octopart database are labeled as true negative. Words/labels that are not part names but assigned to parts in the Octopart database are labeled as false positive. Words/labels that are parts but are not assigned to parts in the Octopart database or assigned to wrong parts in the database are labeled as false negative. The confusion matrix for the manual labeled words and labels are shown in Table 12.

Table 12: Confusion matrix of the manual labeled words (word-level) verified with Octopart database

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

The confusion matrix for the manual labeled words and labels is shown in Table 13.

**Table 13: Confusion matrix of the manual labeled labels (label-level) verified with Octopart database**

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

The accuracy rate on part-level in Table 14 shows how many parts were assigned to a part in the Octopart database whereas the part names were manual labeled and verified with Octopart database on word level.

**Table 14: Accuracy rate of part assignment with manual labeled parts 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 15 shows how many parts were assigned to a part in the Octopart database whereas the part names were manual labeled and verified with Octopart database on label level.

**Table 15: 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 14 and Table 15 show that the analyses of the part names on label level increases a little the accuracy rate on part level compared to the analyses on word level.

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

**Table 16: Confusion matrix of the Tesseract recognized words (word-level) verified with Octopart database**

	Condition: part name	Condition: non-part name
--	----------------------	--------------------------

Test outcome: part name	31/73 (42.2%)	9/161 (5.6%)
Test outcome: non-part 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 13.

**Table 17: Confusion matrix of the Tesseract recognized labels (label-level) verified with Octopart database**

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

The accuracy rate on part level is shown in Table 14.

**Table 18: 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%)

**Table 19: Accuracy rate of part assignment with Tesseract OCR engine on label level verified with Octopart database (part-level)**

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

The confusion matrices for the recognized part markings with the OCR engine OCRMax on word-level is shown in Table 20.

**Table 20: Confusion matrix of the OCRMax recognized words (word-level) verified with Octopart database**

	Condition: part name	Condition: non-part name
Test outcome: part name	44/73 (60.3%)	13/161 (8.1%)
Test outcome: non-part 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 20.

**Table 21: Confusion matrix of the OCRMax recognized labels (label-level) verified with Octopart database**

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

The accuracy rate on part level with OCR engine OCRMax is shown in Table 22.

**Table 22: 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%)

**Table 23: 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%)

#### 5.7.4 Octopart based part price assignment

To evaluate the economic sustainability of reuse of electronic part it is necessary to estimate the economic value of recognized parts. One indicator of valuable parts is the original price of the part. The Octopart database gives the possibility to request the price for a part if the part could be assigned to a part 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:

$$A_{price} = \frac{\#parts_{price}}{\#parts_{assigned}} = \frac{30}{59} = 0.509 (50.9\%) \quad (92)$$

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

To estimate the reuse potential of electronic components a critical economic value for the parts can be estimated which is a balance between the costs of desoldering the part and the costs of quality check one hand and economic value of a part 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 (Irina Stobbe, Hansjörg Griese 2002).

The maximum value of 2.3 Euro was used to estimate the critical price rate which was calculated as follows:

$$A_{price,critical} = \frac{\#parts_{price,critical}}{\#parts_{assigned}} = \frac{10}{59} = 0.17 (17.0\%) \quad (93)$$

where  $\#parts_{price,critical}$  is the number of parts where a price could be estimated and the price was greater than 2.5 Euro and  $\#parts_{assigned}$  is the number of parts that could be assigned to a part in the Octopart database. The critical price rate shows that for around 17% of the assigned parts a price could be estimated which is greater than the critical price of 2.5 Euro based on the Octopart database. A detailed discussion about the reuse potential is made in chapter 0.

## **5.8 Life-cycle inventory analyses results**

sdf

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

Saf

## **5.8.2 Recycling and reuse potential of electronic components**

### *Increase of tantalum concentration in scrap by selective dismantling*

Tantalum is one of the Rare earth materials (REE) 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. (Perine Chancerel, vera Susanne Rotter 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 it 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 (Perine Chancerel, vera Susanne Rotter 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 (<http://tantalumrecycling.com/> 2015).

## **5.8.3 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 example reference due to the fact that an open-source eagle layout already exist and a part list can be easily exported in the eagle software.

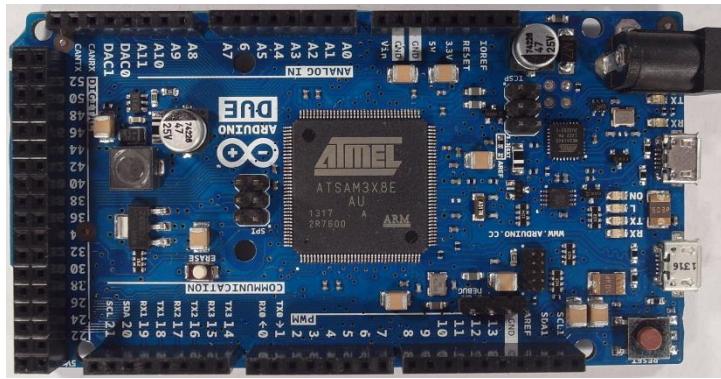


Figure 60: Arduino Due board

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

Each part 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 parts that could not be assigned with an associated part 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 .

The resulting process model input parts for the Arduino Due model are shown in Table 24.

Table 24: 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 parts is shown in Figure 61.

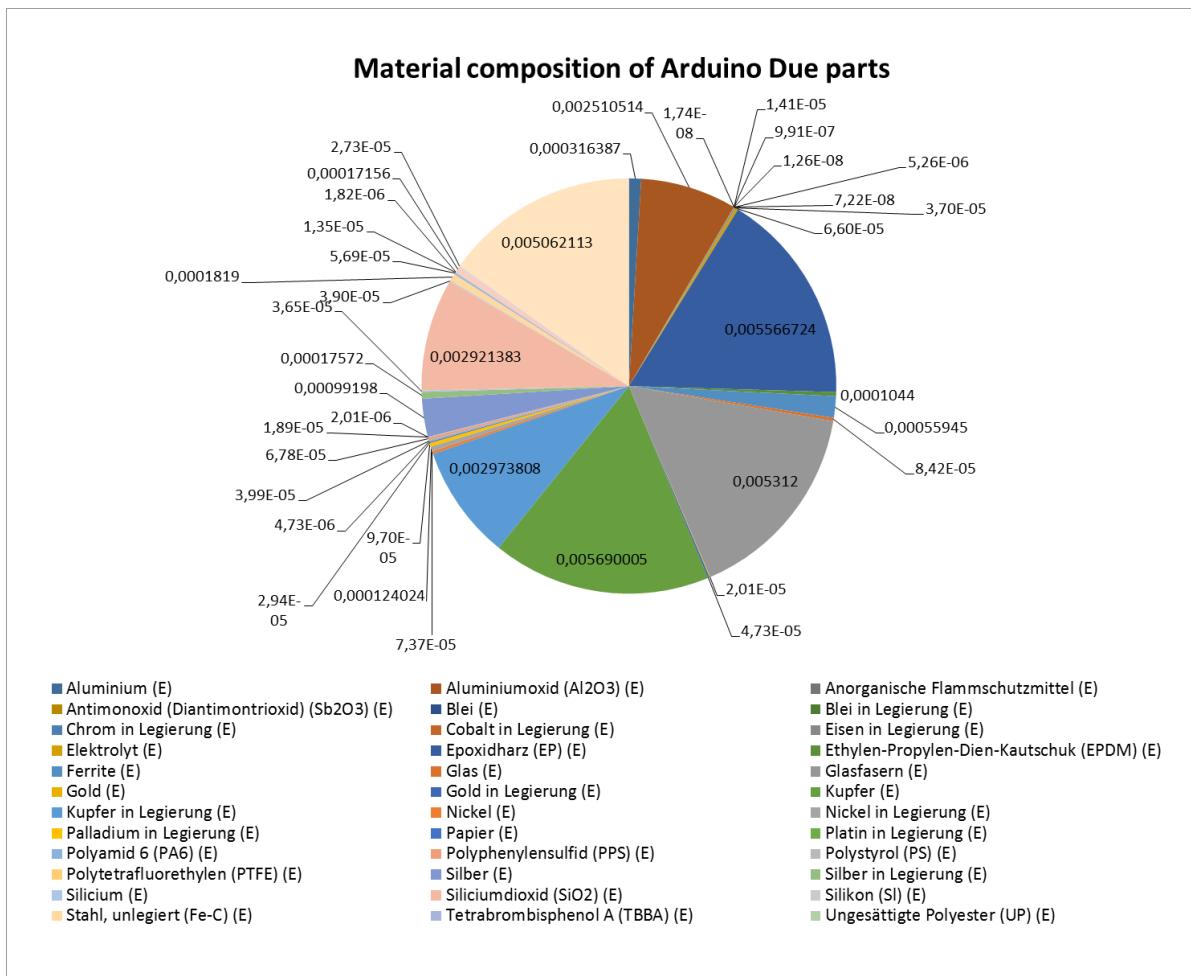


Figure 61: Estimated material composition of Arduino Due parts [kg]

The estimated material prices of the materials contained in the Arduino Due parts are shown in Figure 62.

Not all material prices

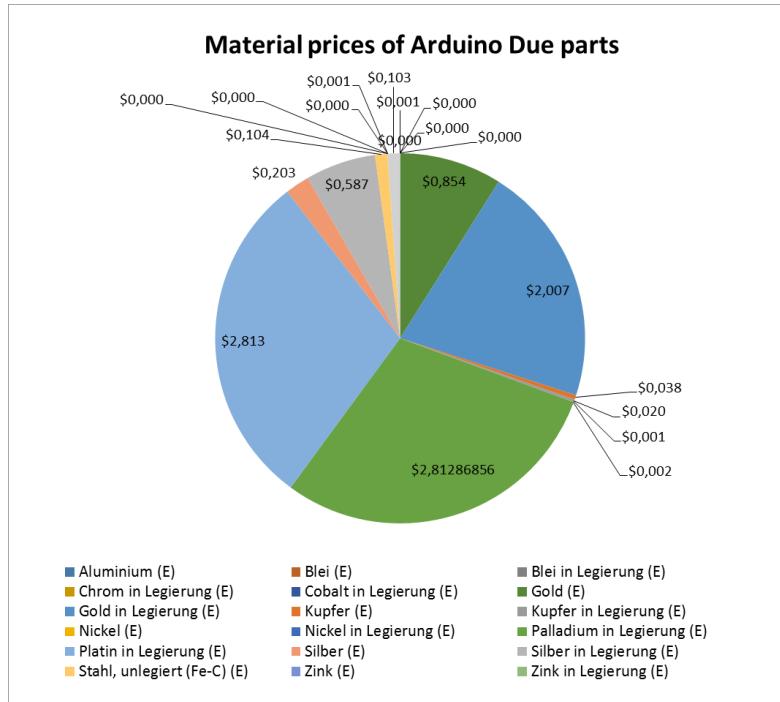


Figure 62: Estimated material prices of Arduino Due parts

Most of the precious metals are valuable materials in PCBs which can be economically recycled. 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 parts of the Arduino Due board is shown Figure 63.

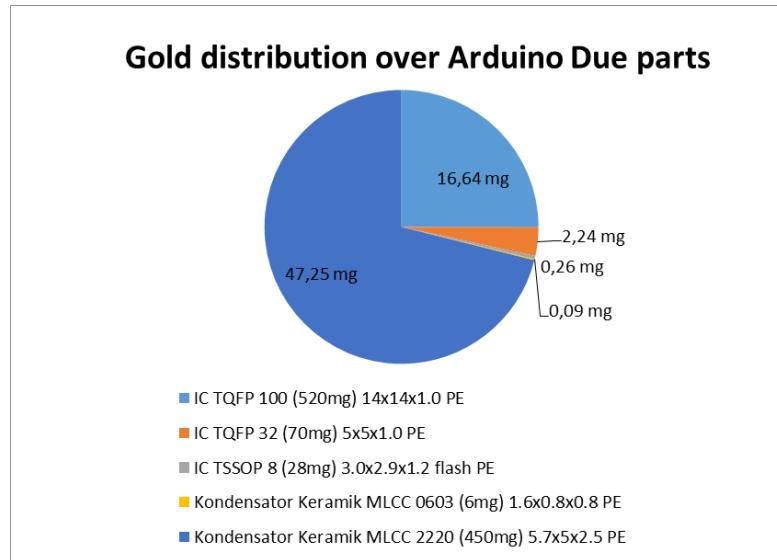


Figure 63: Estimated Gold distribution over Arduino Due parts

Another valuable and recyclable precious metal is palladium. The estimated palladium distribution over the parts of the Arduino Due is shown in Figure 64.

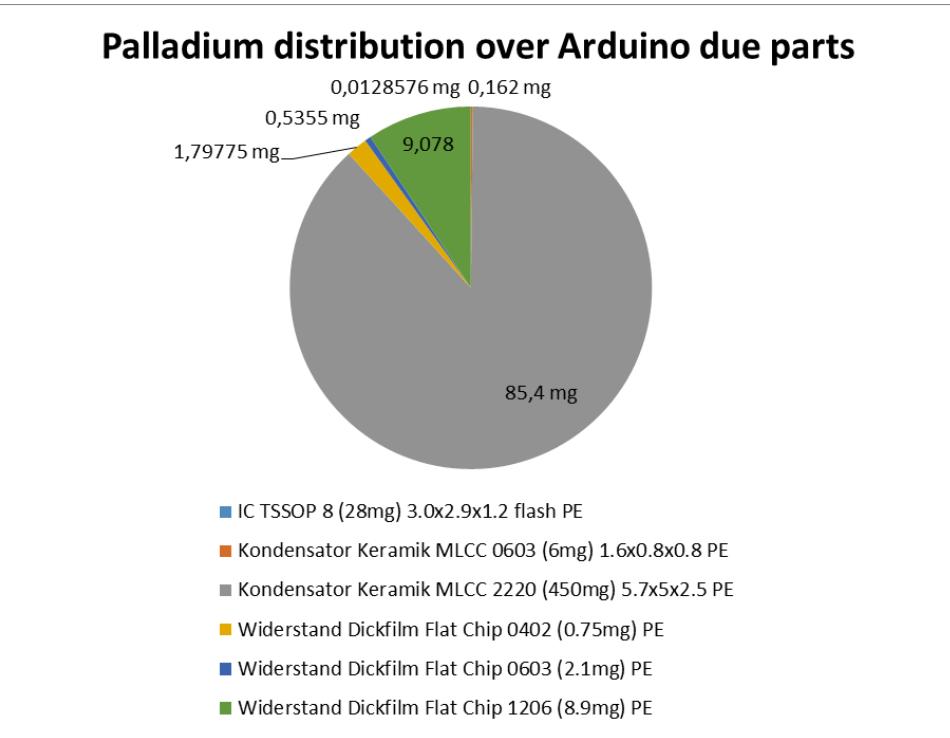


Figure 64: estimated Palladium distribution over Arduino Due parts

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 parts for reuse, the component prices are a strong indicator for reusability. The estimated prices of the Arduino Due part with a price greater 0.1€ are shown in Figure 65 the price of the rest of the parts are summarized to Rest.

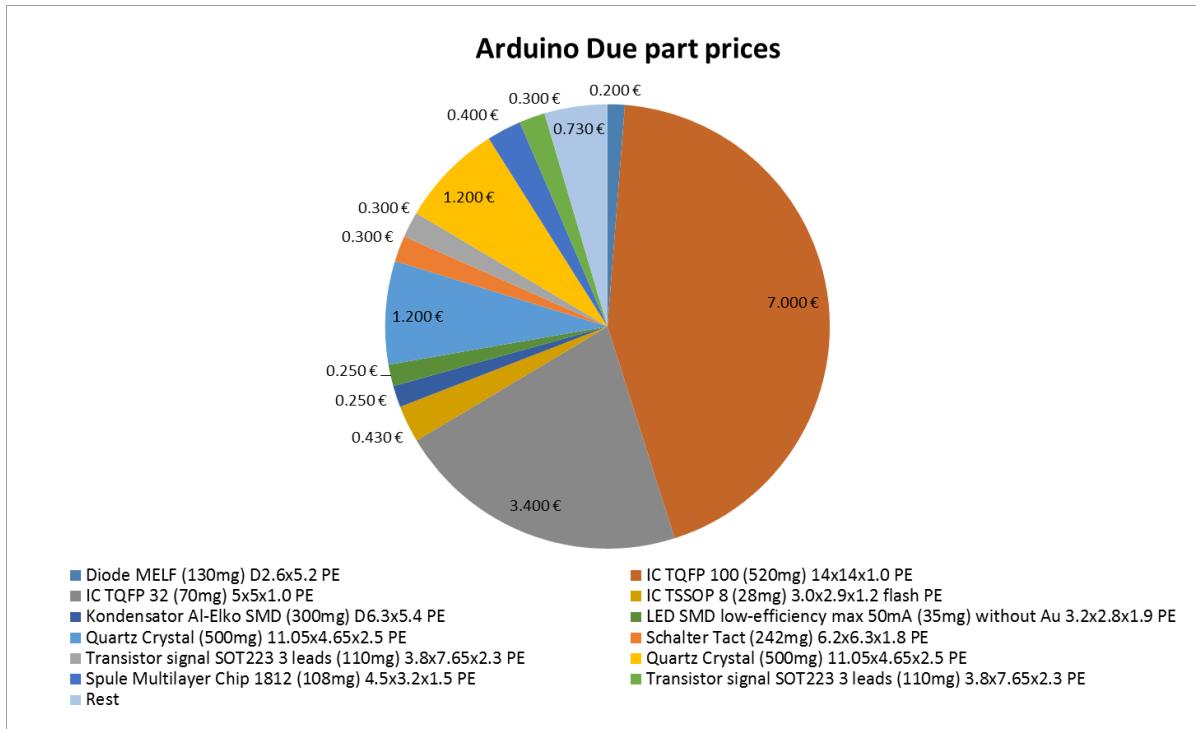


Figure 65: Estimated Arduino Due part prices

- What is arduino Due board
- - 17 No, 16 Yes
- - palladium MLCC article

## 6. Discussion and future work

### 6.1 Electronic component detection

#### 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 the segmentation plane. 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 66. The segmented components can be used to determine the centroid of the component bounding box and us the coordinate as input for the 2D component classification.

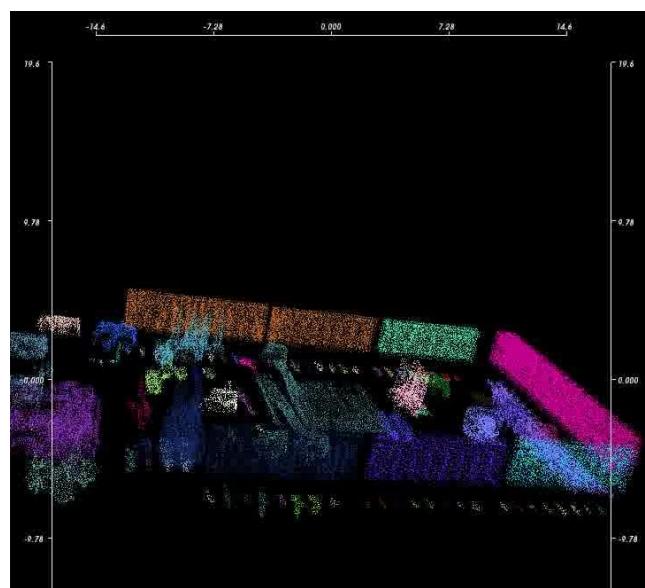


Figure 66: 3D model based component detection

### 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 (Torsten Koch 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 67. 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  $\lambda$  and the known angle  $\gamma$  the height  $h(x)$  can be calculated by

$$h(x) = \frac{\lambda(x)}{\tan(\gamma)} \quad (94)$$

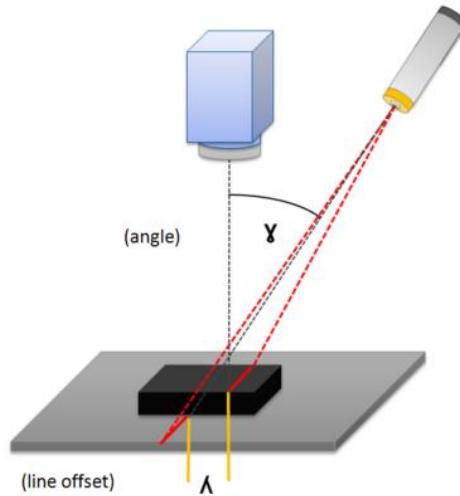


Figure 67: Principle of laser triangulation (Torsten Koch 2013)

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

## 7. Conclusion

- Langer Abschnitt!

## 7.1 Application inclusion in the PCB recycling process chain

The PCB recycling chain in chapter 1.2 was improved with the inclusion of the electronic component recognition application. The improved recycling chain is shown in Figure 70. The improved PCB recycling process chain consists of three steps, the collection of WEEE, the preprocessing and the recovery and disposal. Influenced by the component recognition system are mainly the preprocessing step and the Recovery and disposal step.

In the preprocessing step the manual dismantling has to be changed. In many todays recycling chains the manual dismantling refers to the WEEE products with hazard substances like refrigerators or big capacitors. The whole electronic device is shredded including PCBs which are damaged or destroyed. 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 component 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 resolded 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 components are collected separated according to the substances which increase the concentration of special substances.

## 2- Pre-processing

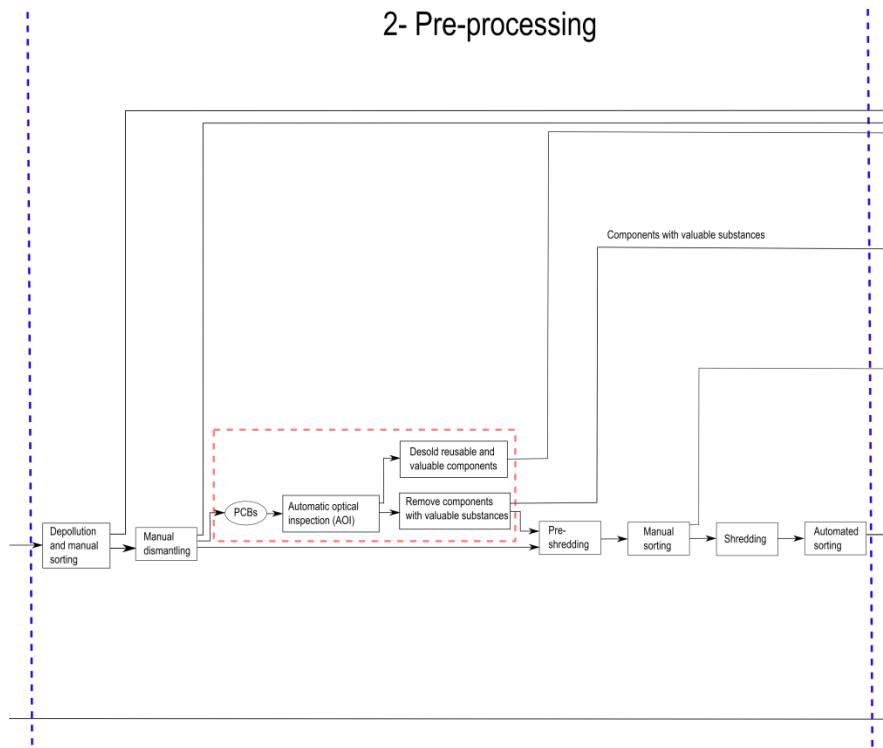


Figure 68: Improved pre-processing step in PCB recycling process chain

In the recovery and disposal step, the components which are tested successfully are prepared for the reuse. The components are cleaned and packed in for sale. Unfortunately the marked for reused electronic components from consumer electronics is still a small marked today, but with height grow potential.

Components with valuable substances which were 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 profitable 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 69.

### 3- Recovery and disposal

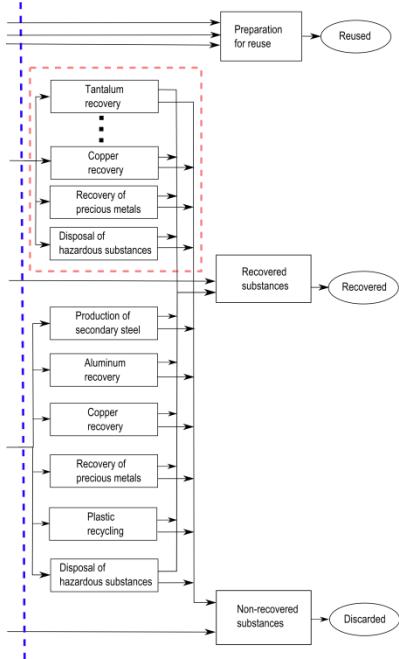


Figure 69: Improved recover and disposal step in PCB recycling process chain

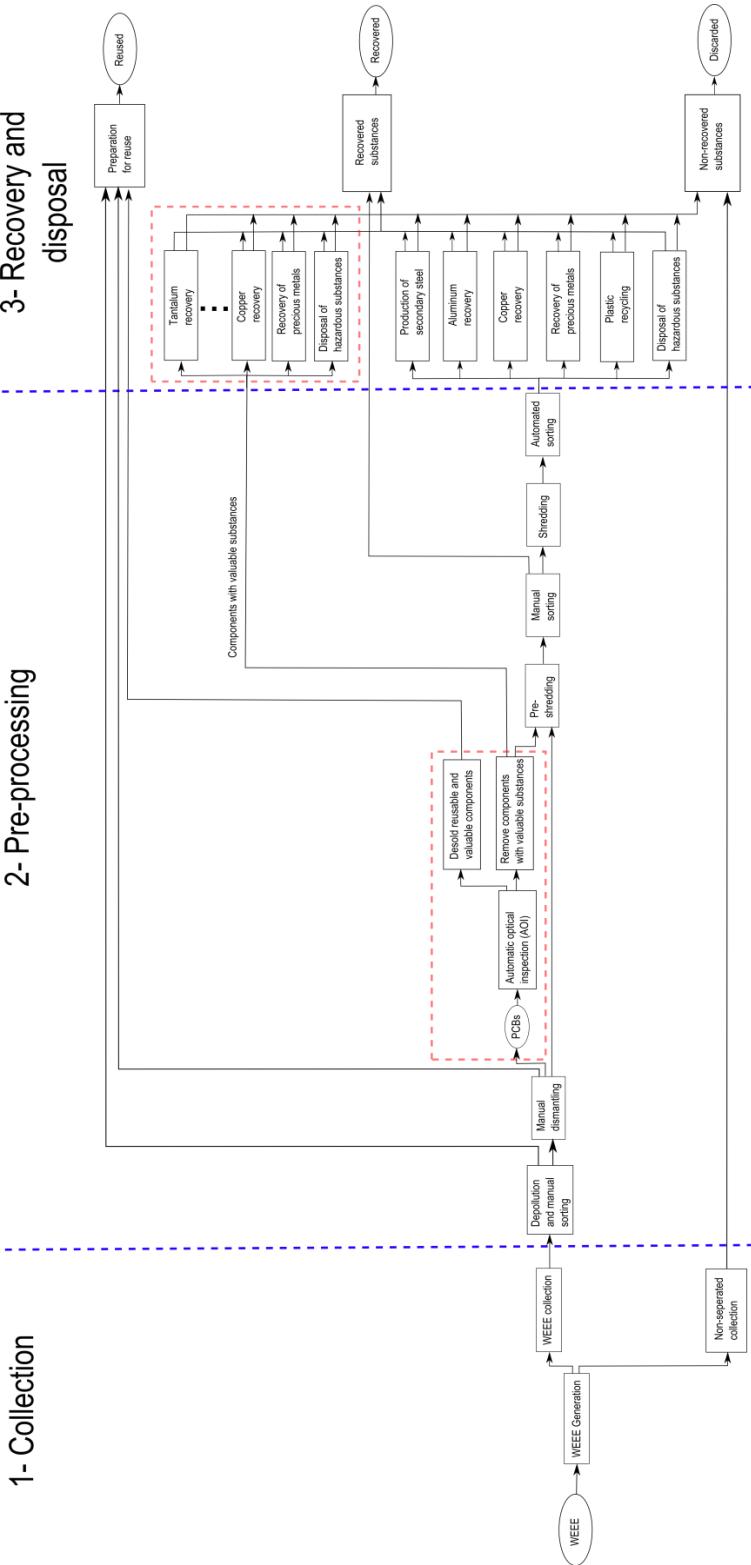
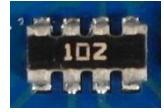


Figure 70: Improved PCB recycling process chain

## Appendix A Recognition database components

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

<p style="text-align: center;"><b>DIP16</b></p> <ul style="list-style-type: none"> <li>- Package: DIP14</li> <li>- 19.5 mm x 7.6 mm</li> </ul>	
<p style="text-align: center;"><b>SMD Resistor 1206</b></p> <ul style="list-style-type: none"> <li>- Imperial code: 1206 (3216 metric)</li> <li>- 3.2 mm x 1.6 mm</li> </ul>	
<p style="text-align: center;"><b>SOIC-8</b></p> <ul style="list-style-type: none"> <li>- Package: SOIC8</li> <li>- 5.0 mm x 6.2 mm</li> </ul>	
<p style="text-align: center;"><b>Ceramic capacitor 1210</b></p> <ul style="list-style-type: none"> <li>- Imperial code: 1210 (3225 metric)</li> <li>- 3.2 mm x 2.5 mm</li> <li>- Color: brown/orange</li> </ul>	
<p style="text-align: center;"><b>SOT223-3</b></p> <ul style="list-style-type: none"> <li>- Package: SOT223-3</li> <li>- 6.5 mm x 7.0 mm</li> </ul>	
<p style="text-align: center;"><b>SMD Resistor 0806</b></p> <ul style="list-style-type: none"> <li>- Imperial code: 0806 (2012 metric)</li> <li>- 2.0 mm x 1.2 mm</li> </ul>	

<p><b>TO 263</b></p> <ul style="list-style-type: none"> <li>- Imperial code: 0806 (2012 metric)</li> <li>- 10.1 mm x 15.0 mm</li> </ul>	
<p><b>Quartz HC-49/S</b></p> <ul style="list-style-type: none"> <li>- Package: HC-49/S-3</li> <li>- 4.7 mm x 11.0 mm</li> </ul>	
<p><b>PCI</b></p> <ul style="list-style-type: none"> <li>- 32-bit PCI slot</li> <li>- 9.0 mm x 85.0 mm</li> </ul>	

## Appendix B      Most important selected features

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

## Appendix C Random forest classification results

		Frequency features	Color features	Segment features	PCA reconstruction feature	Features selection from all feature sets
<b>Tantalum capacitor</b>	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%)
<b>SMD Aluminum electrolytic capacitor</b>	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%)
<b>QFP100</b>	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%)
<b>SMD Resistor Network array 1206, 4 Resistors</b>	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 (100%)	255/266 (95.9%)	204/227 (89.9%)	223/227 (98.2%)	265/266 (99.6%)
<b>SMD Transistor SOT23-3</b>	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%)
<b>DIP14</b>	True positive	111/114 (97.4%)	104/114 (91.2%)	91/99 (91.9%)	95/99 (96.0%)	112/114 (98.2%)
	True negative	112/114 (98.2%)	105/116 (92.1%)	85/95 (89.5%)	91/95 (95.8%)	112/114 (98.2%)
<b>DIP16</b>	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%)
<b>SMD Resistor 1206</b>	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%)

<b>SOIC-8</b>	True positive	104/106 (98.1%)	105/106 (99.1%)	83/87 (94.3%)	87/88 (98.9%)	104/106 (98.1%)
	True negative	105/106 (99.0%)	96/106 (90.6%)	76/92 (82.6%)	92/92 (100%)	106/106 (100%)
<b>Ceramic capacitor 1210</b>	True positive	36/42 (85.7%)	41/42 (97.6%)	35/36 (97.2%)	24/36 (66.7%)	41/42 (97.6%)
	True negative	34/42 (81.0%)	38/42 (90.5%)	34/35 (97.1%)	24/65 (68.6%)	41/42 (97.6%)
<b>SOT223-3</b>	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%)
<b>SMD Resistor 0806</b>	True positive	294/308 (95.4%)	302/308 (98.1%)	249/258 (96.5%)	242/258 (93.8%)	306/308 (99.4%)
	True negative	306/308 (99.4%)	290/308 (94.2%)	252/266 (94.7%)	247/268 (92.9%)	303/308 (98.4%)
<b>TO263</b>	True positive	36/36 (100%)	32/26 (88.9%)	26/29 (89.7%)	28/29 (96.6%)	34/36 (94.4%)
	True negative	36/36 (100%)	30/36 (83.3%)	29/32 (90.6%)	31/32 (96.8%)	35/36 (97.2%)
<b>Quartz HC-49/S</b>	True positive	36/46 (93.5%)	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%)
<b>32-bit-PCI slot</b>	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 D Linear-SVM classification results

		Frequency features	Color features	Segment features	PCA reconstruction feature	Features selection from all feature sets
<b>Tantalum capacitor</b>	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%)
<b>SMD Aluminum electrolytic capacitor</b>	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%)
<b>QFP100</b>	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%)
<b>SMD Resistor Network array 1206, 4 Resistors</b>	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%)
<b>SMD Transistor SOT23-3</b>	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%)
<b>DIP14</b>	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%)
<b>DIP16</b>	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%)
<b>SMD Resistor 1206</b>	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%)
<b>SOIC-8</b>	True positive	103/106 (97.2%)	102/106 (96.2%)	82/88 (93.2%)	86/88 (97.7%)	103/106 (97.2%)

		101/106 (95.3%)	93/106 (87.7%)	78/92 (84.8%)	92/92 (100%)	104/106 (98.1%)
<b>Ceramic capacitor 1210</b>	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%)
<b>SOT223-3</b>	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%)
<b>SMD Resistor 0806</b>	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%)
<b>TO263</b>	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%)
<b>Quartz HC-49/S</b>	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%)
<b>32-bit-PCI slot</b>	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 E RBF-SVM classification results

		Frequency features	Color features	Segment features	PCA reconstruction feature	Features selection from all features sets
<b>Tantalum capacitor</b>	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%)
<b>SMD Aluminum electrolytic capacitor</b>	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%)
<b>QFP100</b>	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%)
<b>SMD Resistor Network array 1206, 4 Resistors</b>	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%)
<b>SMD Transistor SOT23-3</b>	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%)
<b>DIP14</b>	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%)
<b>DIP16</b>	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%)
<b>SMD Resistor 1206</b>	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%)
<b>SOIC-8</b>	True positive	104/106 (98.1%)	105/106 (99.1%)	83/88 (94.3%)	87/88 (98.9%)	104/106 (98.1%)

	True negative	105/106 (99.1%)	96/106 (90.6%)	76/92 (82.6%)	92/92 (100%)	106/106 (100%)
<b>Ceramic capacitor 1210</b>	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%)
<b>SOT223-3</b>	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%)
<b>SMD Resistor 0806</b>	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%)
<b>TO263</b>	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%)
<b>Quartz HC-49/S</b>	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%)
<b>32-bit-PCI slot</b>	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 F Decision level fusion results

## Appendix G      Basis weight determination (PCB mounted)

<b>Length [cm]</b>	<b>Width [cm]</b>	<b>Weight [g]</b>	<b>Area [cm<sup>2</sup>]</b>	<b>Basis weight [<math>\frac{g}{cm^2}</math>]</b>
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
		<b>5700</b>	<b>7608.75</b>	<b>0.749137506</b>

## Appendix H      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	1	No

	leads (10mg) 1.4x3x1		
R0603	Widerstand Dickfilm Flat Chip 0603 (2.1mg)	3	No
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	
PINHEAD_2X07_127	3.5 x Widerstand Dickfilm Flat Chip 0402 (0.75mg)	1	Yes
0805	(Spule 0805)	5	
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 I Metal prices

<b>Material name (GaBi)</b>	<b>Material name (Data source)</b>	<b>Price [\$/kg]</b>	<b>Data source</b>	<b>Year (annual mean)</b>
Aluminium (E) [kg]	Aluminum metal	2.20	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/aluminum/myb1-2012-alumi.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/aluminum/myb1-2012-alumi.pdf</a>	2012
Aluminiumoxid (Al2O3) (E) [kg]	-	0	-	-
Anorganische Flammschutzmittel (E) [kg]	-	0	-	-
Antimonoxid (Diantimontrioxid) (Sb2O3) (E) [kg]	-	0	-	-
Blei (E) [kg]	Lead metal	2.40	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/lead/myb1-2011-lead.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/lead/myb1-2011-lead.pdf</a>	2011
Blei in Legierung (E) [kg]	Lead metal	2.40	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/lead/myb1-2011-lead.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/lead/myb1-2011-lead.pdf</a>	2011
Chrom in Legierung (E) [kg]	Chromium metal	13.30	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/chromium/myb1-2012-chrom.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/chromium/myb1-2012-chrom.pdf</a>	2012
Cobalt in Legierung (E) [kg]	Cobalt (minimum of 99.8% cobalt)	29.30	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/cobalt/myb1-2012-cobal.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/cobalt/myb1-2012-cobal.pdf</a>	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	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/gold/myb1-2011-gold.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/gold/myb1-2011-gold.pdf</a>	2011
Gold in Legierung (E) [kg]	Gold metal	50562.00	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/gold/myb1-2011-gold.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/gold/myb1-2011-gold.pdf</a>	2011
Kupfer (E) [kg]	Copper (London Metal Exchange, highgrade)	7.30	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/copper/mcs-2014-coppe.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/copper/mcs-2014-coppe.pdf</a>	2013

Kupfer in Legierung (E) [kg]	Copper (London Metal Exchange, highgrade)	7.30	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/copper/mcs-2014-coppe.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/copper/mcs-2014-coppe.pdf</a>	2013
Nickel (E) [kg]	Nickel metal	22.90	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/nickel/myb1-2011-nicke.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/nickel/myb1-2011-nicke.pdf</a>	2011
Nickel in Legierung (E) [kg]	Nickel metal	22.90	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/nickel/myb1-2011-nicke.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/nickel/myb1-2011-nicke.pdf</a>	2011
Palladium in Legierung (E) [kg]	Palladium metal	23665.00	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/platinum/mcs-2014-plati.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/platinum/mcs-2014-plati.pdf</a>	2014
Papier (E) [kg]		0	-	
Platin in Legierung (E) [kg]	Platinum metal	48585.00	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/platinum/mcs-2014-plati.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/platinum/mcs-2014-plati.pdf</a>	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	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/silver/myb1-2011-silve.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/silver/myb1-2011-silve.pdf</a>	2011
Silber in Legierung (E) [kg]	-	1133.76	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/silver/myb1-2011-silve.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/silver/myb1-2011-silve.pdf</a>	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	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/iron_and_steel/myb1-2012-feste.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/iron_and_steel/myb1-2012-feste.pdf</a>	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	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/zinc/myb1-2012-zinc.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/zinc/myb1-2012-zinc.pdf</a>	2012
Zink in Legierung (E) [kg]	Zinc (super high grade (SHG) zinc, 99.995% pure)	1.95	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/zinc/myb1-2012-zinc.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/zinc/myb1-2012-zinc.pdf</a>	2012
Zinn (E) [kg]	Tin metal	21.1	<a href="http://minerals.usgs.gov/minerals/pubs/commodity/tin">http://minerals.usgs.gov/minerals/pubs/commodity/tin</a>	2012

Zinn in Legierung (E) [kg]	Tin metal	21.1	n/myb1-2012-tin.pdf <a href="http://minerals.usgs.gov/minerals/pubs/commodity/tin/myb1-2012-tin.pdf">http://minerals.usgs.gov/minerals/pubs/commodity/tin/myb1-2012-tin.pdf</a>	2012
----------------------------	-----------	------	--	------