Modern Architecture for Deep learning-based Automatic Optical Inspection

Johannes Richter*†
*GÖPEL electronic GmbH
Jena, Germany
j.richter@goepel.com

Detlef Streitferdt†
†Faculty of Computer Science and Automation
Technische Universität Ilmenau
Ilmenau, Germany
{johannes.richter | detlef.streitferdt}@tu-ilmenau.de

***Abstract*—The advanced optical inspection of manually placed components on through-hole printed circuit boards demands robust and fast classifiers. To train such classifiers, one needs vast amounts of previously labeled sample images. Datasets like this are currently not available and thus hinder the deployment of deep-learning algorithms in environments like electronics manufacturing. This paper proposes a new architecture, which uses a superposition of active and unsupervised learning to build a problem specific, fully annotated dataset while training a suitable classifier. The system validates human-made annotation by selectively re-asking for a different opinion, to reduce the risk of human error. Our experiments show a simplification of inspection programming in contrast to the existing approaches.**

***Index Terms*—THT, through hole technology, automatic optical inspection, electronics manufacturing, component placement, quality control, deep learning, active learning, classification**

I. Introduction and Motivation

With the rising popularity of renewable energies and electric transportation, the control capabilities of IoT devices require high power electronics (e. g. electric switches, transformers, or converters). Thus, the relevance of Printed Circuit Boards (PCBs) manufactured with Through Hole Technology (THT) increased as well. Products manufactured in a THT process use components with two or more connecting pins which are plugged into pre-drilled holes in the PCB to be then soldered from the opposite side. In contrast to small components assembled in Surface Mounted Technology (SMT), which are mostly placed by fast machines directly onto the solder pads, THT components are much larger and can easily be handled by humans. A lot of electronic manufacturers producing electronics in THT or mixed SMT/THT processes depend on humans to place the components because a specialized machine would be too costly or the number of products manufactured is too low what makes reprogramming of such machines not feasible. In some cases, the use of pick and place machines is impossible, for example, when there is not enough room for the machine to maneuver or when handling components with bent pins. After the components are all placed, the board is fixed to a solder-frame and fed into the soldering process, which is most likely a wave soldering or selective soldering oven. Because humans tend to make more errors than machines do, the inspection of all individual steps becomes very important to minimize production loss. Luckily misplaced or missing THT components can be easily fixed before soldering.

This paper proposes a novel architecture for a multi-user active learning system managing an online dataset to classify manufacturing defects in the domain of electronics with through-hole technology assembled components. The proposed architecture improves the optical inspection of manual THT component placement in production lines. With a new artificial intelligence assisted process, the development and maintenance of inspection programs become more transparent and more straightforward for the user. By using a balanced active-learning approach, the inspection can quickly adapt to changes in the manufacturing process. Balancing the active learning samples makes it harder to intentionally or unintentionally disrupt the training process by passing the wrong labels.

The remainder of this section explains a typical setup with multiple people placing components in the same production line. Distributing the workload of active learning among the workers is beneficial for the training process and the workers. The training dataset gets enriched with samples labeled by different people, and the workers get distracted by the system less often. Using unsupervised learning and cluster analysis helps to reduce the tedious work of labeling samples. By clustering the observed samples, the system can assign labels to them on its own. After this identification of unique characteristics of automatic THT assembly inspection in contrast to the typical surface mounted technology (SMT), the second section discusses the current state of the art. The third section explains the proposed architecture in detail and a real-world application. Finally, part four concludes the findings of this paper and gives an outlook on future work.

A. Optical Inspection of THT Assembly

Solder ovens are the most energy consuming machine in the production line, so their idle time should be reduced to a minimum to lower the production costs. Therefore, it is common practice to have multiple workplaces assembling the same product and feeding into a single oven via the same conveyor belt. Two approaches are typical when inspecting component placement. The first option is to check all boards on the conveyor belt right before soldering. This approach requires a quick inspection system, which can keep pace with

978-1-7281-2607-4/19/$31.00 ©2019 IEEE
DOI 10.1109/COMPSAC.2019.10197

the process to not become the bottleneck and introduces a single point of failure to the whole production chain. Most manufacturers demand the inspection time to be less than 20 seconds per solder-frame. If a solder-frame is classified as faulty, it needs to be ejected from the conveyor belt, thus requiring more handling to route the ejected frames to a repair station or magazine loader. Solder-frames may carry more than a single board.

A single faulty board inevitably ejects the other valid boards as well. However, the most common problem is the fixation of the boards to the solder-frame, which can obstruct the components and make the optical inspection impossible in most cases. Another approach would inspect the assembly right before the fixation is placed onto the boards and the solder-frame feeds onto the conveyor belt. This inspection has to be done at every assembly station but has the advantage that every found mistake could be repaired right away, which removes the need for extra frame handling and dedicated repair stations. Inline inspection systems typically are closed to protect anyone from reaching into moving parts or get irritated by the flashing lights and minimizes the stray light from outside interfering with the inspection. It is unacceptable to place bright flashing lights and moving mechanical parts into or around an assembly station where humans are working. So inspection solutions integrated into the assembly station are typically fixed cameras with constant lights. A missing enclosure makes the system susceptible to false classification due to stray light and human interference.

Checking manual component placement on THT-PCBs is one of the most challenging inspection problems for automatic optical inspection systems to this day. The vast variety of components, manufacturing processes and user dependent preferences and the required high peak throughput of such production-lines put many constraints onto the inspection solution.

II. State of the Art

The automatic optical inspection in electronics manufacturing has sparked new research ideas over the years. Based on our experience electronics with surface mounted devices (SMD) account for the largest market share, so seem the studies involving SMD largely do outnumber the ones targeting THT. The majority of studies about different inspection approaches tend to classify the solder joint as the most critical part. A failure to make a reliable electric connection will most probably result in defect electronics.

There are several recent advances in machine learning algorithms focusing on component placement inspection [1]–[5] by using modern machine vision approaches, robust feature extraction, and deep learning. [1] uses genetic programming to evolve inspection programs for printed circuit boards and to detect defects in the placement of components. The evolved program combines simple arithmetic and logical operations on the pixel values to a complex decision function. By using a sliding window approach, the decision function processes to the whole image piece by piece. [2] defines misplaced components as a root cause for manufacturing problems in SMD production lines and uses image matching based on SIFT (Scale-Invariant Feature Transform) features. [3] proposed a visual inspection system for automatic pick-and-place machines. To detect misplaced or misaligned components, they used shape features to find the components and could show some invariance to illumination changes. [4] applies several image processing steps before classifying SMD components using convolutional neural networks. [5] uses ORB (Oriented FAST and rotated BRIEF) [6] object detection, a modern alternative to SIFT, to find placed components on printed circuit boards. The algorithm is optimized to run on a single board computer. [7] proposed an inspection system solely for THT solder joints. The approach uses machine learning to gain some invariance to noise and changing illumination. None of these approaches addresses the vast variety of THT components and the amount of training data needed to train appropriate classifiers.

Massive problem centered datasets can be acquired relatively fast, but labeling them requires much work of human experts. A typical AOI can acquire thousands of images per hour, but manually labeling samples of datasets in decent quality is time-consuming and thus causes additional efforts. Active learning [8] tries to solve this problem by making the labeling process a human-machine interaction. Traditional active learning assumes a perfect human expert with all-time perfect performance. More modern approaches like cost-sensitive active learning [9] or proactive learning [10] assume an uneven cost distribution for different label classes and imperfect human experts. Research has shown that active learning could be beneficial in multiple fields of interest. For example, [11] uses active learning to improve training sets for intrusion detection in cybersecurity. [12] labels samples through active learning to train classifiers for defect detection in civil infrastructures. [13] assumes perfect human experts and focuses on quality classification for solder joints of SMD components. The approach described in this paper addresses this drawback.

III. Proposed Approach

The proposed architecture uses a novel and combined approach of unsupervised pre-training, cluster analysis, and active learning. Because there are no sufficient datasets to train models in the domain of hand placed THT components, this architecture follows an active learning approach and uses a continuously growing online dataset. While iteratively training the model, the system collects samples and builds a representative training dataset for the given problem.

The following section presents the overall architecture, followed by a discussion of each component. Figure 1 outlines the dataflow, and Figure 2 demonstrates component relations. Although this paper assumes a distributed deployment, using a single machine is still entirely possible.

The main component is the training server, as proposed in a previous paper [14]. The server receives unlabeled and labeled samples of objects to classify. In this instance, images

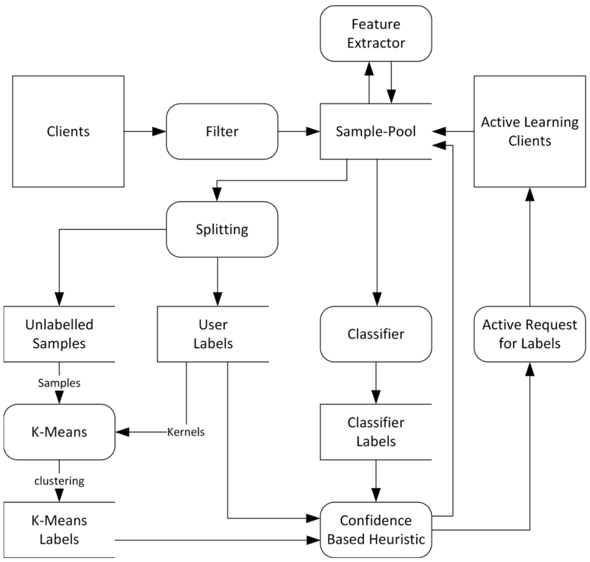

Fig. 1. This figure shows the dataflow diagram of the proposed architecture. Two types of clients feed samples to the dataset, and their features get extracted. After the extraction of features three submodules can assign individual labels to the samples. The confidence-based heuristic selects samples for active learning.

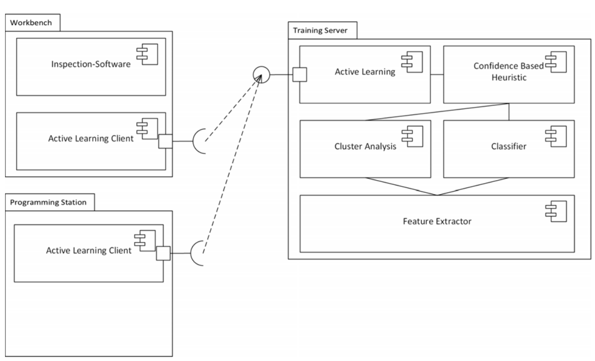

Fig. 2. The component diagram outlines the major components of the system. The relations of the parts as described in the proposed architecture are visible. The training server can also run as a local instance on the workstation computer.

of hand placed THT components. Inspection machines and active learning clients send the samples along with additional information. This information contains a machine identifier, information about the executed inspection step, and in case of the active learning client the assigned label. The server checks all arriving samples for duplicate samples already in its collection. A simple sum of intensity differences and a heuristically adjusted threshold are sufficient for image data. Samples which are too similar to already collected samples get rejected. Rejecting samples helps the underlying models to prevent overfitting to over-represented classes in the dataset and prevents collecting redundant data. The rejection of samples is not done based on extracted feature, because they tend to change in the process of training. By adjusting the threshold for rejecting samples for every class individually, the system collects more samples of otherwise under-represented classes. The server has four submodules. There are feature extraction, classification, cluster analysis, and a confidence based selection for the active learning of unclear samples.

A. Feature Extraction

The first module is of the form of a convolutional autoencoder and extracts the features for the classification and the cluster analysis. Alternatively, a transfer-training approach, as shown in [13], could be used to replace the feature extraction model and eliminate the need to train the extractor. [15] describes autoencoders as a type of neural network copying their input to their output while constraining a hidden layer. With undercomplete autoencoders, as used in this approach, the dimension of the hidden layer is lower than the input dimension. Training the autoencoder on samples from the targeted distribution of data forms essential features in the hidden layers. The following submodules rely on the extracted features. If this module fails to extract meaningful features or the transfer learning assumption fails, the other submodules are not able to assign reliable labels. The result of this would be, that nearly all samples get selected for active learning, which would miss the goal of this proposed architecture.

B. Cluster Analysis

The cluster analysis helps the process with a fast unsupervised way to annotate unlabeled samples based on similarities in their features. The example implementation in this paper uses the K-Means clustering because it is robust and accessible. Any other unsupervised clustering algorithm can replace the K-Means in the proposed architecture.

The cluster analysis is done solely on the extracted feature from the feature extraction module. If the algorithm analyzed the raw image data, only groups of similar appearance in the RGB domain would be identified. K-Means uses a distance metric to sort samples into clusters. A kernel for every cluster defines its position in the data space. The algorithm sorts samples to the cluster with the shortest distance to its kernel. The sense of distance gets lost with higher dimensional data [16], so another step transforms the data for the K-Means algorithm into yet a lower dimension. This transformation can be done using a principal component analysis (PCA). A PCA transforms the variables of a dataset into a new set of variables, called the principal components. These principal components are arranged in such a way, that they are uncorrelated and the first few retain the most of the variation present in all of the original variables [17]. Reducing the dimension of the dataset is done by using the first m principal components of an n-dimensional data set, where $m < n$. The PCA maximizes the variance in the dataset, so the assumption is that the variables with the highest variance are the ones carrying most classification relevant information. Omitting the variables with low variance does not affect the classification in a significant way. This assumption is not always valid and depends on the given problem.

The considerable advantage of the PCA is its ability to work without prior information about the dataset or a model of the data. At this point, the cluster analysis assigns labels to samples with an evident class affiliation. Samples with uncertain clustering or disagreement between the submodules are candidates for active learning and will be labeled manually. After extracting and reducing the features, the actual clustering takes place. The K-Means algorithm needs a start value for the centroid of each class. This centroid is then iteratively moved to its final stable point, and the algorithm terminates. The centroid alone would assign each sample into anonymous classes, so mapping of centroids to class labels is used to assign the actual labels. Typically K-Means centroids are initialized with random values, which would make the mapping of labels to centroids meaningless. So the centroid of every class is initialized with the weighted mean of all samples with manually assigned labels of the same kind.

$$\bar{x} = \frac{\sum_{i=0}^{n} (w_i * x_i)}{\sum_{i=0}^{n} w_i}$$

w_i is the quadratic distance of the sample to the arithmetic mean of all samples with the same label. The weighted mean produces a more reliable centroid than the mean because it is less prone to outliers. After the algorithm terminated, every sample in the dataset has a cluster-analysis-assigned label and a confidence value for the label. The confidence value follows:

$$c_i = 1.0 - \frac{d_{i0}}{d_{i1}}$$

d_{i0} is the quadratic distance of the sample to its assigned centroid and d_{i1} the quadratic distance to its closest neighbor centroid.

C. Classification

The classifier receives the extracted features of the feature extraction module and in contrast to the cluster analysis consumes them as is. This module is highly specific to the inspection task. For a simple classification task, a single neural network receiving the extracted features is sufficient. If the problem requires more than one classification, the submodel can be composed of several independent classifiers receiving the same features. The classifier assigns a class label and a confidence value to each sample in the dataset. The confidence value reaches from 0 to 1, with 1 being the highest confidence.

D. Active Learning

Active learning is used to reassure the assumptions made by the system when combining the already known user labels with the cluster analysis and classifier. When there are multiple clients connected to the proposed server, the workload of manual classification splits between all clients, given that each operator has the needed level of expertise. Being asked to reclassify a sample at an unpredictable time can loosen up the otherwise monotone workflow of manual component placement. The gamification component of this process can even bring some relaxing joy through the expression of progress and achievements to this activity. If the heuristic discussed later in this section selects a sample for active learning, then the sample gets a unique identifier to merge it with the dataset later again. The selected sample is placed onto a stack for the next random client to signal idle. The system removes all other information accumulated in the process up until now not to bias the human expert in his decision. The expert has time to inspect the sample at his own pace and can change the visualization of the sample in such a way that features relevant to him are visible. After inspection, the expert decides to assign one of the possible class labels to the sample. The server receives the assigned label and the unique identifier. At this point, the proposed approach diverges significantly from other approaches as follows. In similar approaches, the human-assigned label is treated as undoubtedly correct and does not get questioned by the system. The proposed system in this paper is designed to take into account, that not every human-assigned label is valid. Upon arrival, the system updates the sample corresponding to the sent and unique identifier with the classification result. An overall user label is formed based on all available labels assigned by human experts. A calculated confidence value reflects the value of the collective human labels for this particular sample. The confidence-metric guarantees (by design) that single human labels cannot reach a top confidence score. The lack of trust in human-assigned labels may introduce some overhead because samples are selected for verification multiple times, but automatically assigned labels, based on the verified human labels, inherit some of their increased value for the dataset.

E. Training

When a large enough amount of samples, at least a couple of hundred per class, is accumulated or added to the dataset, the training server performs a two-phase training of the model. The first phase of the training refines the feature extraction. If transfer learning is used to extract the features from the dataset, then this step is omitted. The samples in the dataset are all used to train the autoencoder without regard to their assigned labels by any of the other submodules. Depending on the inspection task, the training samples can be multiplied by performing some basic image operations on them. For example, shifting, flipping, rotating, slight scaling, or adding some noise. This type of dataset augmentation is a common practice to increase too small datasets. The aim of the first training phase is, of course, the extraction of meaningful features. The assumption with incomplete autoencoders is that once the reconstructed data at the outputs match the input data as close as possible, a meaningful set of features has formed in the hidden layer. The training runs in epochs with mini batches formed from the dataset, until an appropriate metric indicates the similarity between input and output. Such a metric could be the sum of squared distances as used in the filter for new arriving samples. In the first iterations, when the number of samples in the dataset is still low, an alternative criterion can stop this training phase. The training stops after a given number of epochs, to prevent overfitting.

The second phase of training improves the classifier as follows. The classifier assigns each feature vector a single class label. This part of the training process is supervised and thus needs a fully labeled training dataset. The selection heuristic described in the active-learning section assigns each sample a best fitting label based upon the individual labels assigned by each submodule. The training runs in several epochs of multi batches over the whole data set. After an inspection-task-specific amount of steps or if a quality metric reaches the desired level, the training stops. The newly trained feature extractor and classifier are used in the next iteration of the process to refine the dataset. At the same time, the system deploys the new classifiers to the client, to improve the classification at the workstation.

We conducted experiments with experienced AOI operators to validate the improved workflow. The time needed for the initial creation of a new inspection program is less than compared to the classic approach. Each inspected component has just one inspection step instead of multiple individually parametrized ones. The programming of the inspection now only takes marking component locations and defining what components are allowed at said locations. This process can be further sped up by importing component locations from the manufacturing data, if available. Depending on the size of the product and the number of inspected components, a new inspection program can be created in minutes. In our experiments, the time needed to program a sample PCB with the size of $300mm \times 200mm$ and 11 components of 4 different classes took 5-10 minutes.

IV. Conclusion & Future Work

The proposed new architecture uses independent submodules and a decision heuristic to annotate samples with as less user input as possible. It enables users to quickly build training datasets as well as train the classification models on them. The targeted domain for this research is the optical classification of manually placed components in THT production line, but it can quickly be adopted onto other problems as well. The main advantage of this approach over others is that the user comes to satisfying classification result in a short time while leveraging the robustness and classification sharpness of deep learning.

Future work will evaluate further applications for this architecture. Other inspection domains with the same problem of lacking dataset could be candidates for improvements through this approach. Even non-optical inspection could profit from this system. One example would be the acoustic classification of appliances in the end-of-line test. Here are enormous datasets for each tested product needed but not available and unfeasible to manually annotate. Further improvement of the architecture would include a per client statistic and a mechanic to block or flag clients with abnormal behavior like lousy annotations, whether intended or not. Such clients could be left out future training steps or at least their labels could be weighted less.

References

[1] F. Xie, A. Uitdenbogerd, and A. Song, "Detecting pcb component placement defects by genetic programming," *2013 Ieee Congress on Evolutionary Computation (Cec)*, pp. 1138–1145, 2013. [Online]. Available: ⟨GotoISI⟩://WOS:000326235301018

[2] A. Aghamohammadi, M. C. Ang, A. S. Prabuwono, M. Mogharrebi, and K. W. Ng, "Enhancing an automated inspection system on printed circuit boards using affine-sift and triz techniques," *Advances in Visual Informatics*, vol. 8237, pp. 128–137, 2013. [Online]. Available: ⟨GotoISI⟩://WOS:000334104100012

[3] L. F. Bai, X. Q. Yang, and H. J. Gao, "Corner point-based coarse-fine method for surface-mount component positioning," *Ieee Transactions on Industrial Informatics*, vol. 14, no. 3, pp. 877–886, 2018. [Online]. Available: ⟨GotoISI⟩://WOS:000426700600006

[4] Y. G. Kim, D. U. Lim, J. H. Ryu, and T. H. Park, "Smd defect classification by convolution neural network and pcb image transform," *Proceedings on 2018 Ieee 3rd International Conference on Computing, Communication and Security (Icccs)*, pp. 180–183, 2018. [Online]. Available: ⟨GotoISI⟩://WOS:000459208100033

[5] R. Pramudita and F. I. Hariadi, "Development of techniques to determine object shifts for pcb board assembly automatic optical inspection (aoi)," *2018 International Symposium on Electronics and Smart Devices (Isesd 2018): Smart Devices for Big Data Analytic and Machine Learning*, pp. 73–76, 2018. [Online]. Available: ⟨GotoISI⟩://WOS:000458316600015

[6] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," 2011.

[7] C. L. S. C. Fonseka and J. A. K. S. Jayasinghe, "Implementation of an automatic optical inspection system for solder quality classification of tht solder joints," *Ieee Transactions on Components Packaging and Manufacturing Technology*, vol. 9, no. 2, pp. 353–366, 2019. [Online]. Available: ⟨GotoISI⟩://WOS:000457629400018

[8] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Report, 2009.

[9] G. Krempl, D. Kottke, and V. Lemaire, "Optimised probabilistic active learning (opal)," *Machine Learning*, vol. 100, no. 2-3, pp. 449–476, 2015. [Online]. Available: ⟨GotoISI⟩://WOS:000359747100012

[10] P. Donmez and J. G. Carbonell, "Proactive learning: cost-sensitive active learning with multiple imperfect oracles," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, Conference Proceedings, pp. 619–628.

[11] S. McElwee, "Active learning intrusion detection using k-means clustering selection," *Southeastcon 2017*, 2017. [Online]. Available: ⟨GotoISI⟩://WOS:000403420500123

[12] C. Feng, M. Y. Liu, C. C. Kao, and T. Y. Lee, "Deep active learning for civil infrastructure defect detection and classification," *Computing in Civil Engineering 2017: Information Modelling and Data Analytics*, pp. 298–306, 2017. [Online]. Available: ⟨GotoISI⟩://WOS:000425807900036

[13] W. Dai, A. Mujeeb, M. Erdt, and A. Sourin, "Towards automatic optical inspection of soldering defects," in *2018 International Conference on Cyberworlds (CW)*. IEEE, Conference Proceedings, pp. 375–382.

[14] J. Richter, D. Streitferdt, E. Rozova, and M. Kirchhoff, "Improving the optical inspection of through hole resistors with additional spectral illuminations," *2017 8th Ieee Annual Information Technology, Electronics and Mobile Communication Conference (Iemcon)*, pp. 273–277, 2017. [Online]. Available: ⟨GotoISI⟩://WOS:000418335300049

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ser. Deep Learning, 2016. [Online]. Available: ⟨GotoISI⟩://WOS:000412476200021

[16] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," *Database Theory - Icdt 2001, Proceedings*, vol. 1973, pp. 420–434, 2001. [Online]. Available: ⟨GotoISI⟩://WOS:000171312300027

[17] I. Jolliffe, *Principal component analysis*. Springer, 2011.