FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Automated optical inspection of solder joints by detecting the reflections of colored light
Student:	Jan Bouček
Supervisor:	Ing. Lukáš Brchl
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2022/2023

Instructions

Cílem práce je navrhnout potřebné vybavení a implementovat softwarový nástroj, který bude složit k automatické optické kontrole napájených součástek na desce plošných spojů. Systém bude určen primárně pro malé výrobny elektroniky, které si finančně nemohou dovolit zakoupit produkční stroj v hodnotě desítek tisíc dolarů. Systém se bude skládat z běžné barevné kamery a barevného světla, které bude nasvětlovat plošný spoj. Za pomocí algoritmů pro práci s obrazem budou následně detekovány součástky a zkontrolovány defekty spojů.

- Proveďte rešerši existujících řešení automatické optické inspekce.
- Vytvořte vhodnou snímací soustavu a navrhněte vlastní systém osvětlení vhodný pro optickou inspekci defektů.
- Implementujte algoritmy automatické optické inspekce pro detekci běžných defektů napájených součástek na plošných spojích.
- Proveďte zhodnocení dosažených výsledků a navrhněte budoucí rozšíření.

Electronically approved by Ing. Karel Klouda, Ph.D. on 12 February 2021 in Prague.

Bachelor's Thesis

AUTOMATED OPTICAL INSPECTION OF SOLDER JOINTS BY DETECTING THE REFLECTIONS OF COLORED LIGHT

Jan Bouček

Faculty of Information Technology CTU in Prague
Department of Applied Mathematics
Supervisor: Ing. Lukáš Brchl
June 27, 2021

České vysoké učení technické v Praze
Fakulta informačních technologií

Odkaz na tuto práci: Jan Bouček. *Automated optical inspection of solder joints by detecting the reflections of colored light.* Bachelor's Thesis. České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Contents

Acknowledgement vi

Declaration vii

Abstract viii

Abbreviations ix

1 Theoretical Background 3

1.1 SMT Manufacturing Process 3

1.1.1 Solder Paste Printing 3

1.1.2 Component Placement 3

1.1.3 Reflow Soldering 4

1.2 Inspection Phases in Manufacturing 5

1.2.1 Solder Paste Inspection 6

1.2.2 Automatic Placement Inspection 6

1.2.3 Post-Soldering Inspection 6

1.2.4 PSI Defects 7

1.2.5 Tiered Color Illumination System 8

1.3 Image Processing Methods 9

1.3.1 Convolutional Neural Network 9

1.3.2 VGG-16 9

1.3.3 ConvNet Transfer Learning 10

1.3.4 Principal Component Analysis 11

1.3.5 Data Augmentation 11

1.3.6 Decision Tree Classification 11

1.3.7 Random Forest 12

1.3.8 Cross-validation 12

2 Related Work 15

2.1 Image Acquisition 15

2.1.1 Illumination Systems 15

2.2 Image Processing 16

2.2.1 Solder Joint Localization 16

2.2.2 Feature Extraction 16

2.2.3 Classification 17

3 Proposed Approach 19
3.1 Image Acquisition . . . 19
3.1.1 Illumination Geometry . . . 19
3.1.2 Dome Enclosure Design . . . 20
3.1.3 Illumination Hardware Design . . . 21
3.1.4 Image Capture Station . . . 22
3.1.5 Camera . . . 23
3.1.6 Price Comparison . . . 23
3.1.7 Inspection Samples . . . 23
3.2 Image Processing . . . 25
3.2.1 Solder Joint Localization . . . 25
3.2.2 Feature Extraction . . . 26
3.2.3 Classification . . . 26

4 Results 29
4.1 Experiment Results . . . 29
4.2 Additional Considerations . . . 30
4.2.1 Simplified Dataset . . . 30
4.2.2 Classification Model Selection . . . 30
4.2.3 PCA Improvements . . . 31
4.2.4 Future Improvements . . . 31

5 Conclusion 35

Contents of the Enclosed Medium 43

List of Figures

1.1 Solder paste printing 4
1.2 SMT Assembly line including inspection machines [4]. 5
1.3 Fringe pattern projection SPI [2] 6
1.4 Tiered color light illumination system 8
1.5 An example of a ConvNet architecture [12]. 9
1.6 VGG-16 architecture [16]. 10

2.1 Alternative two color illumination system[37] 16

3.1 Illumination geometry model 20
3.2 Dome enclosure 3D model 21
3.3 Illumination hardware 21
3.4 Capture station structure models 22
3.5 Image capturing station 22
3.6 Solder joint localization process 25

4.1 Illumination system comparison 32

List of Tables

1.1 PSI soldering defect overview 7

3.1 Camera parameters used in image acquisition 23
3.2 Estimated cost of our illumination system 24
3.3 Sample dataset overview 24
3.4 Random forest classifier hyperparameters 27

4.1 Optimized random forest hyperparameters 29
4.2 Classification results for all classes 30
4.3 Classification results with a simplified dataset 30
4.4 Classification results for the AdaBoost experiment 31
4.5 Classification results for the experiment without PCA 31

I am thankful to my supervisor, Ing. Lukáš Brchl, for his mentorship and patience throughout my work on this thesis. I would also like to thank Ing. Tomáš Beneš for his help with hardware design and assembly.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on 27 June 2021 ....

Abstract

Quality control is an integral part of SMT electronics manufacturing. Automated inspection systems are available but costly. In this thesis, we propose a hardware and software solution that is less expensive and easier to implement. Our image capturing system is composed of 3D-printed parts and additional electronic parts for illumination. The proposed defect detection method consists of a simple thresholding-based localization method and a random forest classifier, which uses features extracted with a pre-trained VGG-16 neural network. Our model reaches an 80 % accuracy on our dataset, showing the success of our solution but also exposing shortcomings of our dataset and method. The thesis concludes with an analysis of results and suggestions for future improvements.

Keywords Automated Optical Inspection, Solder Joint Inspection, Surface Mount Technology, Printed circuit board, Transfer Learning, Illumination

Abstrakt

Kontrola kvality je nedílná součást výroby SMT elektroniky. Automatizované kontrolní systémy jsou dostupné, ale nákladné. V této práci navrhujeme hardwarové a softwarové řešení, které je levnější a jednodušší na implementaci. Náš snímací systém se skládá z 3D tištěných dílů a dodatečných elektronických součástek pro osvětlení. Navrhovaná metoda pro detekci defektů sestává z jednoduché prahovací metody lokalizace a klasifikace náhodným lesem, který používá příznaky extrahované pomocí předtrénované neuronové sítě VGG-16. Náš model dosahuje přesnosti 80 % na našem datasetu, čímž ukazuje úspěch našeho řešení, ale odhaluje také nedostatky našeho datasetu a postupu. Práci uzavírá analýza výsledků a návrhy pro budoucí vylepšení.

Klíčová slova automatická optická inspekce, inspekce pájených spojů, SMT, deska plošných spojů, transfer learning, osvětlení

Abbreviations

AOI Automated Optical Inspection.

API Automatic Placement Inspection.

CART Classification and Regression Trees.

ConvNet Convolutional Neural Network, also CNN.

HSV Hue, Saturation, Value.

IC Integrated Circuit.

LED Light-Emitting Diode.

LVQ Learning Vector Quantization.

MLP Multilayer Perceptron.

PCA Principal Component Analysis.

PCB Printed Circuit Board.

PSI Post-Soldering Inspection.

PWM Pulse Width Modulation.

RoI Region of Interest.

SJI Solder Joint Inspection.

SMD Surface Mount Device.

SMT Surface Mount Technology.

SPI Solder Paste Inspection.

SVM Support Vector Machine.

YOLO You Only Look Once.

Introduction

Most of the Surface Mount Technology (SMT) electronics production process is fully or partially automated to match the demand for high quality and low cost. Aside from fabrication itself, the quality control stages of production can benefit from automation just as well. Various commercial solutions are available for large-scale operations that leverage different imaging methods and evaluation of part quality, but none of these solutions are suitable for smaller production environments primarily due to high cost.

In this thesis, we aim to create a prototype consisting of software and hardware that could be used as a final production stage to visually verify a finished SMT product. We employ one type of Automated Optical Inspection (AOI) that focuses on imaging solder joints using a colored light system and use a transfer learning method combined with a random forest classifier to find any defects. The prototype is constructed using readily available parts and devices that are feasible for use in small-scale operations while still attempting to achieve high accuracy to bring a meaningful improvement in production efficiency.

In the first chapter we lay out the foundation of the SMT process, the AOI field, and the methods we use in our solution. In the second chapter, we describe and discuss the present solutions of AOI in general and specifically Post-Soldering Inspection (PSI). The following two chapters are dedicated to designing and implementing our prototype, followed by an analysis of the results we achieved.

We conclude the thesis by discussing possible modifications of our approach in the current state, along with suggestions for future work that could improve our solution with a significantly different method. These improvements may lead to a more robust solution and help implement it in production.

Our prototype shows promising results and demonstrates that it is possible to construct an automated quality control process even in low-volume production. The solution could be transformed into a working product with some modification and bring meaningful improvement to manufacturers.

Chapter 1
Theoretical Background

This chapter starts with a brief introduction to the field of SMT manufacturing, AOI in general, and the specific case of inspection we focus on. In the latter part, we describe image acquisition and processing methods, which we use in the following chapter to design and implement our prototype.

1.1 SMT Manufacturing Process

SMT production aims to mount SMD parts onto a Printed Circuit Board (PCB) using solder. There is an older method called through-hole technology (THT), which achieves the same goal by securing parts in drilled holes in PCBs, but SMT is dominant in manufacturing because, as per [1], it has many advantages, such as higher component density, ease of automation and lower part cost.

As described in [2, 3], SMT assembly (excluding inspection steps) is done in three phases: solder paste printing, component placement, and reflow soldering. These stages are described in the following sections. Sometimes, as mentioned in [3], *wave soldering* is used instead of *reflow soldering*, but the latter is dominant in SMT.

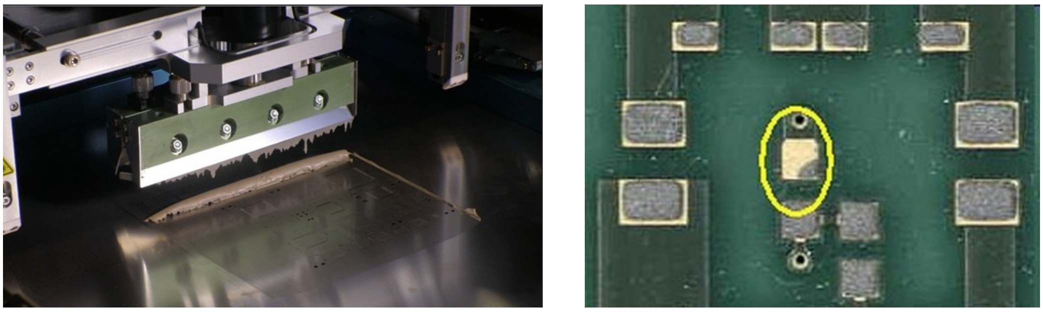
1.1.1 Solder Paste Printing

In the first phase, solder paste – a mixture of microscopic balls of solder and viscous flux – is applied to *solder pads* (exposed metal contacts on a PCB) in a *solder paste printer*. This machine, shown in fig. 1.1a, uses a metal stencil to apply solder paste precisely in all places on the PCB where parts will be soldered.

Common defects in solder paste printing are applying too much or too little solder paste on the PCB. If not corrected, this leads to parts completely or partially unattached to soldering pads or *bridges* – unintended electrical connections between neighboring soldering pads.

1.1.2 Component Placement

SMD components are placed onto a PCB in a *pick-and-place* machine. The parts are put in their place by a fast-moving head with a nozzle that holds components using a



(a) A solder paste printing machine [2]. **(b)** A solder paste printing defect [2]

Figure 1.1 Solder paste printing

vacuum on their way from a *part feeder* to their destination. As [2] states, these machines operate at up to 100,000 cph (components per hour) and can place a wide range of sizes and types of parts by using different nozzles and feeders. This machine commonly uses a vision system to place components correctly and check them for defects before placing them.

1.1.3 Reflow Soldering

In the last phase of assembly, the prepared PCB is heated up, solder paste is melted, and all connections between components and the board are made permanent. This process called *reflow soldering* consists of four stages:

1. preheating
2. soaking
3. reflow
4. cooling

This entire procedure is done in a *reflow oven*, a long machine where the prepared board slowly goes through all these steps on a conveyor system. First, the board is preheated to a high temperature but not enough to melt any solder. This is done slowly (at 2 °C or less per second, according to [2]) to avoid damaging any components.

The preheating phase is followed by a soaking period, where the temperature is kept stable to make sure all components are heated evenly. This is especially important for assemblies with a wide range of component sizes, as smaller components heat up faster than larger ones.

In the *reflow* step, the temperature is raised above the melting temperature for a short period. Correct timing and temperature are crucial to form a good connection and not damage components, which are often rated for temperatures just slightly above the reflow temperature. Short reflow time can result in unmelted solder paste that does not form a proper connection, but solder melted for too long can be prone to cracking.

Finally, the finished board is cooled down slowly to prevent similar issues as in the preheating step. This concludes the fabrication process, and the finished board is ready for final inspection and testing.

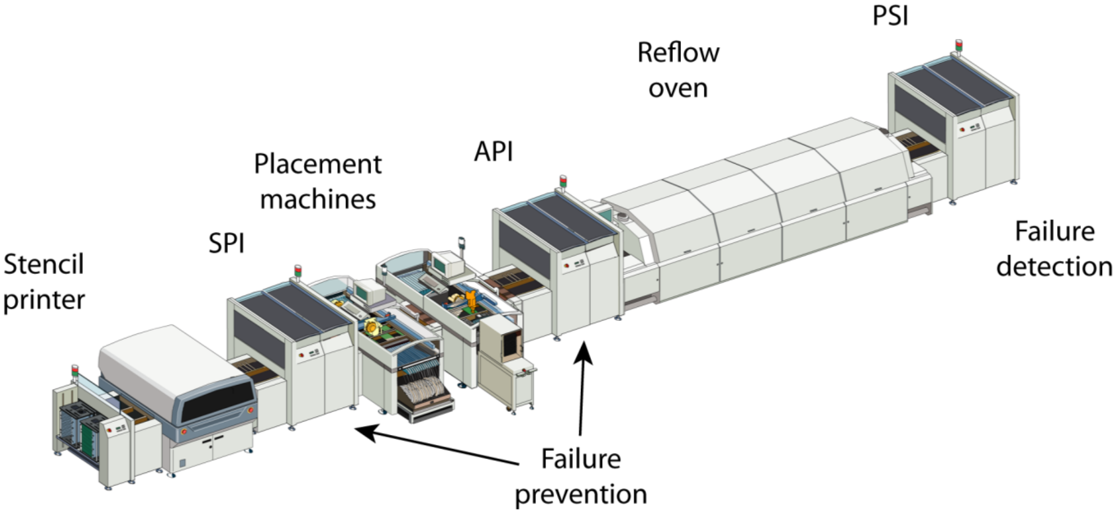


Figure 1.2 SMT Assembly line including inspection machines [4].

1.2 Inspection Phases in Manufacturing

As we described in the previous section, the SMT manufacturing process consists of numerous steps, each of which can result in some kind of defect that usually results in a product that does not function properly or has a risk of developing a malfunction later on.

There are many ways of testing and inspecting the PCB throughout the manufacturing process to prevent these issues. Functional testing is usually done at the end to verify the finished product's functional electrical properties, but it cannot be done in earlier stages, and it is not suitable for quality control in terms of long-term reliability. Functional testing methods are out of the scope of this thesis, and we will further focus on visual methods of inspection used in the SMT process.

Similarly to the assembly process, inspection is done in three main stages, each after one of the assembly steps to make sure it was executed correctly, and the board can be passed on to the next machine.

As described in [4, 2], the inspection stages are:

1. Solder Paste Inspection (SPI)
2. Automatic Placement Inspection (API), also known as Pre-Reflow Automated Optical Inspection or Post-Placement Inspection
3. Post-Soldering Inspection (PSI), also known as Post-Reflow Automated Optical Inspection

As shown in fig. 1.2, SPI and API are targeted at failure prevention because nothing is soldered and finalized at that stage, but PSI is only done for failure detection on a finished product, which can be then discarded or, in some cases, fixed manually.

The inspection steps are primarily done optically, hence the term Automated Optical Inspection. Although this name is used as an umbrella term for all three of the stages, the actual inspection methods, in the sense of image capturing and image processing, vary

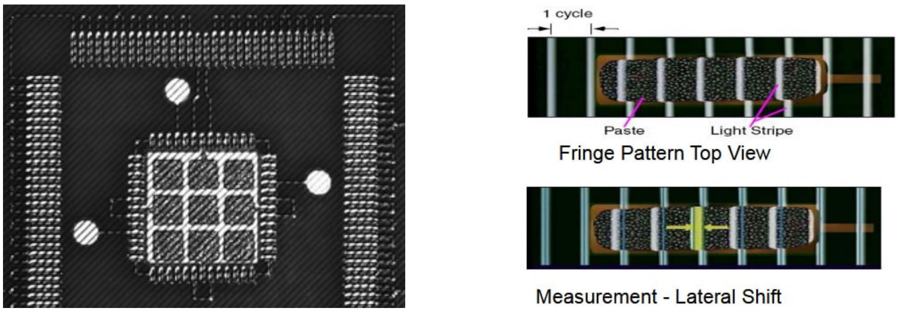


Figure 1.3 Fringe pattern projection SPI [2]

greatly. We will focus on these methods in the following sections with a brief description of SPI and API, followed by a deeper look into PSI, which is the main subject of this thesis.

1.2.1 Solder Paste Inspection

The base goal of SPI is to make sure that solder paste is present in all locations on the board, where it should be, and is not in places where it is not supposed to be applied. This goal can be achieved using a 2D SPI machine. However, modern ones are designed to do more than that – measure if the applied solder paste *volume* in each location is correct. To achieve inspection of volume, various optical methods are used to capture a 3D representation of the board, mainly *laser scanning* [5, 6, 7] and *fringe pattern projection* [8, 9, 10], shown in fig. 1.3.

1.2.2 Automatic Placement Inspection

As described in [4], API is focused on defects regarding placement of each SMD component, and current solutions can to check for errors in rotation, X/Y shift on the board surface, polarity, and more.

To achieve this, [2] mentions that *“the best way of doing this is by using an AOI machine”* in the sense that API is done with a machine working on the same principles as in PSI, which is (confusingly enough) often itself referred to as AOI. This is no surprise given that the only difference between API and PSI is the state of solder – printed solder paste and finished joints, but many defects in component placement are the same, regardless if they are caused by an error in placement or reflow.

1.2.3 Post-Soldering Inspection

As we discussed earlier, some defects may not become obvious in functional testing, and [2] mentions that manufacturers often require good post-reflow AOI to prevent early board failure, which may be costly. PSI is also by far the most researched form of inspection, as stated in [4].

A common approach to PSI is Solder Joint Inspection (SJI), which is focused on visually checking the quality of joints between the board and components. This is also

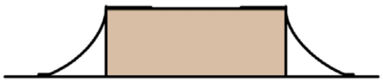
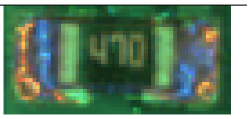
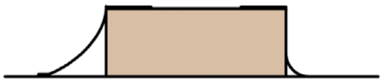
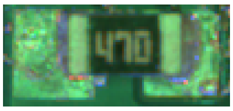
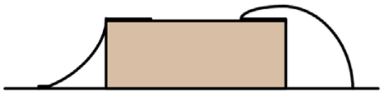
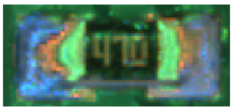
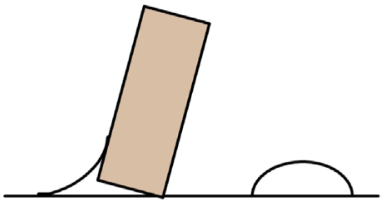
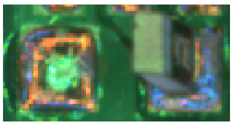
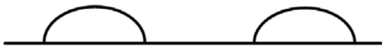
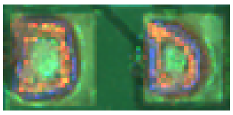
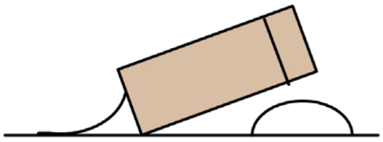
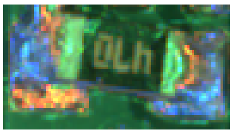
identifier	class name	class diagram [11]	dataset image
0	Acceptable solder joint		
1	Insufficient solder		
2	Excessive solder		
3	Tombstone		
4	Component missing		
5	Component shifted		

Table 1.1 PSI soldering defect overview

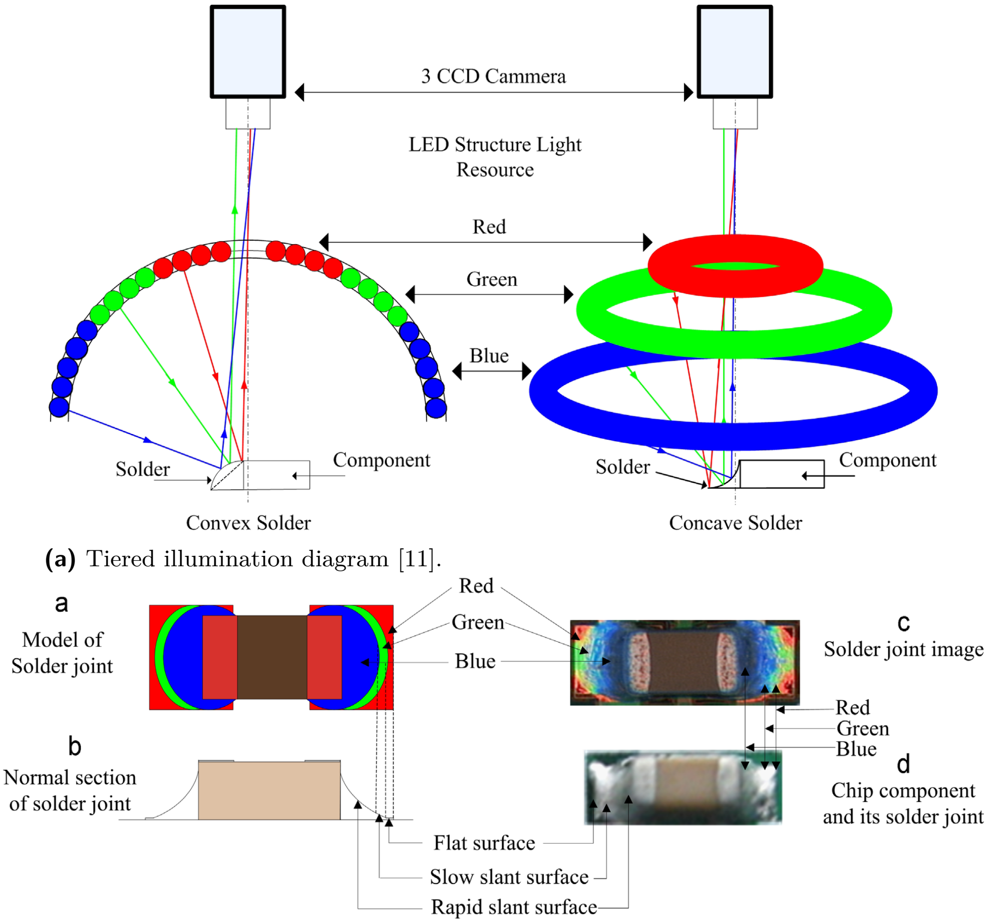
the type of AOI we chose for this thesis, as it is relatively easy to implement and very effective at discovering most of the frequent defects.

Some other methods do not inspect only joints but also all components on the board, which is more complex, but can uncover more subtle defects such as polarity flip and placement of a different type of component.

1.2.4 PSI Defects

We have encountered many ways of categorizing soldering defects across the AOI literature. For the context of this thesis, we selected 6 of the most common classes, inspired by [11]. All of them are shown in table 1.1. Dataset images are captured using the 3-color method described in the following section.

The first two types of defects are caused primarily by incorrect solder paste application, and the three remaining are usually a result of a failure in component placement, but a shifted component can often also be a result of excessive solder application.



(a) Tiered illumination diagram [11].

(b) Solder fillet color reflection effect [11]

Figure 1.4 Tiered color light illumination system

1.2.5 Tiered Color Illumination System

Our image acquisition approach uses an illumination system based on tiered color lights. As further discussed in section 2.1.1, this is a commonly used system in AOI, which has been implemented and improved upon multiple times.

As shown in fig. 1.4a, the most common configuration consists of three tiers of red, green, and blue circular light sources mounted in a dome-shaped enclosure. The camera is mounted above the center of the dome, pointed directly at the captured board, which is located below.

The advantage of this illumination system is that it allows us to evaluate the angle of a slope of solder at any point, giving us information about the 3D shape of a solder connection just from a 2D image captured by a single camera. This effect can be seen in fig. 1.4b, where parts of a solder fillet reflect the light of different colors based on which color of light meets the solder with the best angle of incidence to be captured by the camera above. The resulting color pattern makes it easier to determine the shape of solder and can be utilized in many ways by an image processing method.

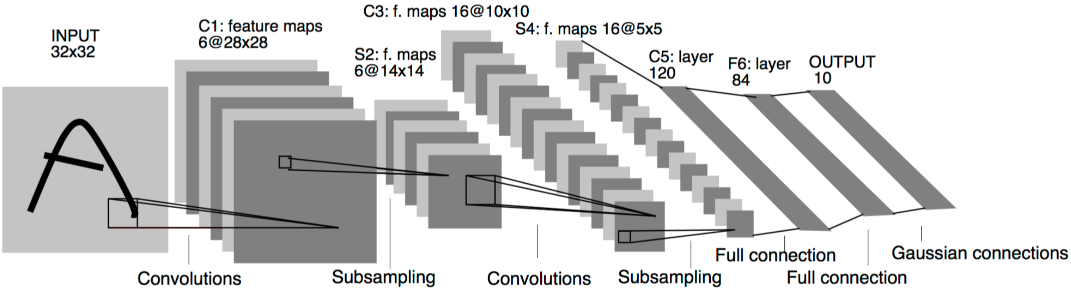


Figure 1.5 An example of a ConvNet architecture [12].

1.3 Image Processing Methods

In this section, we describe the main methods we used in our image processing pipeline. Some of them are related directly to image processing, other are more general algorithms of machine learning applied in our case of image classification.

1.3.1 Convolutional Neural Network

A Convolutional Neural Network, also CNN (ConvNet), is a type of neural network that is commonly used to extract information from images, especially for classification tasks.

The main building block of ConvNets are convolutional layers. Their purpose is to extract information from a localized region of the source image or an intermediary layer. An example of an architecture with convolutional layers is shown in fig. 1.5. These layers apply a discrete convolution to the input of each layer with filters (also *kernels*), which are composed of weights applied to each pixel in the region. Filter weights are optimized throughout the learning process, gradually creating filters that extract the most valuable information available in the layer.

Aside from convolutional layers, ConvNets usually contain pooling layers that reduce the dimension of the processed image (or rather *feature maps* – intermediary results of convolution). As [13] describes them, pooling layers also work on a localized level, most commonly with a 2×2 filter size and a *max* operation – compressing the region to its maximum value, reducing the feature map resolution to half in both dimensions.

1.3.2 VGG-16

VGG-16 [14] is a deep convolutional neural network (ConvNet) architecture originally developed for the 2014 ILSVRC competition [15] by the Visual Geometry Group (VGG) at the University of Oxford. The number 16 denotes the number of layers in the network, but the authors also experimented with configurations ranging from 11 to 19 layers, with VGG-19 being another popular choice for some applications. As shown in fig. 1.6, the VGG-16 architecture consists of 4 types of layers – convolutional (with 3×3 filters), max pooling (with 2×2 filters), fully connected, and softmax. The first two types have been described in the previous section, and the latter two are specific to the classification purpose of the model.

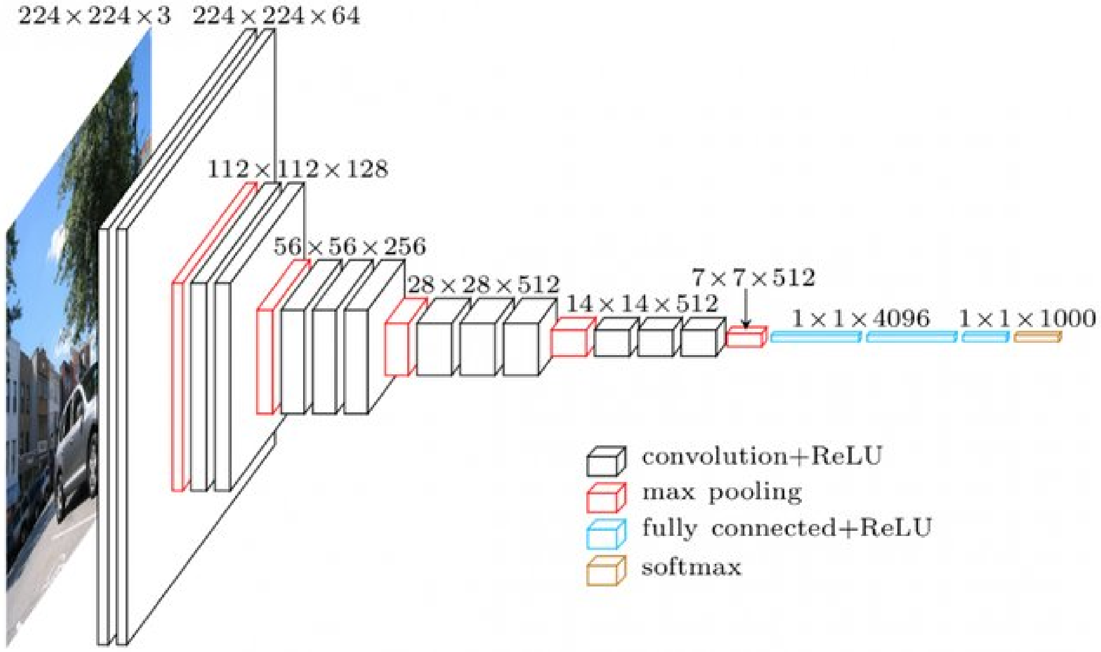


Figure 1.6 VGG-16 architecture [16].

Fully connected layers, also used in traditional neural networks, compose each element of the output matrix by applying weights to the whole input feature map, gaining the ability to extract global information across the whole processed image, but at the cost of dramatically increasing the number of parameters (weights) in the model.

The final layer of the VGG-16 architecture is a softmax layer that converts the data from the previous layers to the predicted probabilities of each class in the classification task – the popular ImageNet dataset containing 1000 categories.

1.3.3 ConvNet Transfer Learning

As described by [17], transfer learning is a method of using gained knowledge on a *source domain* in a *source learning task* and using this knowledge to improve learning on a new *target domain* in the *target learning task*.

In ConvNet image classification tasks, this method is used based on the assumption that the source domain (usually a large image dataset) can be used to learn knowledge that is also useful in the target task – another classification task with a target domain. However, this precondition cannot be applied blindly. There are rare cases, such as [18], where the target domain is different from the source domain in ways that do not allow for effective transfer learning.

This approach is often used solely because the target domain contains significantly fewer data points than the source domain, and pre-trained ConvNets on source domains (most commonly the ImageNet dataset) are widely available.

As [19] describes the process, transfer learning on ConvNets is practically done in the following way:

1. train the desired ConvNet for the source task (or use a pre-trained one)
2. remove the last layer, which is specific to the classes used in the source domain
3. add a new final layer with the correct dimension for our target domain
4. randomly re-initialize all fully connected layers that lead up to the final layer
5. train the network on the target domain, potentially leaving the convolutional layers without change (*freezing* them)

The other ConvNet transfer learning approach is to use the pre-trained network as a generic feature extractor, and as [20, 21] show, this is a viable method in the AOI context. This method leaves out the whole training phase and uses the source ConvNet without any changes, only to extract raw data that can be processed in some way for the target task. This is usually done by selecting one of the last layers in the network and using it as the output, based on the assumption that this data contains valuable information relevant to the target task. The last convolutional layer is most commonly selected as the output, but as [20] suggests, features from fully connected layers can also be used to extract more *high-level* features related to the source classification task.

1.3.4 Principal Component Analysis

Principal Component Analysis (PCA) is a popular method for dimensionality reduction, meaning that it allows us to lower the dimension of data without losing much information. This is done by finding an orthogonal linear transformation that, when applied to our data (an $n \times p$ matrix X with n data points and p features), results in a new matrix containing new features with the maximum variance, sorted in descending order, giving us the most *"important"* information in the data [22].

The method uses a modified matrix X' that is centered by the sample mean in each dimension. A covariance matrix is then computed as $\frac{1}{n-1}X'^T X'$, and the desired components are the eigenvectors of this matrix sorted by their eigenvalue. These vectors can then be used as a matrix for the orthogonal transformation on the source data.

1.3.5 Data Augmentation

Data augmentation is a technique most frequently used in image-based machine learning to enlarge a dataset and reduce overfitting. The concept of augmentation is to create new data points (images) using the original dataset. As [23] describes, there are many ways of manipulating an image to produce a new sample, which is effective in learning. This includes simple geometric methods – flipping, rotation, cropping, translation, and more sophisticated methods, such as noise injection and color space transformations.

1.3.6 Decision Tree Classification

Decision trees, also known as Classification and Regression Trees (CART) [24], are, as the name suggests, is a predictive model type used both for classification and regression. Each prediction in this model starts at the tree's root node, then each node represents a single decision based on a parametrized condition, and the prediction ends when a leaf node is reached – each of the leaf nodes is assigned with a prediction value.

The decision tree is iteratively built from the root, based on the training dataset. For the classification task, we have fractions $p_1, p_2, \ldots, p_n$ for the current node in each iteration, where p_i is the proportion of class i in the subset T of training inputs predicted to this node by the current tree structure [25].

If $|T| > 1$ (1 can be substituted by a constant), the algorithm finds the best way to create a new condition (based on available features) to split the training subset into two parts with the highest information gain IG, based on the parent node entropy $H(T)$, and weighted entropy (by proportion in T) of each of the proposed child node training subsets T_0 and T_1, calculated by the following formula:

$$H(T) = \sum_{i=0}^{n-1} p_i(T) \log p_i(T)$$

$$IG = H(T) - t_0 \cdot H(T_0) - t_1 \cdot H(T_1)$$

There is also a popular alternative with Gini impurity instead of entropy, the information gain is then calculated as follows:

$$IG = 1 - \sum_{i=0}^{n-1} p_i(T)^2$$

1.3.7 Random Forest

Random Forest [26] is an *ensemble learning* model suitable for both classification and regression tasks. The model is composed of many decision trees (described in section 1.3.6), which do not perform as well on their own, but they are used to *"vote"* on the prediction result, enabling more robust generalization, and thus reducing overfitting, which simple decision trees are prone to. The trees in a Random Forest are also usually much smaller than a single decision tree model would be for the same task.

Two main methods are used to create decision trees that are different from each other (with lesser correlation) – Bootstrap Aggregation (*Bagging*) and feature selection (*feature bagging*). Bagging assigns each decision tree a different input training set by selecting a specified number of samples with replacement. Feature bagging is applied in each node split during the learning process by selecting a random subset of features to consider for splitting. This forces the decision trees to split using more obscure features instead of a few popular ones used in most of the trees.

1.3.8 Cross-validation

Similar to data augmentation (described in section 1.3.5), cross-validation is another method frequently used to improve a model's performance on a small dataset. Traditional training methods use a separate training, validation, and test dataset, limiting the size of the training dataset to only a small portion of the total dataset size, especially when the test and validation dataset needs to be large enough to evaluate the quality of the model.

Cross-validation removes this shortcoming by eliminating the validation dataset and generating different sets of training and validation data multiple times to evaluate the quality of the same model configuration.

The *k-fold* method used in this thesis achieves this by splitting the dataset into k *folds* – k randomly selected equally sized subsets. For each of these subsets, we run our training procedure with all data not included in the subset and evaluate the model on the remaining data in the subset. This produces k scores for our model, and their average is then presented as the final estimation of the model's performance metric.

As we stated earlier, this method is beneficial in cases where we don't have enough data. However, it comes at a performance penalty because the whole training process needs to be run k times for a single evaluation.

Chapter 2

Related Work

There have been numerous attempts at the task of detecting soldering defects over the past decades. This chapter introduces some notable approaches and compares their differences.

2.1 Image Acquisition

Both 2D and 3D image capturing methods are used in the SMT industry. However, as [2] warns, the difference is not clear because AOI system manufacturers often claim *3D capability*, even if the system uses only a 2D image to assume the 3D properties of the inspected article. This approach is also sometimes referred to as *2.5D*.

A *"pure"* 2D capturing solution can be as simple as taking an image of the finished board with a DSLR camera as in [27]. Commercial *"full 3D"* solutions are widely available for purchase, such as [28], and some technical information about them is available [29], but the methods behind 3D reconstruction-based post-reflow AOI are currently not well represented in literature.

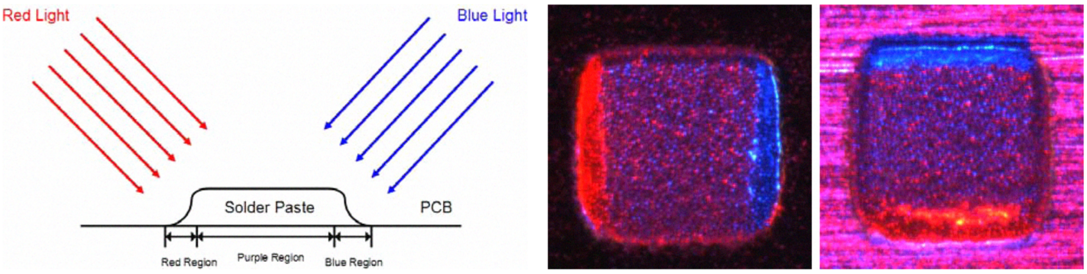
2.1.1 Illumination Systems

The vast majority of research has, for decades, focused on development of 2D based solutions, where the main difference in image capturing is often the illumination system used to highlight key features of the inspected board.

One of the illumination solutions, arguably the most popular one, is a *tiered color light system* first described in 1988 [30], which was later improved upon and used by [31, 11, 21, 32, 33, 34, 35, 36].

As described in section 1.2.5, this system usually consists of three tiers of red, green and blue circular light sources, but different configurations have also been implemented with two [30] and four [33] levels and colors. Older solutions, such as [30], use fluorescent lamps to achieve this setup, while modern ones dominantly use circular LED arrays, usually mounted inside a dome-shaped enclosure as described in [11, 32, 31, 33].

Other, less popular, illumination approaches exist, such as a two-color system [37], where three images are taken, one with top-mounted white light and two with opposing red and blue light, as shown in fig. 2.1.



(a) Two color illumination diagram **(b)** Two color illumination image example

Figure 2.1 Alternative two color illumination system[37]

This system is easier to implement, but it is highly dependent on the orientation of SMD components, which may, in practice, be mounted at any angle.

2.2 Image Processing

Depending on the image capturing method, there are many ways of processing this data to extract meaningful information, mainly as a task of classification - deciding on whether the inspected part is defective or not, usually specifying a type of defect.

2.2.1 Solder Joint Localization

Solder joints must be first localized to extract features from a board image. This is usually done by matching a template of a manually verified sample [27, 38, 39] or by dynamically localizing solder pads in each sample, which can be done in different ways. More straightforward thresholding-based approaches are common, such as [40, 41, 42, 43]. There is also a recent example of a modern You Only Look Once (YOLO) convolutional neural network object detection method used by [21] with good results.

2.2.2 Feature Extraction

When a solder joint is localized, the extracted RoI can be directly used in a classification algorithm, or, more commonly, features are extracted from the image to create data that is more suitable for use in the chosen classification algorithm, while often bringing out some desirable properties of the image. Feature extraction methods vary both in the types of features and the method used to extract them. A review of previous works [44] shows that many of the earlier approaches use wavelet transform and various geometric properties, such as length, area, perimeter, and compactness.

A recent article [21] also describes feature extraction based on transfer learning – using a pre-trained neural network's ability to generalize for image classification tasks to generate features for an unrelated dataset (further described in section 1.3.3). This approach uses transfer learning only to extract features, but their work is also based on [19], which shows that transfer learning can also be used directly for classification (in a similar AOI task) by modifying the pre-trained network and fine-tuning.

2.2.3 Classification

Like the previous stages of the inspection pipeline, the choice of a classification algorithm is not clear. Many methods have been used successfully with no obvious performance differences. The accuracy of each classification model may vary greatly based on the features used, number and types of defects targeted, and there is also no well known public dataset to be used as a benchmark for comparisons. However, some literature uses datasets from other AOI tasks to compare performance, such as a metal strip surface defect dataset [45].

Neural network based classifiers have been a popular choice, even as early as 1995, starting with Multilayer Perceptron (MLP) [46] and Learning Vector Quantization (LVQ) models [35]. There have also been successful attempts with a Support Vector Machine (SVM) classifier [21, 47], a genetic algorithm combined with a neural network [48], and ensemble approaches with AdaBoost [49] and Random Forest [50].

Chapter 3

Proposed Approach

As we stated in the introduction, our primary goal is to create a prototype of an AOI system suitable for small-scale SMD manufacturers, such as startup companies and hobbyists. To achieve this, we need an affordable hardware system and defect detection software that is suitable to use even in a small production run of a few boards instead of millions. In the following sections, we will go through the design and implementation of our system.

3.1 Image Acquisition

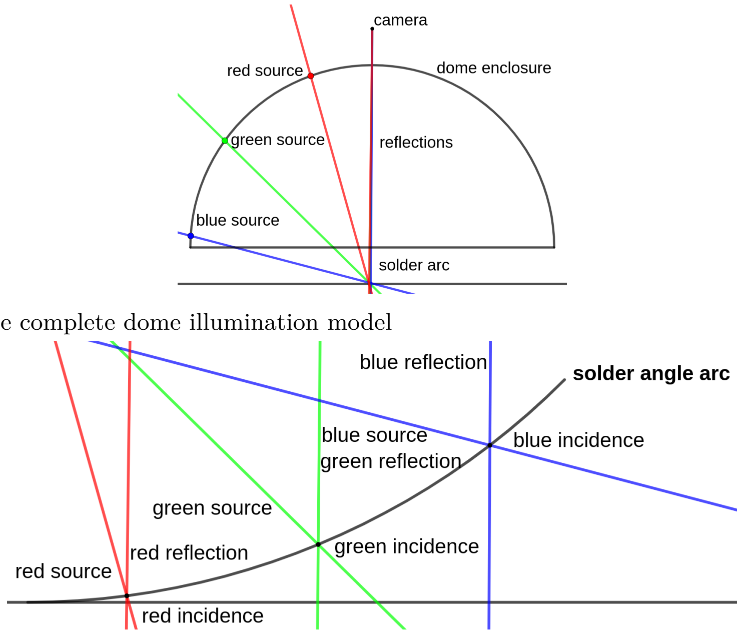
Our image acquisition solution is composed of multiple parts, most of which are designed and made by us. In this section, we will describe all these parts, starting with the illumination system and ending with the camera configuration.

As we discussed in 2.1.1, a popular illumination solution is a dome-shaped three-color system that highlights the three-dimensional shape of solder. We set out to design and build our solution that can be made using a 3D printer and relatively inexpensive electronic parts. The following sections will focus on the specifics of this design, and in section 3.1.6 we will summarize the price comparison against a commercial solution.

3.1.1 Illumination Geometry

To achieve the desired color pattern reflection, we devised a geometric model, shown in fig. 3.1. When using a dome-shaped enclosure, we can place the light sources in such a way that we can reflect light into the top-mounted camera from solder at an angle of 0° to 45°, or practically a slightly narrower range due to the top cut-out for a camera and the vertical gap between the bottom of the dome and the inspected board.

We divided this range of angles into three parts – 0° to 15°, 15° to 30°, and 30° to 45°. For each of these angle ranges, we modeled a reflection from a light source mounted on the dome's surface towards a simplified pin-hole model camera with an optical center 10 mm above the top of the dome. The modeled light rays are reflected from a circular arc (with a radius of 1 mm) placed near the horizontal center 10 mm below the bottom rim of the dome. We believe this representation is close to the real inspection the dome will be used for.



(a) The complete dome illumination model

(b) 3 light ray reflections across the 45° angle range

Figure 3.1 Illumination geometry model

The result of this model is a set of computed positions and angles for three layers of light sources to be mounted inside the dome.

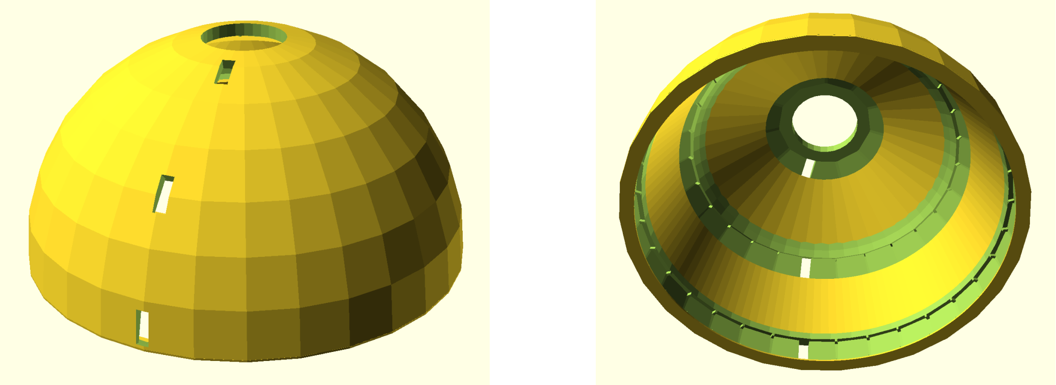
3.1.2 Dome Enclosure Design

For the light sources in each of the three layers, we decided on an array of small PCB segments, each of which is mounted with three LED components of one color. These segments are of trapezoidal shape in order to form a circular array at the specified angle with as small gaps as possible, so that we could solder these components directly together without any wires.

Based on preliminary design of the segments (discussed in detail in the next section), we decided to work with a minimum size of 8.1 mm vertically and 7.5 mm horizontally, increasing the horizontal dimension on one side to form a trapezoid of the desired size for each layer.

We then wrote a Python script to calculate the number of segments and their dimensions for each layer. Based on the results, we finalized the PCB design for the segments and created their 3D models. These models were then used to design the whole dome enclosure using *SolidPython* [51], a Python library for generating code for *OpenSCAD* [52] – a code-based CAD modeler.

The resulting model shown in fig. 3.2 is a dome with cut-outs to fit and hold each layer, a hole for the camera and additional holes for the light array wiring. We then printed this structure in a 3D printer and mounted it with illumination hardware discussed in the following section.



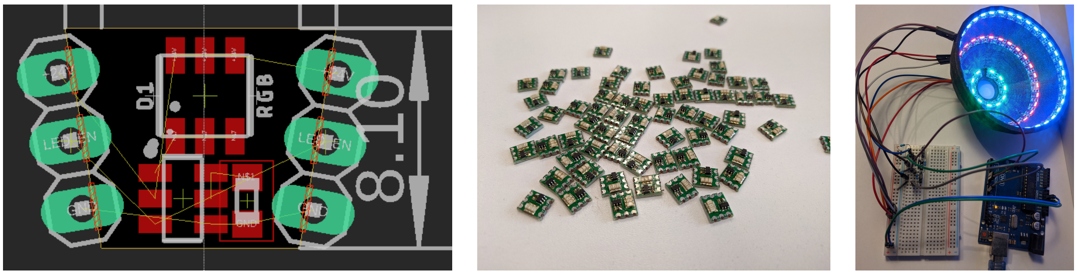
(a) Dome enclosure 3D model exterior

(b) Dome enclosure 3D model interior

Figure 3.2 Dome enclosure 3D model

3.1.3 Illumination Hardware Design

The illumination system hardware, pictured in fig. 3.3, consists of three main components – the three layers of LED segments, a breadboard for wiring and additional components, and an Arduino Uno Rev3 microcontroller board.



(a) LED segment hardware design

(b) LED segments soldered with components

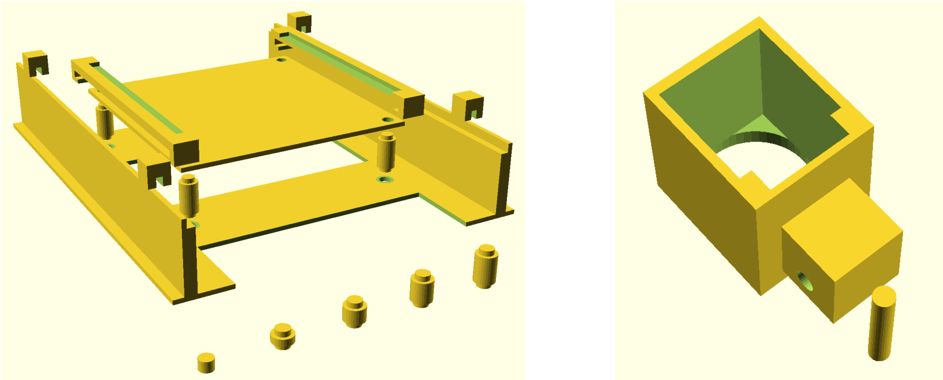
(c) Complete illumination system

Figure 3.3 Illumination hardware

As we mentioned in the previous section, the LED light arrays consist of many PCB segments. These segments, shown in figs. 3.3a and 3.3b, are mounted with 5 SMD components: 3 color LEDs, 1 resistor and 1 driver IC for the LEDs. The segments are joined by 3 soldered connections on each side, providing 5 V power, ground, and light intensity control. One connection in each layer is wired out to the breadboard, where a Pulse Width Modulation (PWM) signal from the Arduino board controls the light intensity using 3 transistors.

The USB-powered Arduino board is used both as a power source and as a simple computer that controls the light sources by setting 3 output pins to a PWM mode with an experimentally selected duty cycle length.

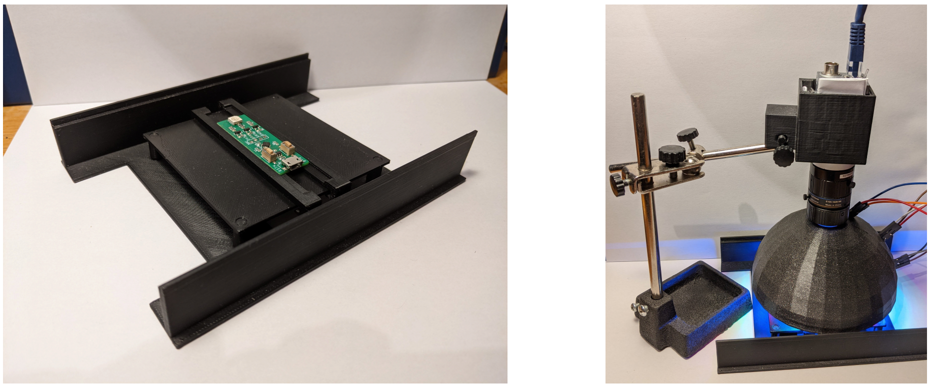
Unlike most other tiered color light designs discussed in section 2.1.1, we decided to change the order of the two top layers to use the LEDs with the highest light output (green) in the top layer, where we could only mount 10 segments. This choice results in visually different images but should not affect performance in any way.



(a) Sample positioning structure

(b) Camera mounting bracket

Figure 3.4 Capture station structure models



(a) Sample positioning structure

(b) Complete image capturing station

Figure 3.5 Image capturing station

3.1.4 Image Capture Station

To consistently capture images of all sample boards, we needed a structure to hold all components of the capturing equipment together while allowing for variable positioning in order to position the inspected component in the center of the illumination system according to the model described in section 3.1.1.

To meet all these requirements, we developed another 3D printed structure made up of many parts, shown in fig. 3.4a. At the very bottom is a base with 4 small holes and two rails that the illumination dome can slide on. The holes in the bottom are used to hold 4 cylindrical posts of variable length (shown in front) that allow us to choose a vertical position of the sample from 0 mm to 12 mm. On top of the posts, we placed a flat plate that holds two long brackets. These brackets slide across the plate in one axis and have an indentation on their inner side to clamp the inspected sample and enable positioning in the perpendicular horizontal axis. The final components of this structure are 4 small cube-shaped pegs that attach to the side rails to lock the dome position.

The second part of the capturing station is the camera arm, shown in fig. 3.5b. We repurposed a simple Powerfix brand *"helping hands"* holder for soldering and 3D printed a mounting bracket (pictured in fig. 3.4b) for our camera with a small pin to lock it onto one of the arm attachment parts.

The complete image capturing station, including the camera (further discussed in the following section), is shown in fig. 3.5b.

3.1.5 Camera

At first, we experimented with a simple and inexpensive *USB130W01MT* model RGB camera with a resolution of 640x480, but we found that the resolution and overall image quality were not good enough to capture small SMD parts. The camera body we used to capture our dataset is a *Basler acA1920-50gc* [53] – an RGB camera with a resolution of 1920x1200, and its list price is 19342 CZK (at the time of writing). For the camera lens, we chose a *Basler C125-1620-5M* [54] with a fixed focal length of 16 mm, listed for 2777 CZK.

We captured images from this camera over an Ethernet connection using *pylon Viewer* [55] software provided by Basler. After some experimentation, we settled on hardware and software camera parameters in table 3.1. All unlisted parameters performed best at their default value.

parameter	parameter value
Aperture	F3.0
Exposure	5000 µs
White Balance Correction	disabled
Color Adjustment	disabled

Table 3.1 Camera parameters used in image acquisition

3.1.6 Price Comparison

Our solution is composed of multiple parts that we described in the previous sections. In table 3.2, we summarize the cost of each part. For comparison, a commercial tiered color illuminator is available for purchase locally from *ATEsystem s.r.o.*, which sells a unit of *OPT-RIA200-RGB* for over 60,000 CZK. With a total estimated cost of 1860.86 CZK, we think our solution accomplishes its goals in terms of price by a large margin.

3.1.7 Inspection Samples

As the sample PCB for our experiments, we selected the board described in section 3.1.3 and used to construct the illumination system because we had a large number of extra units, many of which had genuine defects that we unintentionally created during our production.

From this board, pictured in fig. 3.6a, we decided to inspect only one of the parts – a resistor with two solder pads. The other components could also be inspected, but the three LED modules on each board had a single joined solder pad on each side, caused

part name	quantity used	unit price estimate (CZK)
array segment PCB, wide variant	10	5.34
array segment PCB, narrow variant	60	5.23
LED driver	70	4.60
red LED light	70	0.67
blue LED light	70	1.83
green LED light	70	2.75
array segment resistor	70	0.13
transistor	3	5.50
1 kOhm resistor	3	3
5 kOhm resistor	3	3
Arduino Uno v3	1	600
breadboard	1	62
wires	14	2.48
3D printing	106.4 g	0.6
total price		1860.86

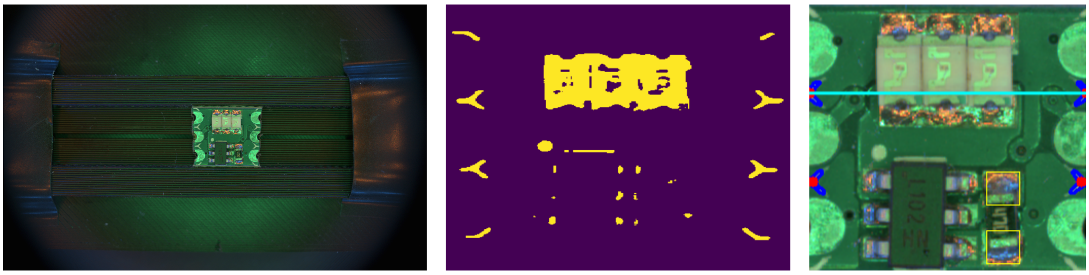
■ **Table 3.2** Estimated cost of our illumination system

by a mistake in design, and we also chose not to inspect the LED driver IC because, as [21] suggests, IC solder joints are more challenging to inspect and are often solved using different methods than regular non-IC components.

Most of the sample images we collected were from our small production using a reflow oven, but we also had to create some defects manually using a soldering iron because some defects did not occur naturally as often. However, many synthetic samples do not look as consistent as those soldered in the reflow oven, partly because of a different solder type (lead-based solder wire instead of solder paste) and also because of unsatisfactory soldering skills, which also lead to many of these sample images being discarded. This manual process was time-consuming, so there is still a significant disparity between the different classes of defects in the resulting dataset, as shown in table 3.3. The whole dataset is also included in the attached medium.

class identifier	sample class	number of samples in dataset
0	Acceptable solder joint	92
1	Insufficient solder	51
2	Excessive solder	45
3	Tombstone	12
4	Component missing	136
5	Component shifted	18

■ **Table 3.3** Sample dataset overview



(a) Source image from our dataset

(b) Processed mask for landmark extraction

(c) Cropped image with landmarks, an angle correction line and solder pad RoIs

Figure 3.6 Solder joint localization process

3.2 Image Processing

This section will go over all the steps in our image processing pipeline, starting with the dataset we captured with the method described in the previous sections and ending with the classification of the processed data.

3.2.1 Solder Joint Localization

First, we need to extract the solder joints from the source image. As discussed in section 3.1.7, the task is to extract two solder joints of one component from each image.

We decided on an ad-hoc solution tailored to our dataset because a generalized approach, such as object detection using a YOLO model used by [21], would require a significantly larger dataset with manually located solder pads and would be more difficult to implement. The following approach would not be suitable for real-world use in an AOI application, but it was sufficient to extract data for our classification experiment.

We created a Python script for this task and used the OpenCV library [56] for most of the image processing.

In the beginning, the source image (example shown in fig. 3.6a) is cropped down to the center section – all the target PCBs are located in the center as a result of our positioning system. Then we create a version of the image in the HSV color representation, which is better suited for color-based thresholding. To precisely locate the board, we used four white arrow-shaped marks, which are present at the edges of each board, as landmarks.

We first filtered the image using two thresholds to extract these shapes, which results in a binary mask containing only parts of the image similar (in terms of color) to the landmarks. We then applied a blur using a median filter with a size of 5×5 to reduce the effect of inconsistency in the image, such as noise and dust particles on the sample, and eliminate any small gaps in the contours contained in the mask. One finished mask example is shown in fig. 3.6b.

In the next step, we extract all contours from the mask using the algorithm [57]. These contours are then filtered by multiple properties to remove all unwanted contours that are of similar color as our landmarks:

- bounding box width and height

- bounding box position relative to the minimum and maximum x/y position of all contours (removes contours from the center of the PCB)
- bounding box aspect ratio
- contour perimeter length

Based on the position of the four landmarks, we use their centers to create two horizontal lines (one line drawn as yellow in fig. 3.6c) and calculate their angles. The average of these angles, together with the positions of the landmarks, is then used to rotate and crop the image so that we have a consistent image of the inspected board.

The final step is the solder pad Region of Interest (RoI) extraction, which is done using a template of RoI positions relative to the board image dimensions. These templates were slightly different for the two variants of the sample PCB.

3.2.2 Feature Extraction

Inspired by [21], we chose a transfer learning based approach for feature extraction using the VGG-16 model (discussed in section 1.3.2) pre-trained on the ImageNet dataset, provided by the Keras [58] library.

This means that we used the prepared RoI image dataset images as the input for the model but recorded the output from an intermediary feature map as the output, specifically the last layer before the final fully-connected layers. This yields us 25088 features that could be potentially used in any classification algorithm.

To reduce the number of features, we used the PCA algorithm, discussed in section 1.3.4. This step was not necessary, but it helped both in terms of performance and in order to remove many features that carry little information useful for classification, which helps our classifier with feature selection. The use of PCA allowed us to reduce the number of features from the original 25088 down to 30 without any significant loss of accuracy.

We also used a simple data augmentation method to increase the total number of samples in our dataset. As discussed in section 1.3.5, there are many of ways to augment an image to create a new one, such as horizontal/vertical shift, flip, rotation, and zoom. We chose to use only the vertical and horizontal flip methods to generate 3 additional images from one source, as most of the other augmentation methods could produce an image very similar to a defect from an image of an acceptable joint.

3.2.3 Classification

The final part of the whole image processing pipeline is the classification stage – an *ensemble learning* method based on a Random Forest model (described in section 1.3.7) composed of *decision trees*.

Our implementation is based on the scikit-learn [59] library, which provides a model that can be tuned with many hyperparameters. We chose 4 of these parameters; their values and descriptions are listed in table 3.4.

The model was trained for each combination of these parameters, and its performance was measured using the *k-fold* cross-validation method (described in section 1.3.8) with

parameter name	description	value range	step
`n_estimators`	number of decision trees	100–220	20
`max_depth`	maximum decision tree depth	4–14	2
`criterion`	function used to measure split quality	Gini, entropy	–
`max_eatures`	maximum features used by a decision tree	4–14	2

■ **Table 3.4** Random forest classifier hyperparameters

5 folds. We chose cross-validation over the traditional training/validation data split primarily to avoid decreasing our effective training dataset size.

Chapter 4
Results

4.1 Experiment Results

Our classification experiment, described in the previous chapter, was executed in a Google Colaboratory PRO [60] environment with a Tesla P100 GPU [61]. Both the feature extraction stage and classification training with all hyperparameter combinations took less than an hour. The best model achieved an overall 85.28 % mean cross-validation accuracy and an 80.33 % accuracy on the test dataset (25 % of all samples) with hyperparameters listed in table 4.1

parameter name	value
`n_estimators`	120
`max_depth`	12
`criterion`	entropy
`max_eatures`	8

Table 4.1 Optimized random forest hyperparameters

When we look at the model's performance on each of the classes with the precision and recall metrics in table 4.2, it is evident that a large part of the errors come from the three least represented classes. This issue could be caused simply by the low number of samples, making it harder for the classifier to *learn* these classes. However, it could also be caused by the representation differences in the training dataset because predicting one of the larger classes is less likely to be an error, thus making it *"riskier"* for the model (in terms of cross-validation scoring) to predict a smaller class. This effect can be observed from the recall metric, which shows that the model predicted these classes far less frequently than the larger ones, especially the Tombstone class with only 17 % recall.

The disappointing performance on the Tombstone class is very likely also caused by the wildly different visual appearance of each sample, of which only one is organic, and the rest was created artificially.

class identifier	class name	precision	recall	samples in test dataset
0	Acceptable solder joint	79 %	97 %	104
1	Insufficient solder	81 %	88 %	48
2	Excessive solder	55 %	45 %	40
3	Tombstone	44 %	17 %	24
4	Component missing	93 %	93 %	120
5	Component shifted	67 %	50 %	20

Table 4.2 Classification results for all classes

class identifier	class name	precision	recall	samples in test dataset
0	Acceptable solder joint	85 %	98 %	100
1	Insufficient solder	85 %	75 %	44
2	Excessive solder	97 %	62 %	48
4	Component missing	92 %	97 %	132

Table 4.3 Classification results with a simplified dataset

4.2 Additional Considerations

Aside from our main experiment, we also tried changing some components of the approach. Some of these experiments are documented in the following section, along with an analysis of the results. The section ends with some suggestions for future work based on the outcome of our experiments.

4.2.1 Simplified Dataset

As an experiment to demonstrate the performance of our approach without the issues caused by insufficient samples in some classes, we ran the same experiment without the two least represented classes. This resulted in a test dataset accuracy of 89.19 %, with significantly more consistent metrics across the dataset (shown in table 4.3). These results show that our model has the potential for better accuracy with a more balanced dataset and more data in each class. However, this does not eliminate the risk that the omitted classes are more difficult to classify even with more data, which is entirely possible.

4.2.2 Classification Model Selection

To confirm our classification model selection, we tried a very similar experiment with another ensemble learning based AdaBoost classifier [62, 63], also provided by scikit-lean. The results of this experiment in table 4.4 show very similar results with an accuracy of 79.77 % on the whole test dataset.

This experiment demonstrates that our specific selection of a Random Forest classifier

class identifier	class name	precision	recall	samples in test dataset
0	Acceptable solder joint	75 %	95 %	104
1	Insufficient solder	80 %	90 %	48
2	Excessive solder	64 %	40 %	40
3	Tombstone	43 %	12 %	24
4	Component missing	90 %	94 %	120
5	Component shifted	77 %	50 %	20

Table 4.4 Classification results for the AdaBoost experiment

class identifier	class name	precision	recall	samples in test dataset
0	Acceptable solder joint	70 %	97 %	104
1	Insufficient solder	94 %	71 %	48
2	Excessive solder	54 %	38 %	40
3	Tombstone	00 %	0 %	24
4	Component missing	82 %	100 %	120
5	Component shifted	00 %	5 %	20

Table 4.5 Classification results for the experiment without PCA

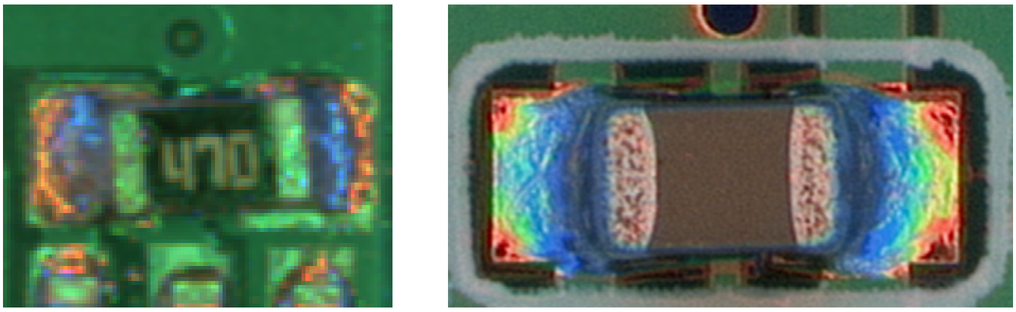
did not negatively affect the results, but it also shows that we would likely need to choose an entirely different kind of model to improve the performance of our classification stage alone, perhaps an SVM classifier that is also popular in AOI (as mentioned in section 2.2.3).

4.2.3 PCA Improvements

As mentioned in section 3.2.2, we processed our raw features using the PCA algorithm (described in section 1.3.4) and reduced their number to 30 without any significant performance loss. The feature transformation also greatly improved the accuracy of our classification model, which we tested by running the same experiment, only without PCA – using the original 25088 features. This configuration reaches only a 76.12 % accuracy and the results in table 4.5 show that in this case, the model was even less able to detect tombstones and achieved the best accuracy by abandoning the class altogether, resulting in a 0 % recall.

4.2.4 Future Improvements

The hardware and software developed for this thesis show some positive results, but to be used in a production environment, we would need to make several improvements for it to function in a manufacturing workflow and perform with high accuracy. In the following section, we will go over some of the most important changes to consider in future work.



(a) Image captured using our illumination system

(b) Image captured using a standard illumination system [11]

Figure 4.1 Illumination system comparison

4.2.4.1 Positioning System

Our image capturing station (described in section 3.1.4) is useful to gather data for our experiment, but for real-world use, especially on larger PCBs, an automated positioning system would be helpful to take images exactly centered to each SMD part.

We think it is possible to attach our illumination system (with some modifications) and camera to a pick-and-place machine, a part of SMT equipment already used by many small scale-manufacturers and hobbyists thanks to the OpenPnP project [64].

4.2.4.2 Illumination System

Our illumination system proved to extract valuable information image, but we think it has one flaw that could be addressed in a new iteration. As shown in fig. 4.1, a standard illumination product, commonly used in the industry, produces an image with more color blending and smooth transitions across the three colors, while our solution produces discrete color regions with some dark regions in between. We think this issue could be solved by adding more LED layers to the illumination system, filling in the space between the three arrays in our dome solution.

4.2.4.3 Camera

The comparison in fig. 4.1 clearly shows that aside from illumination issues, our system also has a meaningful difference in the resolution of the captured RoI. This could be solved by using a higher resolution camera. However, a more sensible solution would be a different lens that captures a narrower field of view with more detail.

In the current configuration, most of the image area captured by our system is not used for inspection, and even if we used a larger board as a sample, we would not use the outer parts of the image, as the correct illumination is only valid for the parts directly in the center of the image.

4.2.4.4 Localization

The solder pad localization method we used in this thesis was tailored directly to our experiment so that a real-world application would need a completely different solution.

As discussed in section 2.2.1, there are many approaches, but most of them are difficult to implement. Only the mentioned YOLO solution seems easier to use in an application, but it would require a much larger dataset with manually labeled data.

4.2.4.5 Classification Using Fine-tuned VGG-16

As we mentioned in section 1.3.3, a more common way of leveraging a pre-trained model in a transfer learning approach is reusing the ConvNet even for classification with some fine-tuning. This would entirely eliminate our feature extraction and classification methods, and it could be a more effective system.

Chapter 5
Conclusion

In this thesis, we set out to create a prototype of an AOI system for inspecting soldered SMD components targeted at small-scale manufacturing environments. First, we analyzed the SMT process, the defects that can occur during production and methods used to detect them. We designed and built a complete inspection hardware system that we later used to build a dataset of SMD component images.

We used this dataset to develop a software solution that localizes solder joints, which are then processed using a transfer learning method to extract valuable information. This data was then used in a Random Forest classifier to identify soldering defects. Our approach experimentally demonstrates that it is possible to achieve the inspection goal with a significantly less expensive hardware solution that can be easily reproduced.

There is, however, many improvements to be made before this system is viable in a real-world application.

Bibliography

1. KOSHAL, D. *Manufacturing engineer's reference book*. Elsevier, 2014.
2. KEENS, M. *Surface Mount Process* [online]. Surface Mount Process, 2015 [visited on 2021-04-11]. Available from: `https://www.surfacemountprocess.com/`.
3. ELECTRONICS NOTES. *PCB Assembly and Production Process* [online]. Electronics Notes, 2019 [visited on 2021-04-11]. Available from: `https://www.electronics-notes.com/articles/constructional_techniques/pcb-assembly-process-manufacture/automated-pcb-assembly-manufacture.php`.
4. JANÓCZKI, M.; BECKER, Á.; JAKAB, L.; GRÓF, R.; TAKÁCS, T. Automatic Optical Inspection of Soldering. In: MASTAI, Y. (ed.). *Materials Science*. Rijeka: IntechOpen, 2013, chap. 16. Available from DOI: `10.5772/51699`.
5. OKURA, T.; KANAI, M.; OGATA, S.; TAKEI, T.; TAKAKUSAGI, H. Optimization of solder paste printability with laser inspection technique. In: *Seventeenth IEEE/CPMT International Electronics Manufacturing Technology Symposium. 'Manufacturing Technologies - Present and Future'*. 1995, pp. 361–365. Available from DOI: `10.1109/IEMT.1995.526187`.
6. HORIJON, J. L.; AMSTEL, W. D. van; COUWELEERS, F. C. M.; LIGTHART, W. C. Optical system of an industrial 3D laser scanner for solder paste inspection. In: DESCOUR, M. R.; HARDING, K. G.; SVETKOFF, D. J. (eds.). *Three-Dimensional and Unconventional Imaging for Industrial Inspection and Metrology*. SPIE, 1996, vol. 2599, pp. 162–170. Available from DOI: `10.1117/12.230375`.
7. LATHROP, R. R. Solder paste print qualification using laser triangulation. *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part C*. 1997, vol. 20, no. 3, pp. 174–182. Available from DOI: `10.1109/3476.649437`.
8. STEFANO, L. D.; BOLAND, F. Solder-paste inspection by structured light methods based on phase measurement. In: WU, F. Y.; YE, S. (eds.). *Automated Optical Inspection for Industry*. SPIE, 1996, vol. 2899, pp. 702–713. Available from DOI: `10.1117/12.253007`.
9. WU, X.; CHUNG, W. K.; CHENG, J.; TONG, H.; XU, Y. A Parallel-Structure Solder Paste Inspection System. *IEEE/ASME Transactions on Mechatronics*. 2009, vol. 14, no. 5, pp. 590–597. Available from DOI: `10.1109/TMECH.2009.2009638`.

10. KOH YOUNG TECHNOLOGY INC. *True 3D AOIs to Improve your Production Quality* [online]. Koh Young Technology inc., 2018 [visited on 2021-04-17]. Available from: `https://www.kohyoung.com/en/news/board_detail/356`.

11. WU, F.; ZHANG, X. An inspection and classification method for chip solder joints using color grads and Boolean rules. *Robotics and Computer-Integrated Manufacturing* [online]. 2014, vol. 30, no. 5, pp. 517–526 [visited on 2021-04-21]. ISSN 0736-5845. Available from DOI: `10.1016/j.rcim.2014.03.003`.

12. LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 1998, vol. 86, no. 11, pp. 2278–2324. ISSN 1558-2256. Available from DOI: `10.1109/5.726791`.

13. KARPATHY, A. *Convolutional Neural Networks (CNNs / ConvNets)* [online]. Stanford University, 2015 [visited on 2021-06-20]. Available from: `https://cs231n.github.io/convolutional-networks/`.

14. SIMONYAN, K.; ZISSERMAN, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]* [online]. 2015 [visited on 2021-06-19]. Available from: `http://arxiv.org/abs/1409.1556`. arXiv: 1409.1556.

15. RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M.; BERG, A. C.; FEI-FEI, L. ImageNet Large Scale Visual Recognition Challenge [online]. 2014 [visited on 2021-06-20]. Available from: `https://arxiv.org/abs/1409.0575v3`.

16. LOUKADAKIS, M.; CANO, J.; O'BOYLE, M. *Accelerating Deep Neural Networks on Low Power Heterogeneous Architectures* [online]. Manchester, UK, 2018 [visited on 2021-06-26]. Available from: `https://eprints.gla.ac.uk/183819/`.

17. LIN, Y.-P.; JUNG, T.-P. Improving EEG-Based Emotion Classification Using Conditional Transfer Learning. *Frontiers in Human Neuroscience* [online]. 2017, vol. 11, p. 334 [visited on 2021-06-25]. ISSN 1662-5161. Available from DOI: `10.3389/fnhum.2017.00334`.

18. SHIN, H.-C.; ROTH, H. R.; GAO, M.; LU, L.; XU, Z.; NOGUES, I.; YAO, J.; MOLLURA, D.; SUMMERS, R. M. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE transactions on medical imaging*. 2016, vol. 35, no. 5, pp. 1285–1298. ISSN 1558-254X. Available from DOI: `10.1109/TMI.2016.2528162`.

19. KIM, S.; KIM, W.; NOH, Y.-K.; PARK, F. C. Transfer learning for automated optical inspection. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. 2017, pp. 2517–2524. Available from DOI: `10.1109/IJCNN.2017.7966162`. ISSN: 2161-4407.

20. NATARAJAN, V.; HUNG, T.-Y.; VAIKUNDAM, S.; CHIA, L.-T. Convolutional networks for voting-based anomaly classification in metal surface inspection. In: *2017 IEEE International Conference on Industrial Technology (ICIT)*. 2017, pp. 986–991. Available from DOI: `10.1109/ICIT.2017.7915495`.

21. DAI, W.; MUJEEB, A.; ERDT, M.; SOURIN, A. Soldering defect detection in automatic optical inspection. *Advanced Engineering Informatics* [online]. 2020, vol. 43, p. 101004 [visited on 2021-04-21]. ISSN 1474-0346. Available from DOI: `10.1016/j.aei.2019.101004`.

22. JOLLIFFE, I. T. *Principal Component Analysis*. New York, NY: Springer New York, 1986. ISBN 978-1-4757-1904-8.

23. SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* [online]. 2019, vol. 6, no. 1, p. 60 [visited on 2021-06-26]. ISSN 2196-1115. Available from DOI: `10.1186/s40537-019-0197-0`.

24. BREIMAN, L.; FRIEDMAN, J.; STONE, C. J.; OLSHEN, R. A. *Classification and Regression Trees*. Chapman Hall/CRC, 1984. ISBN 9780412048418.

25. WITTEN, I. H. *Data mining : practical machine learning tools and techniques*. Burlington, MA: Morgan Kaufmann, 2011. ISBN 978-0-12-374856-0.

26. BREIMAN, L. Random Forests. *Machine Learning* [online]. 2001, vol. 45, no. 1, pp. 5–32 [visited on 2021-06-26]. ISSN 1573-0565. Available from DOI: `10.1023/A:1010933404324`.

27. KUNTE, O. *Automatic Optical Inspection of Printed Circuit Boards*. Prague, 2018. Available also from: `http://hdl.handle.net/10467/74054`. MA thesis. Czech Technical University in Prague, Computing and Information Centre. Supervised by R. ŠKOVIERA.

28. CORPORATION, O. *VT-S730 Industries Highest Capability Full 3D-AOI System* [online]. Omron Corporation, 2021 [visited on 2021-06-26]. Available from: `https://inspection.omron.eu/en/products/vt-s730`.

29. SIMECEK, O. *TRI White Paper – Introduction to AOI Technology* [online]. Test Research, Inc., 2013 [visited on 2021-06-20]. Available from: `https://www.circuitnet.com/news/uploads/2/WP_TRI_Introduction_to_AOI_CN.pdf`.

30. CAPSON, D.; ENG, S.-K. A tiered-color illumination approach for machine inspection of solder joints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1988, vol. 10, no. 3, pp. 387–393. Available from DOI: `10.1109/34.3902`.

31. LU, S.-L.; ZHANG, X.-M.; KUANG, Y.-C. Optimized design of an AOI illuminator. In: *2007 International Conference on Wavelet Analysis and Pattern Recognition*. 2007, vol. 2, pp. 924–928. Available from DOI: `10.1109/ICWAPR.2007.4420801`.

32. KIM, T.-H.; CHO, T.-H.; MOON, Y. S.; PARK, S. H. Visual inspection system for the classification of solder joints. *Pattern Recognition* [online]. 1999, vol. 32, no. 4, pp. 565–575 [visited on 2021-04-21]. ISSN 0031-3203. Available from DOI: `10.1016/S0031-3203(98)00103-4`.

33. XIONG, G.; MA, S.; NIE, X.; TANG, X. New multicolor illumination system for automatic optical inspection. In: ZHANG, Y.; SASIAN, J. M.; XIANG, L.; TO, S. (eds.). *5th International Symposium on Advanced Optical Manufacturing and Testing Technologies: Optical Test and Measurement Technology and Equipment*. SPIE, 2010, vol. 7656, pp. 1038–1046. Available from DOI: `10.1117/12.863984`.

34. KO, K. W.; CHO, H. S. Solder joints inspection using a neural network and fuzzy rule-based classification method. *IEEE Transactions on Electronics Packaging Manufacturing.* 2000, vol. 23, no. 2, pp. 93–103. ISSN 1558-0822. Available from DOI: 10.1109/6104.846932. Conference Name: IEEE Transactions on Electronics Packaging Manufacturing.

35. KIM, J.; CHO, H. Neural network-based inspection of solder joints using a circular illumination. *Image and Vision Computing* [online]. 1995, vol. 13, no. 6, pp. 479–490 [visited on 2021-04-21]. ISSN 0262-8856. Available from DOI: 10.1016/0262-8856(95)94381-9.

36. HONGWEI, X.; XIANMIN, Z.; YONGCONG, K.; GAOFEI, O. Solder Joint Inspection Method for Chip Component Using Improved AdaBoost and Decision Tree. *IEEE Transactions on Components, Packaging and Manufacturing Technology.* 2011, vol. 1, no. 12, pp. 2018–2027. Available from DOI: 10.1109/TCPMT.2011.2168531.

37. PANG, G. K.; CHU, M.-H. Automated optical inspection of solder paste based on 2.5D visual images. In: *2009 International Conference on Mechatronics and Automation.* 2009, pp. 982–987. ISSN 2152-744X. Available from DOI: 10.1109/ICMA.2009.5246351.

38. CRISPIN, A. J.; RANKOV, V. Automated inspection of PCB components using a genetic algorithm template-matching approach. *The International Journal of Advanced Manufacturing Technology* [online]. 2007, vol. 35, no. 3, pp. 293–300 [visited on 2021-06-16]. ISSN 1433-3015. Available from DOI: 10.1007/s00170-006-0730-0.

39. HAO, X.; LI, W.; SUN, Z.; ZHU, S.; YAN, S.; ZHAO, Z. Detection of ball grid array solder joints based on adaptive template matching. *International Journal of Heat and Technology* [online]. 2018, vol. 36, no. 1, pp. 189–194 [visited on 2021-06-16]. ISSN 03928764. Available from DOI: 10.18280/ijht.360125.

40. SA-NGUANNAM, A.; SRINONCHAT, J. Analysis Ball Grid Array defects by using new image technique. In: *2008 9th International Conference on Signal Processing.* 2008, pp. 785–788. Available from DOI: 10.1109/ICOSP.2008.4697247. ISSN: 2164-523X.

41. WU, W.-Y.; CHEN, C.-C. A system for automated BGA inspection. In: *IEEE Conference on Cybernetics and Intelligent Systems, 2004.* 2004, vol. 2, pp. 786–790. Available from DOI: 10.1109/ICCIS.2004.1460688.

42. ABDELHAMEED, M. M.; AWAD, M. A.; ABD EL-AZIZ, H. M. A robust methodology for solder joints extraction. In: *2013 8th International Conference on Computer Engineering Systems (ICCES).* 2013, pp. 268–273. Available from DOI: 10.1109/ICCES.2013.6707217.

43. GAO, H.; JIN, W.; YANG, X.; KAYNAK, O. A Line-Based-Clustering Approach for Ball Grid Array Component Inspection in Surface-Mount Technology. *IEEE Transactions on Industrial Electronics.* 2017, vol. 64, no. 4, pp. 3030–3038. ISSN 1557-9948. Available from DOI: 10.1109/TIE.2016.2643600. Conference Name: IEEE Transactions on Industrial Electronics.

44. HANI, A. F. M.; MALIK, A. S.; KAMIL, R.; THONG, C.-M. A review of SMD-PCB defects and detection algorithms. In: *Fourth International Conference on Machine Vision (ICMV 2011): Computer Vision and Image Analysis; Pattern Recognition and Basic Technologies* [online]. International Society for Optics and Photonics, 2012, vol. 8350, 83501P [visited on 2021-04-21]. Available from DOI: `10.1117/12.920531`.

45. SONG, K.; YAN, Y. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science* [online]. 2013, vol. 285, pp. 858–864 [visited on 2021-06-18]. ISSN 0169-4332. Available from DOI: `10.1016/j.apsusc.2013.09.002`.

46. SANKARAN, V.; CHARTRAND, B.; LILLARD, D.; EMBRECHTS, M.; KRAFT, R. Automated inspection of solder joints-a neural network approach. In: *Seventeenth IEEE/CPMT International Electronics Manufacturing Technology Symposium. 'Manufacturing Technologies - Present and Future'.* 1995, pp. 232–237. Available from DOI: `10.1109/IEMT.1995.526120`.

47. WU, H.; ZHANG, X.; XIE, H.; KUANG, Y.; OUYANG, G. Classification of Solder Joint Using Feature Selection Based on Bayes and Support Vector Machine. *IEEE Transactions on Components, Packaging and Manufacturing Technology.* 2013, vol. 3, no. 3, pp. 516–522. ISSN 2156-3985. Available from DOI: `10.1109/TCPMT.2012.2231902`. Conference Name: IEEE Transactions on Components, Packaging and Manufacturing Technology.

48. HAO, W.; XIANMIN, Z.; YONGCONG, K.; GAOFEI, O.; HONGWEI, X. Solder joint inspection based on neural network combined with genetic algorithm. *Optik* [online]. 2013, vol. 124, no. 20, pp. 4110–4116 [visited on 2021-06-07]. ISSN 0030-4026. Available from DOI: `10.1016/j.ijleo.2012.12.030`.

49. LUO, B.; ZHANG, Y.; YU, G.; ZHOU, X. ANN Ensembles Based Machine Vision Inspection for Solder Joints. In: *2007 IEEE International Conference on Control and Automation.* 2007, pp. 3111–3115. Available from DOI: `10.1109/ICCA.2007.4376934`. ISSN: 1948-3457.

50. WU, H. Solder joint defect classification based on ensemble learning. *Soldering & Surface Mount Technology.* 2017, vol. 29. Available from DOI: `10.1108/SSMT-08-2016-0016`.

51. JONES, E. *SolidPython* [online]. SolidPython, 2021 [visited on 2021-06-23]. Available from: `https://github.com/SolidCode/SolidPython`.

52. OPENSCAD. *OpenSCAD* [online]. OpenSCAD, 2021 [visited on 2021-06-23]. Available from: `https://openscad.org`.

53. BASLER AG. *acA1920-50gc - Basler ace* [online]. Basler AG, 2021 [visited on 2021-06-23]. Available from: `https://www.baslerweb.com/en/products/cameras/area-scan-cameras/ace/aca1920-50gc/`.

54. BASLER AG. *Basler Lens C125-1620-5M-P f16mm - Lens* [online]. Basler AG, 2021 [visited on 2021-06-23]. Available from: `https://www.baslerweb.com/en/products/vision-components/lenses/basler-lens-c125-1620-5m-p-f16mm/`.

55. BASLER AG. *pylon Viewer* [online]. Basler AG, 2021 [visited on 2021-06-23]. Available from: `https://www.baslerweb.com/en/products/software/basler-pylon-camera-software-suite/pylon-viewer/`.

56. BRADSKI, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*. 2000.

57. SUZUKI, S.; BE, K. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*. 1985, vol. 30, no. 1, pp. 32–46. ISSN 0734-189X. Available from DOI: `https://doi.org/10.1016/0734-189X(85)90016-7`.

58. CHOLLET, F. et al. *Keras* [online]. Keras, 2015 [visited on 2021-06-20]. Available from: `https://keras.io`.

59. PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011, vol. 12, pp. 2825–2830.

60. GOOGLE LLC. *Google Colaboratory* [online]. Google LLC, 2021 [visited on 2021-06-20]. Available from: `https://colab.research.google.com/`.

61. NVIDIA CORPORATION. *NVIDIA TESLA P100* [online]. NVIDIA Corporation, 2021 [visited on 2021-06-20]. Available from: `https://www.nvidia.com/en-us/data-center/tesla-p100/`.

62. FREUND, Y.; SCHAPIRE, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences* [online]. 1997, vol. 55, no. 1, pp. 119–139 [visited on 2021-06-25]. ISSN 0022-0000. Available from DOI: `10.1006/jcss.1997.1504`.

63. HASTIE, T.; ROSSET, S.; ZHU, J.; ZOU, H. Multi-class adaboost. *Statistics and its Interface*. 2009, vol. 2, no. 3, pp. 349–360.

64. OPENPNP. *OpenPnP* [online]. OpenPnP, 2021 [visited on 2021-06-20]. Available from: `https://openpnp.org/`.

Contents of the Enclosed Medium

- `README.pdf` instructions needed to reproduce all results using the included code
- `data` directory containing all the input data
  - `dataset` image dataset directory
  - `labels.csv` class labels for the image dataset
  - `stl` directory with segment PCB STL files
- `out` directory for all outputs of the included code
- `domemodel` directory containing 3D modelling code
- `imgproc` directory containing additional code for image processing
- `text` Latex source files of this thesis
- `thesis.pdf` thesis text in the PDF format
- `requirements.txt` Python requirements file for the included code