AN ACTIVE LEARNING APPROACH FOR DETECTING CUSTOMER INDUCED DAMAGES IN MOTHERBOARDS WITH DEEP NEURAL NETWORKS

Lucas Cabral, Victor Farias, Lucas Sena,
Iago Chaves, Joao Paulo Pordeus, Joao Pedro Santiago,
Diego Sa, Javam Machado, Joao Paulo Madeiro
Computer Science Department, Universidade Federal do Ceara
{lucas.cabral, victor.farias, lucas.sena, iago.chaves, joao.pordeus,
pedro.santiago, diego.sa, javam.machado, jpaulo.vale}@lsbd.ufc.br

Abstract – Identifying Customer Induced Damage (CID) is a key part in warranty programs of electronics manufacturers. CID is defined as any damage in the unit performed by an unauthorized person including the customer in a Printed Circuit Board (PCB). In such cases, damaged units are not covered by warranty. The inspection of CIDs is usually performed by humans which may be costly and error prone. Modern computer vision techniques for object detection using deep neural networks can automatically and accurately detect CIDs on PCBs. The training of such networks requires a large labeled dataset of image examples of CIDs. Daily, hardware factories and repair centers generate hundreds of unlabeled images. Labeling them manually is laborious and time-consuming. Therefore, it is crucial to label the minimum amount of images such that the trained neural network can achieve comparable accuracy as if it were trained with the whole dataset. To this end, we propose an active learning approach that selects the most informative images for the object detector. For that, our approach is based on the uncertainty of the object detector, i.e., it selects new images based on class probability distribution given by the object detector. Also, we tackle some challenges that are intrinsic to this problem: i) it is a multiclass object detection problem since there are many types of defects; ii) there is a class accuracy imbalance; iii) there is a focus on recall, e.g. false positives are less harmful than false negatives, and iv) there are many images with no object which should not be selected for labeling. We evaluate this approach by using it to iteratively sample data, train and evaluate a model, and compare it with randomly sampled data. The results show that our method consistently outperforms random sampling by an average margin of 21.6%, proving to be a viable alternative for reducing the labeling cost and increasing detection accuracy in this domain.

Keywords – Customer Induced Damage, Printed Circuit Board (PCB), Deep Learning, Computer Vision.

1 Introduction

Printed Circuit Boards (PCB) are crucial components in many electronic devices. It consists of conductive and insulating layers that integrate other electronic components. Given its fundamental role, the inspection of PCB is a critical task that is performed by electronic manufacturers aiming for quality control. It is well known that the inspection of PCB is mainly done manually by humans [1]. Since the task is time-consuming and the damages may be very small or subtle, manual inspection is considered labor-intensive and subject to errors [2]. For these reasons, many researchers have been working on automatic inspection methods for PCB [1–8]. While most works address the automatic inspection of damages in PCB originated during manufacturing process, another relevant kind of damage is Costumer Induced Damage (CID). This damages occur due misuse of the product, and their inspection impact in the cost of warranties and repairs processes of hardware manufacturers.

First approaches on automatic PCB damage detection were based on various classical image processing techniques such as template matching [7], image subtraction [8], and mathematical morphology-based features [1]. More recently, remarkable progress has been made by the use of deep neural networks for object detection. More specifically, these approaches are based on convolutional networks [2, 3], YOLO (You-Only-Look-Once, [9]) [4] and Transformers [5, 6].

Training deep learning networks requires a dataset of images labeled with the category of the damage and its bounding box. The availability of unlabeled images collected from factories and repair centers is abundant, which is not the case for labeled images. Building a labeled dataset for damage detection demands humans with specialized knowledge to annotate a unlabeled pool of images, which is a costly and laborious process.

Reducing costs is a subject of great interest to the industry. Therefore, active learning (also called optimal experiment design) is subfield of machine learning [10] where the goal is to label the least number of images of the unlabeled pool so that a learning algorithm obtain a similar accuracy compared to the methodology where we label the entire unlabeled pool and feed it to the learning algorithm. An active learning approach poses queries to the human annotator for labeling images that are most informative.

Uncertainty-based active learning approaches take an object detector that predicts the class distribution probability per object prediction to select new images for labeling. In the CID detection problem, some challenges arise: i) there are many type of damages, i.e., it is a multiclass object detection problem; ii) there is a class accuracy imbalance, some classes have lower accuracy than the others; iii) it is preferable to have a false positive, where a specialist can double check if the predicted damage is true, than a false negative where the hardware is shipped to the client with damage which leads to customer dissatisfaction and cost of repairing. Thus, there is a focus on recall; iv) many images have no visible damage, i.e., no objects to detect and therefore should be avoided to be selected for annotation.

General purpose active learning approaches based on uncertainty for generic problems have been proposed in the literature [11, 12]. Also, many works tackle specific related computer vision problems as façade defects detection [13], surface defect detection [14] and chip welding short circuit defect detection [15].

To this end, we present our active learning approach for CID detection in PCB using deep learning methods for object detection. Our approach is an iterative process with a human annotator in the loop for labeling images. It is composed by two main stages: i) Image Scoring stage where, given an image, the uncertainty scores of the object predictions for this image are aggregated, grouped by object classes, and the score of the class with highest uncertainty is used to create the priority score, for prioritize images with many uncertain objects predictions regarding one class; ii) Image Selection stage, where our approach assign images to classes with highest uncertainty and takes into account the class recall imbalance to select more images from the classes that have a smaller recall. We named this strategy Categorical Uncertainty Sampling Active Learning (CUSAL). Note that our approach is object detector-agnostic, we do not require modifications in the base object detector, it is just required that it computes the class probability distribution per object prediction.

We evaluate the effectiveness of our method in a loop of training and testing models with state-of-the-art architecture, where at each iteration we add a fixed quantity of data sampled with our method to the training set. As a baseline, we do a similar loop but with randomly sampled images, simulating the developing of a machine learning model without active learning. We compare the evolution of the learning curves. The dataset used in this work is a real-world set of images from a large hardware manufacturer repair center of motherboards. The images were collected under diverse conditions of illumination, background, PCB position and orientation, making it a challenging problem. The experimental results show that our method is a promising approach, as it consistently outperforms random sampling in terms of the standard object detection performance metric, mean Average Precision (mAP), by an average of 21.6%. The recall improved by an average of 39%, with a improvement of 60% for the class with lower accuracy. It also has considerably better result in sampling useful images, i.e., images with visible damage, which would make a relevant impact in reducing annotation efforts.

The remainder of this paper is as follows. The main concepts are presented in Section 2. Section 3 presents the related works found in literature. In Section 4 we describe our proposed method. Section 5 presents our experimental evaluation and the discussion of the results. Finally, Section 6 presents the conclusion and future research directions.

2 Theoretical Background

2.1 Customer Induced Damage

Damages in electronic components, such as motherboards, can lead to some malfunction or failure on their devices. Those damages may be originated in different ways. Often related works treat damages originated by manufacturing process. The electrical component companies' facilities avoid those damages through the quality control process. But there is another industrial common way to originate damage that is the Customer Induced Damage (CID). CID is defined as any damage in the unit performed by an unauthorized person, including the customer. Examples of types of CID include cracks, scratches, pin bent, liquid spillage and burns.

Such kind of damages may lead the companies to spend a huge amount of money on warranties, and because of that, it is important to identify CID. A default standard for the industry is that the warranty does not cover CID and thus, CID detection may help the company to save money.

2.2 Deep Neural Networks

A Deep Neural Network (DNN) is a stack of standard neural networks, such as a feed-forward neural network or even a convolutional network. This deep version outperforms the shallow networks [16] in the object identification and classification problem. The DNNs are able to locate and classify precisely objects in images.

Commonly, DNNs for object detection and segmentation are divided into the backbone and prediction network. The backbone is an FPN-like DNN (Feature Pyramid Networks) [17]. It recognizes the image features and generates the feature map. It is usually trained with a general-purpose image dataset, such as ImageNet [18]. Then, the feature map is passed to the prediction network, which predicts the class and the bounding box containing the detected object. Bounding boxes are described by the (x, y) coordinates of the upper left corner, the width and the height.

In this work we investigate as prediction head the Mask-RCNN [19], using as backbone network the Swin architecture [20].

2.2.1 Mask R-CNN

The Mask R-CNN is a framework of Deep Neural Networks proposed by Facebook AI Research [19]. Its objective is to detect objects in images and generate high-quality segmentation. This approach extends Faster R-CNN [21] and applies a new branch of segmentation for object mask prediction. The process is applied parallel with the existing process to localize the bounding box.

Once the model has the feature maps from the FPN, it feeds the Region Proposal Network (RPN), a lightweight binary classifier. The RPN proposes regions that may contain objects. After that, another network, called RoIAlign, receives the proposed regions as input and assigns them to areas of the feature map, generating the bounding boxes. Moreover, now it fed the fully connected layers to make the classification. The RoIAlign output also fed the CNNs to process the segmentation. Currently, this technique is state-of-the-art on Real-Time object detection on COCO minival benchmark [22].

2.2.2 Swin

This backbone network is a new method of Transformers used in the computer vision field proposed by Microsoft Research Asia [20]. The purpose is to develop a new general backbone to vision tasks adapting Transformer from language domain. The method uses hierarchical Transformer computing with shifted windows. In this schema, the self-attention is computed to non-overlapping local windows while allowing for cross-window connection. It brings excellent efficiency compared to the default. Currently, this technique is state-of-the-art on Object Detection and Instance Segmentation on COCO test-dev.

2.3 Evaluation metrics in object detection

In the evaluation of models of object detection some key metrics are used. In this work we focus our attention in two: recall and mAP. The mean Average Precision (mAP) is the most widely used metric to judge performance in object detection task. It provides an overall evaluation of the model's ability to accurately detect objects and localize them in an image.

To compute mAP, the following steps are typically followed. We are first given a list of predictions for each image by the detector. Each ground truth (GT) bounding box in the image is then matched to the highest overlap detection. To qualify as a true positive match, the detection must have the same class as the ground truth and an intersection-over-union (IoU) overlap greater than some threshold. The IoU metric, compares two bounding boxes A and B as $IoU = \frac{|A \cap B|}{|A \cup B|}$. A typical IoU threshold is 0.5, which is referred to as $mAP@50$.

GTs without a correct match are computed as false negatives, and predictions that does not match any GT are false positives. The precision is the ratio of true positives to the sum of true positives and false positives, while the recall is the ratio of true positives to the sum of true positives and false negatives.

For each class in the dataset, one model's predictions are ranked based on confidence scores. A precision-recall curve is then constructed by varying the threshold for accepting a detection as positive. The area under the precision-recall curve is calculated, yielding the average precision (AP) for each class. AP represents the model's performance in terms of precision and recall for a specific class. Finally, the AP values for all classes are averaged to obtain the mAP, which gives a single performance score for the object detection model across all classes.

2.4 Active Learning

Active learning is a machine learning paradigm that aims to maximize model performance while minimizing the labeling effort [10]. It addresses scenarios where labeled data is scarce or expensive to obtain, such as in speech recognition [23], document classification [24], image recognition [25] and object detection [11]. The key principle underlying active learning is to iteratively select the most informative instances from an unlabeled dataset and label them for model training, making it perform better with less data. The instance informativeness refers to the quality of an instance in terms of its potential to improve the learning model's performance. One of the most popular and effective strategies in active learning is uncertainty sampling [26].

Uncertainty sampling is a query approach that leverages the model's ability to quantify its uncertainty in predictions to select samples for labeling. The underlying principle is that the model should prioritize labeling instances about which it is most uncertain, as these instances are expected to provide the most valuable information to improve the model's performance. This uncertainty can be measured in various ways, but three common uncertainty measures are typically used in practice: (1) least confidence sampling [27], (2) margin sampling [28] and (3) entropy sampling.

Least confidence sampling operates under the premise that instances with the lowest confidence in predictions should be labeled first. This strategy selects instances for which the model is least certain about the correct class. These instances can be informative as they are more likely to benefit from additional labeled data to reduce model uncertainty. Margin sampling aims to select instances for which the model assigns multiple class labels with similar confidence, i.e., cases where the predicted probabilities for the top candidate classes are close to each other. These instances are considered more challenging and informative for model improvement, as the classifier is uncertain about the correct classification. Entropy sampling leverages the concept of information entropy to gauge uncertainty. For each instance, the entropy of the predicted class probabilities is calculated. Higher entropy values indicate greater uncertainty, as they signify that the model's predictions are more evenly distributed across multiple classes. Consequently, instances with higher entropy are prioritized for labeling. The choice of the specific uncertainty measure often depends on the problem at hand and the characteristics of the dataset.

Another important active learning strategy is diversity sampling, which focuses on selecting instances for labeling that offer a diverse set of perspectives or information to the model, representing the entire data distribution [29]. The goal is to choose instances that are not only uncertain but also diverse in their content or characteristics. This approach helps ensure that the model receives a well-rounded set of examples, improving its generalization and robustness. Common approaches often rely on quantifying and minimizing the similarity between samples. Practitioners must balance the trade-off between exploration, seeking diverse and challenging instances, and exploitation, focusing on instances where the model's improvement is most critical.

Commonly, in scenarios where active learning is applicable, a large pool of unlabeled data $\mathcal{U}$ is available for selection. In such scenarios, a substantial pool of unlabeled instances is initially gathered. The active learning algorithm then iteratively selects a subset of these instances for labeling, with the goal of maximizing the model's performance. The ability to actively select informative samples for labeling has the potential to significantly reduce the annotation burden and accelerate the training process, making it a valuable technique for a wide range of machine learning tasks.

3 Related Work

3.1 PCB Defect Detection

The field of automatic defect detection in PCBs has gained attention from researchers due its vital role in industry. The work by Crispin et al. [7] addresses the automation of PCB component identification using a genetic algorithm template-matching approach. Other studies have demonstrated the use of the subtraction method [8]. While these proposed approaches can successfully locate defects, they can also incur high costs. A similar issue is addressed in [1], where features based on mathematical morphology are employed. More recently, significant advancements have been achieved by utilizing deep neural networks for object detection [2, 3]. Specifically, convolutional networks have been extensively experimented with in these approaches.

The work of Adibhatla et al. [9] utilizes a YOLO (You-Only-Look-Once, [4]) CNN to locate defects on PCBs. The proposed model is capable of locating defects but does not provide any classification of such defects. A new CNN architecture based on the attention mechanism is proposed in [5]. This novel architecture is used to locate and classify defects on the surface of PCBs and has already demonstrated promising results [6].

However, none of these works address the problem of scarcity of labeled data for training deep learning models or the high cost of image annotation. In [30, 31], the authors used semi-supervised learning techniques to exploit to the available unlabeled data to improve model performance. However, one major challenge in semi-supervised learning is dealing with the quality of the unlabeled data. If the unlabeled data contains noisy or irrelevant examples, which occurs in the scenario addressed in this work, it can negatively impact model performance. Therefore, active learning poses as a promising strategy to improve performance and reduce annotation cost by selecting high quality and informative images for annotation.

3.2 Active Learning Strategies

Applying active learning techniques in deep object detection is currently an open problem, as most works found in literature focused in classification problems. In [11], the authors combined a novel active learning approach for object detection with an incremental learning scheme, enabling the continuous exploration of new unlabeled datasets. Scores from individual detection are aggregated into a score for the whole image, and the margins of predictions are combined to identify valuable instances. The authors proposed several uncertainty based active learning metrics for object detection.

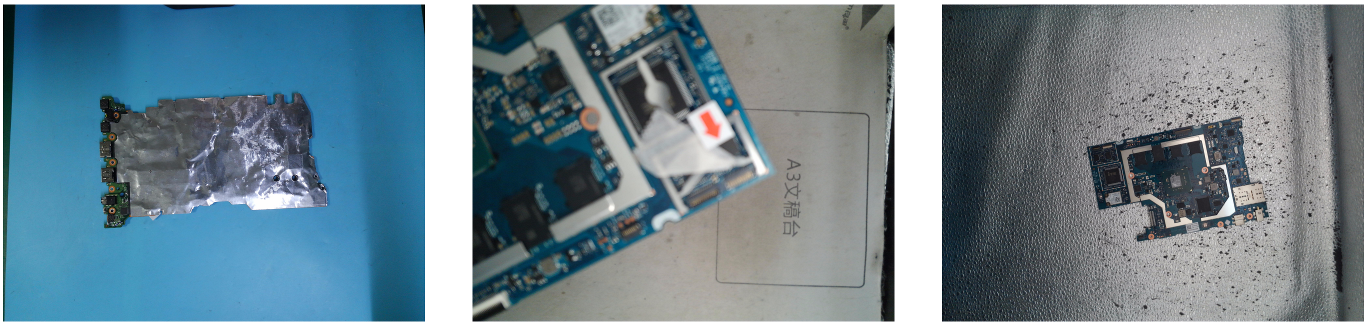
In [12], the authors proposed a two-stage Plug and Play Active Learning (PPAL) algorithm for active learning. Firstly, the proposed method utilizes a category-specific difficulty coefficient that considers classification and location to reweight object uncertainties for uncertainty-based sampling. Subsequently, they measure similarities between images from multiple instances, where each object is paired with its most similar counterpart in another image to calculate per-instance simulation. Model performance is evaluated using Mean Average Precision (mAP). The experiments demonstrate that PPAL exhibits strong generalization capabilities and outperforms previous approaches across multiple architectures and datasets.

Regarding the creation of active learning methods for specific domain tasks, in [32] the authors introduced Active Learning for Automated Visual Inspection of Manufactured Products. The authors compared three active learning approaches and five machine learning algorithms applied to the visual processing of defects using real-world data. The active learning strategies employed were pool-based, stream-based, and committee-based querying. It was found that the MLP model achieved the best performance regardless of the active learning strategy.

In recent years, various approaches have been proposed for defect detection, involving the analysis of different active learning methods [13], [14], [15]. Typically, active learning methods adopt the degree of uncertainty approach and other approaches such as the degree of representativeness and the mean margin method.

In [14], the authors employed an uncertainty sampling approach based on defect image information. Initially, a list of candidate products for annotation was selected using this uncertainty sampling. Then, the sampling scale was estimated using the average margin method. The essence of uncertainty lies in information entropy, which measures the amount of information, with higher information entropy denoting richer information. The results demonstrated the effectiveness of the proposed active learning method.

In this work, we present an active learning approach for PCB damage detection using deep learning methods for object detection. Our method differs from the other presented works by considering specific problems in the PCB damage detection domain: an emphasis on recall, a class accuracy imbalance, an available data pool of unlabeled images with a high percentage of images with no visible damage. To the best of our knowledge, no other work addresses the same issues or uses the same strategy adopted in this work.



(a) Non visible damage, as the PCB is covered

(b) Blurred or unfocused image, with no visible damage

(c) Photography taken from a long distance, making the area of the damage very small

Figure 1: Examples of low quality and non-informative images found in the dataset.

4 Categorical Uncertainty Sampling Active Learning

4.1 Problem definition

As stated previously, in the scenario addressed by this work, a large set of non-annotated images of PCB containing CID is available. Those images were collected from repair centers of a large hardware company. However, no standard procedure was followed in the capture of the photographs, resulting in images with diverse conditions of illumination, zooming, background, position and orientation of PCB.

It is observed that a considerable portion of images lack informativeness for training deep learning models, due the absence of any visible CID or due the low quality of the image, which lowers the signal-noise ratio. Figure 1 illustrate some types of challenging images found in the data. Images without visible CID can be resulted from human error or lack of care in the photo framing. Such images increases the annotation cost, since they demand time and attention of the annotator just to be ultimately discarded, since they do not contribute for the learning of the model.

Other types of low quality images includes images that are blurred, out of focus, have low resolution and zoomed-out images. There is also an imbalance in the quantity of images regarding the different categories of CID. Furthermore, the proportion of low quality images is higher in some categories, making the detection of such categories even harder. Particularly, images of the CID category of pin damage are minority in comparison with other categories and also had a higher proportion of low quality images and images without visible CID. It was verified empirically from initial exploratory experiments that a detection model achieves lower performance regarding this kind of damage. This impacts specially the recall, which has a higher cost in this domain.

Thus, in order to optimize annotation efforts, an active learning method for this scenario must address those issues, selecting the most informative images from a pool with a considerable proportion of inadequate images and with classes that are harder to detect, with focus in recall. Such method should have a higher probability of sampling images with visible objects, reducing the human annotation effort of searching for CID in such images. We highlight that although this described scenario refers to a particular task and dataset, there are many real world scenarios that share the same or similar characteristics: a task of object detection, a large unlabeled data pool with a high proportion of low-quality or non-informative images, a class accuracy imbalance and a focus on recall. Thus, the proposed method could be applied in such scenarios.

4.2 Overview of the method

The proposed Categorical Uncertainty Sampling Active Learning (CUSAL) method is a form of uncertainty sampling adapted for the task of object detection, where we use a previously trained and evaluated model to make predictions on a unlabeled data pool $\mathcal{U}$ and use this predictions to sample k images to be annotated. CUSAL is model-agnostic, as it do not require modifications in the architecture of the object detector, just requiring that it computes the class probability distribution per object prediction.

Our method is divided in two main stages: Image Scoring and Image Selection. The first stage aims to assign a priority score for each image in the unlabeled images pool. The priority score is calculated by summing the uncertainty scores of all predicted objects, grouped per class. We named this value as categorical uncertainty. Then, the priority score assigned to the image is the highest categorical uncertainty. This strategy aims to select images that are representative of each class, which usually is not a characteristic of the task of object detection, since one image may contain objects from multiple classes. In this stage we also assign the class with highest sum of uncertainty to the whole image, as if it was classification task. The second stage aims to select k images, balancing the quantity of images from each assigned class by the recall per class, which is calculated from a test set.

Intuitively, our method aims to select images that a model is least confident regarding one class, which would improve its learning in detecting objects of that class, while also balancing the examples of objects of different classes by their difficulty of detection, inferred by the recall. It is expected that this strategy results in a faster ascending learning curve, as well as reduce the gap of detection performance between classes. We also expect it to reduce the proportion of images with absence of visible damage and low quality images, since a model is likely to made fewer predictions on such images.

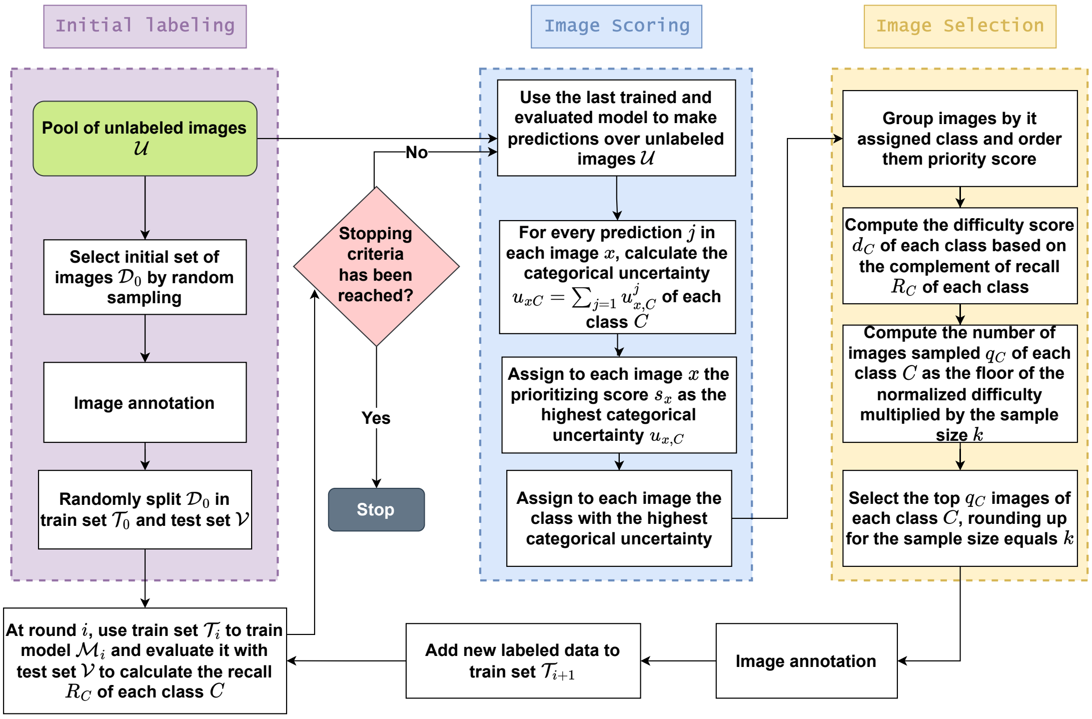

Initial labeling
Image Scoring
Image Selection
Pool of unlabeled images $\mathcal{U}$
Select initial set of images $\mathcal{D}_0$ by random sampling
Image annotation
Randomly split $\mathcal{D}_0$ in train set $\mathcal{T}_0$ and test set $\mathcal{V}$
No
Stopping criteria has been reached?
Yes
Stop
Use the last trained and evaluated model to make predictions over unlabeled images $\mathcal{U}$
For every prediction j in each image x, calculate the categorical uncertainty $u_{xC} = \sum_{j=1} u^j_{x,C}$ of each class C
Assign to each image x the prioritizing score s_x as the highest categorical uncertainty $u_{x,C}$
Assign to each image the class with the highest categorical uncertainty
Group images by it assigned class and order them priority score
Compute the difficulty score d_C of each class based on the complement of recall R_C of each class
Compute the number of images sampled q_C of each class C as the floor of the normalized difficulty multiplied by the sample size k
Select the top q_C images of each class C, rounding up for the sample size equals k
At round i, use train set $\mathcal{T}_i$ to train model $\mathcal{M}_i$ and evaluate it with test set $\mathcal{V}$ to calculate the recall R_C of each class C
Add new labeled data to train set $\mathcal{T}_{i+1}$
Image annotation

Figure 2: Overview of the proposed Categorical Uncertainty Sampling Active Learning method

Our method can be represented as a function $A(k,\mathcal{U}) = \mathcal{S}$, where $\mathcal{S}$ is a sample of k images from $\mathcal{U}$. This function can be applied in a loop, sampling k images at each iteration, with a human annotator in the loop labeling those images, re-training and evaluating a model and using it to sample more images in the next iteration. The cycle finishes when a stop condition is satisfied, which can be the spending of an annotation budget or achieving required level of performance. The complete process is illustrated in Figure 2. The next subsections details each stage thoroughly.

4.3 Initial labeling

As stated before, we assume that a large pool of unlabeled images is available. Before the first iteration of our method, is necessary to randomly select a sample of images from the pool for annotation in order to train and evaluate the first model $\mathcal{M}_0$. Therefore, the selected sample will be divided into a initial train set $\mathcal{T}_0$ and a fixed test set $\mathcal{V}$.

Although the exact size of the initial sample cannot be precised, it must be large enough so the test set is representative of the task, as it will remain fixed during the next iterations. In the initial iteration, the test set can be considerably larger than the train test. After train the first model, it most be evaluated with the test set, calculating the recall of each class. The recall will be used in Image Selection stage.

The train set must be just large enough to allow the first model achieve a minimum performance. If one model has not learned enough it may not make any prediction in images, i.e., all the image is considered background. In such cases our method will fail, since the prioritization score depends on the class probability of predictions. If no prediction is made in none image, they cannot be ordered by priority. Empirically, we found that achieving $mAP@50 > 0$ is enough for our method to work in the initial iteration.

4.4 Image Scoring

The objective of the Image Scoring stage is to assign a priority score and a class for each image from the unlabeled images pool, by exploiting the uncertainty of predictions of a trained and evaluated model. In classification tasks, is straightforward to assign an uncertainty to each input, since there is one uncertainty by input. But in object detection tasks we can have more than one output by input, i.e. it is a one-to-many type of task, where one image can have multiple objects. As so, we need to devise a strategy to use the uncertainty of predictions to the whole image.

We took a similar approach as [11], where the priority is calculated as a aggregation, as the sum or average, of the uncertainty of all predictions of a trained model in one image. However, in order to address the class unbalance accuracy, we sum the uncertainties grouped by class, which we call categorical uncertainty, and chose the priority score as the highest categorical uncertainty. Although in object detection we do not have one class associated by each image, we also make this attribution as it will be necessary for the stage of Image Selection, when the number of sampled images will be balanced by the assigned classes. This strategy should sample the most informative images regarding each class, reducing the gap of performance between classes. The steps of Image Scoring stage are described below:

1. At round i, use a trained and evaluated detection model $\mathcal{M}_i$ to make inference over all images of pool $\mathcal{U}$. On each image x, a set of $Q = \{b_x^1, b_x^2, ..., b_x^Q\}$ bounding boxes are predicted . Each bounding box contains a distribution of probability $\{p_B^j, p_1^j, p_2^j, ..., p_N^j\} \subset b_x^j$ of the predicted bounding box b_x^j belonging to a class $C \subset \{B, 1, 2, ..., N\}$ of N predetermined classes, including the background B.

2. Calculate the uncertainty u_x^j of each prediction b_x^j by the *least confidence sampling* method. This uncertainty is defined as $u_{x,C}^j = 1 - p_C^j$, where $p_C^j = \text{argmax}\{p_1^j, p_2^j, ..., p_N^j\}$, i.e., the maximum confidence score on the class C, excluding the confidence of the background.

3. Calculate the categorical uncertainties of each image x, which are defined as the sum of uncertainties of each class. For each image x there will be a set of categorical uncertainties $\{u_{x1}, u_{x1}, ..., u_{xN}\} \subset x$, where $u_{xC} = \sum_{j=1}^{Q} u_{x,C}^j$.

4. Assign to a image x a priority score s_x equals to the highest categorical uncertainty $s_x = \max\{u_{x1}, u_{x1}, ..., u_{xN}\}$ and assign to the image the class of the highest categorical uncertainty $x_C = \text{argmax}\{u_{x1}, u_{x1}, ..., u_{xN}\}$.

The intuition of the Image Scoring stage is that it will prioritize images with higher uncertainty regarding just one class. We argument that this strategy allows to select the most informative images of each class, which would benefit the goal of improve the general performance and reduce the gap of the classes harder to detect. Differently of the approach taken by [11], where any uncertainty measure could be used as priority score, such as entropy of margin sampling, in our approach we are set to the least confidence, as it allows to assign an uncertainty for each class. In the next stage, our method balances the number of examples of each class.

4.5 Image Selection

In the Image Selection stage, a given number of k previously scored images will be selected for manual annotation. As stated before, it is observed that some classes are potentially harder to be detected than others. As so, more examples of images containing objects of such classes must be selected in the sample in order to improve the capacity of detection in such classes. The Image Selection stage make this balancing using the recall metric. It can be divided in the following steps:

1. At each round i, group all images in subsets G_C of images of the assigned class C during the Image Scoring stage.

2. In each subset G_C, order the images by their prioritizing score s_x.

3. Compute the difficulty score of each class C as $d_C = 1 - R_{C,i}$, where $R_{C,i}$ is the recall of class C, which is previously calculated during evaluation of model $\mathcal{M}_i$. $R_{C,i}$ is calculate for a fixed threshold of confidence t_s.

4. Compute the number of samples of each class C as the floor of normalized difficulty score multiplied by the number k of images: $q_C = \lfloor \frac{k \times d_c}{\sum_{C=1}^{N} d_C} \rfloor$.

5. Select the top q_C images of each respective subset. If the number of selected images is less than k, add the remaining images to the subset of higher q_C, so it becomes $q'_{max} = q_{max} + k - \sum_{C=1}^{N} q_C$.

After the Image Selection stage, the selected sample of k images will be manually annotated by human experts and so can be added to the training set and used to retrain and evaluate a model. As stated before, in this framework, the cycle of sampling, annotation, training and evaluation can be repeated until a desired condition is satisfied.

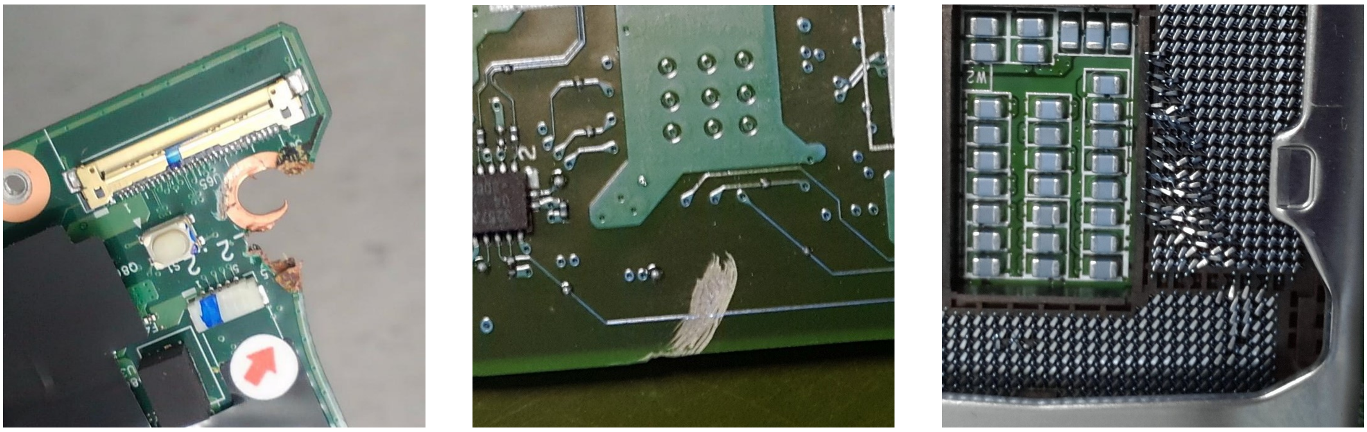
5 Experiments

In this section we detail the experimental setup devised to validate our proposed method, present and discuss the achieved results. Our experiments were performed on Google's Colab Pro+ environment, using a NVIDIA A100-SXM GPU with 40960MB of memory. We used Python 3.10.12 [33], Pytorch 2.0.1 [34], Cuda 11.8 [35] and the framework MMDetection in 2.25.3 version [36].

5.1 Methodology

Our experiments intend to validate our proposed active learning method as an effective alternative for the reduce the labeling cost in the task of developing a CID detector in PCB. For such, we use an annotated dataset to evaluate the performance of a detection model in a loop, where at each iteration we add k images to the train set, sampled using CUSAL. As a baseline for comparison, we do the same loop of evaluation, but adding images to the train set by random sampling. This setup is supposed to be representative of a real-case scenario of training a machine learning without active learning, where images are sampled from a large data pool and annotated at each iteration. This ideal setup was not used due the unavailability of human experts for data annotation. We detailed the active learning loop as follows:

1. From a annotated dataset $\mathcal{D}$, randomly select 20% of the images to be the test set $\mathcal{V}$ that will be used for evaluation. The remaining 80% will be used as it was an unlabelled pool of images $\mathcal{U}_0$ from where data will be sampled at the first iteration.



(a) Class D, break

(b) Class S, scratch

(c) Class P, pin damage

Figure 3: Illustration of each class of CID present in the dataset. The images were zoomed in to highlight the damages.

2. Create the first training set $\mathcal{T}_0$ as a random sample of $\mathcal{U}_0$ with size of $0.1 \times |\mathcal{U}_0|$.

3. At each iteration i, use $\mathcal{T}_i$ to train a detection model $\mathcal{M}_i$. Evaluate $\mathcal{M}_i$ over $\mathcal{V}$, calculating the recall by class R_C and the mAP.

4. If there are still images remaining in the data pool $\mathcal{U}_{i+1} = \mathcal{U}_1 - \mathcal{T}_i$, apply $A(k, \mathcal{U}_i) = \mathcal{S}_{i+1}$ to sample k new images, where $k = 0.05 \times |\mathcal{U}_1|$. I.e., at each iteration approximately 5% of the initial pool are sampled. If there are no images left, end the loop.

5. Add the sampled images $\mathcal{S}_{i+1}$ to the current training set to create the next iteration training set $\mathcal{T}_{i+1} = \mathcal{T}_i + \mathcal{S}_{i+1}$. Go to 3.

The random sampling loop follows an analogous process, except that in the step 3 the sampling of the images is done randomly, with uniform probability over all images. At the end of each loop we can compare the learning curve of both sampling methods, i.e., the evolution of $mAP@50$ as the training set increases size. Other relevant metric is the quantity of useful sampled images at each iteration. As disclosed before, there is a considerable proportion of images without any visible damage present in the dataset. We will compare the proportion of those images in the iterations of both loops. Finally, since our method aims to improve the performance of detection on the classes that are harder to detect, it will be analysed the evolution of recall by class in both loops.

Some considerations must be made regarding the conditions of the data pool $\mathcal{U}_1$. One assumption made by our method is that a large data pool with a high diversity of unlabeled images would be available, from where the active learning method would sample the most informative images. However, due the unavailability of human experts to iteratively annotate images and the limited number of annotated images to experiment with, this assumption does not completely hold on in these experiments.

As so, the results are limited by two aspects: i) the dataset $\mathcal{D}$ was sampled randomly from the ”true” large data pool $\mathcal{U}$ and annotated by experts, which means that it probably does not contains the most informative data of $\mathcal{U}$. I.e., our method is limited to sample only the most informative data from $\mathcal{D}$ which is random subset of the much larger set $\mathcal{U}$; ii) as the number of iterations increases, the train data used in the active learning loop and the train data used in the random sampling loop will become increasingly similar until become the same, as the images in $\mathcal{U}_i$ are sampled to exhaustion.

Given these considerations, we intend to analyze the difference between random sampling and the active learning method, with emphasis in the initial iterations, when the assumptions of our method better hold on.

5.2 Dataset

The dataset $\mathcal{D}$ contains a total of 2,844 RGB images of motherboards photographed by technicians from repair centers of a large hardware manufacturer company. The images have an average resolution of 2300 per 3200 pixels, with 96 dpi, and an average file size of 1.6MB. The images were analyzed by experts that annotated the location and the class of the CID in the images, when present. The annotations follows the standard COCO dataset format [22]. There are 3 classes of CID present in the annotations, that follows the company guideline: S, that includes abrasion and scratches; D, that includes cracks, break, and distortion; and P, pin damage. The description of this classes are summarized in Table 1 and Figure 3 illustrates a example of each class. We point out that there are other possible types of classes of CID, such as burn damage and liquid spillage. Such classes are not present in the data we had access to, but they can be addressed in future work.

As mentioned before, in the scenario at hand, a considerable amount of images does not contain any visible damage. Of the total images in dataset $\mathcal{D}$, 2,285 contains at least one annotation, corresponding to about 80% of the total images. We choose to maintain these images in our experiment to better reflect a real-life situation and also to evaluate the effectiveness of our method in sample useful images.

Table 1: Description of CID classes.

CID class	Description
S	Scratch, abrasion, nicks
P	Pin damage, including pin bent, missing, etc.
D	Cracks, breaks, distortion, smash and bump corners, etc

Table 2: Description of test set and data pool of the first iteration, with number of images and number of annotations of each class. The class P is a minority, with less than half the number of examples than the other two classes.

	$\mathcal{V}$	$\mathcal{U}_1$
N of images	543	2301
% of images with annotation	94	77
N of annotations of S class	200	840
N of annotations of P class	106	395
N of annotations of D class	249	861

After the random split of $\mathcal{D}$, the test set $\mathcal{V}$ contains 543 images, with 94% of these images having annotations. The pool $\mathcal{U}_1$ contains 2,301 images, from which 77% contains annotations. Table 2 summarize the information about the datasets, as well as the number of annotations of each class. It can be observed that the class P, corresponding to pin damage, is unbalanced in regard to the classes S (scratches) and D (breaks), which may cause to the class to be underrepresented during the learning of a detection model.

5.3 Detection model

As mentioned in Section 4, our method is model-agnostic, requiring only that each prediction of the model has an associated class probability distribution, as the confidence score will be used for the image scoring. In our experiments, we choose as the architecture of object detector a DNN with Swin-T as the backbone for feature extraction and the Mask-RCNN as the head, both discussed in Section 2. This is a state of the art architecture and achieved good results in preliminary experimentation. In the work of [6], which deals with the same task, the best results were achieved with the Swin architecture as the backbone. Figure 4 illustrates this architecture.

We use the default configuration of the smallest Swin-T with Mask-RCNN available in the MMDetection framework[1], pre-trained with the COCO dataset for 12 epochs, without multi-scale crop. This architecture has 29M parameters and consumes 7.6GB of RAM and 4.5 GFLOPs during training. For the training of the model is used the AdamW optimizer [37], with a initial learning rate of 0.0001 and a weight decay of 0.05. At each iteration of the experimental loops, models are trained by 36 epochs. That number of epochs was defined empirically, at it was observed that was necessary to the model to converge.

5.4 Results

As stated in subsection 5.1, both active learning and random sampling loops starts with a random sampled training set with 10% of the total images in data pool, which equals to 230 images. At each iteration on both loops, approximately 5% of the pool size is added to the training sets. This equals to a $k = 115$ images at each iteration. A total of 19 iterations are performed in both loops. The execution of all experiments took approximately 50 hours.

As previously mentioned, at the latter iterations the probability of the training sets of both loops have a large intersection increases, as the images are sampled until all images of $\mathcal{U}_1$ are exhausted. Therefore, the results converge in the latter iterations of both loops. As so, we focus the analysis of the results in the first 16 iterations, when 85% of the data of the pool $\mathcal{U}_1$ are sampled at each loop. We do so to better reflect a scenario where the assumptions of our method are satisfied.

5.4.1 Learning curve

Figure 5a illustrate the learning curves, considering the $mAP@50$ metric, calculated on the test set $\mathcal{V}$ through the iterations of both loops. It can been observed that even so the experimental setup does not completely hold on the assumptions of our method, it consistently surpass the performance of the random sampling, achieving better results at each iteration. Figure 5b shows the proportional difference between both loops at each iteration. Our method performed, on average, 21.6% better than random sampling, with a maximum of 65% at iteration 2, with 345 images (15% of the pool $\mathcal{U}_1$).

[1] https://github.com/open-mmlab/mmdetection/blob/v2.25.3/configs/swin/mask_rcnn_swin-t-p4-w7_fpn_1x_coco.py

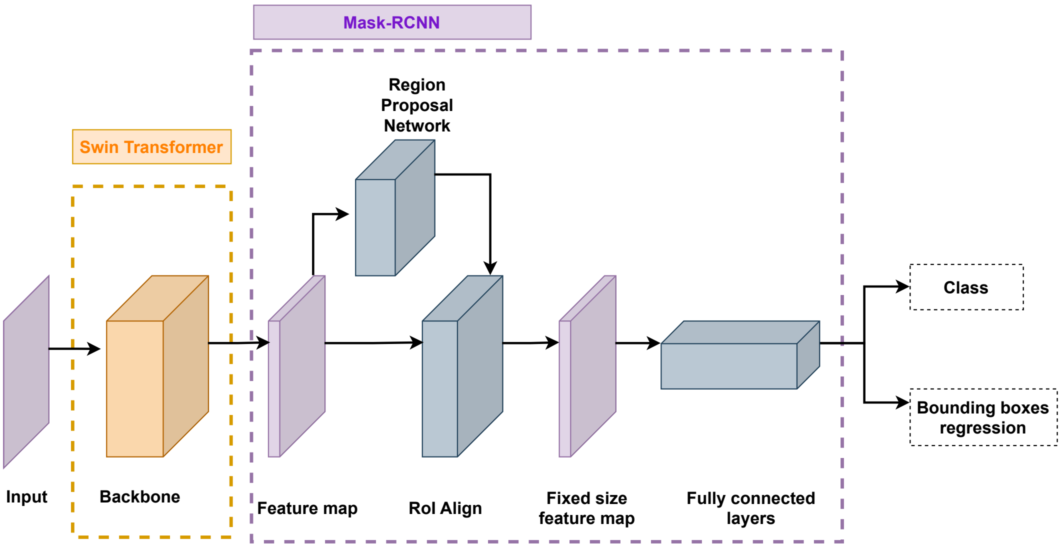
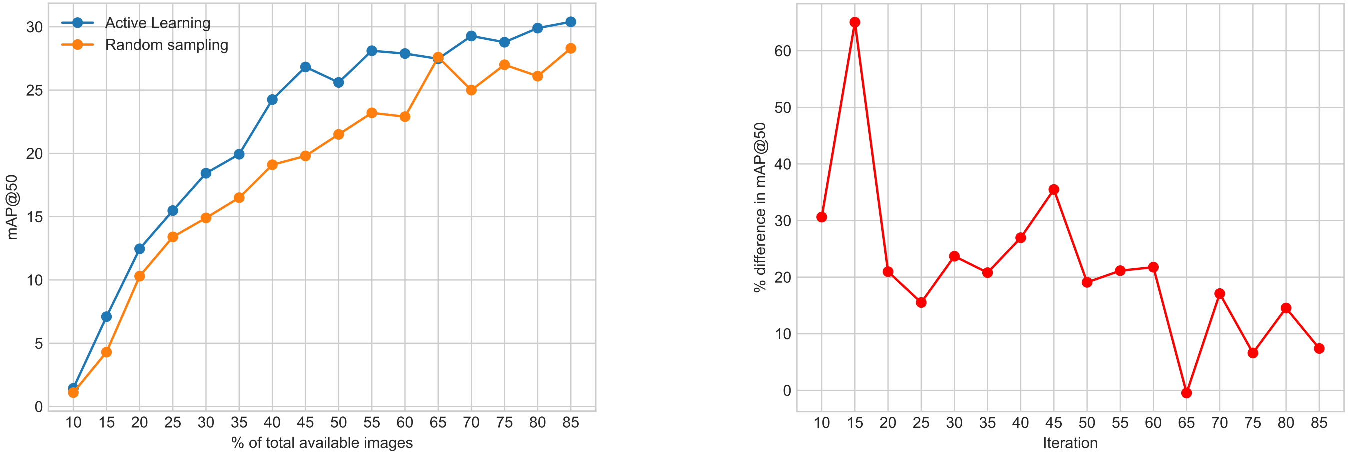


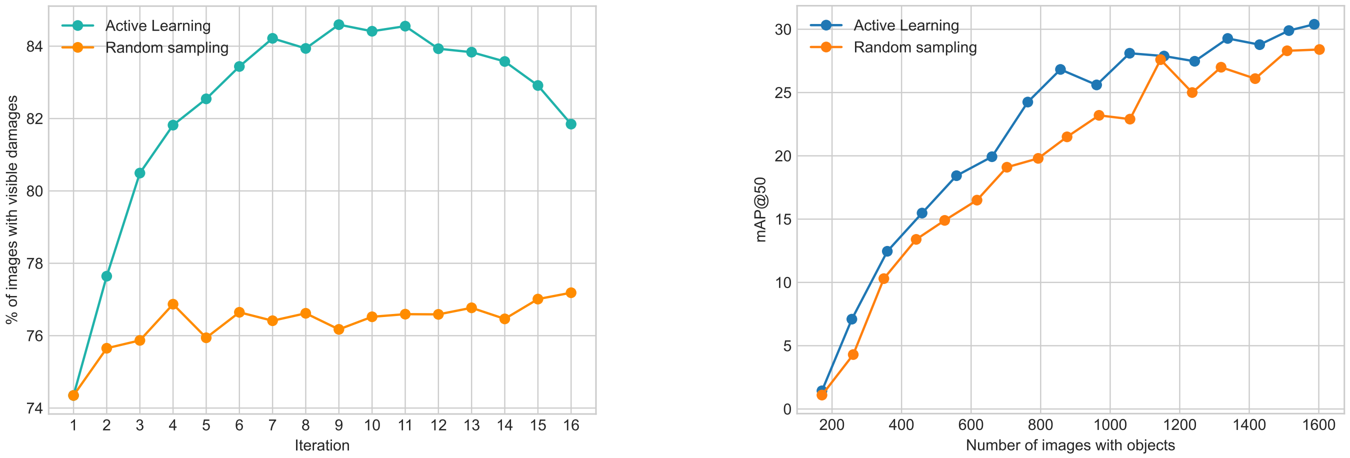
Figure 4: High level view of the architecture of the object detector adopted in the experiments.



(a) Learning curves of active learning and random sampling

(b) Proportional difference between learning curves

Figure 5: Learning curves by iteration. In iteration 1, 10% of the available data of $\mathcal{U}_1$ is used and 5% is added at each iteration.



(a) Proportion of total images with visible damage by iteration on each loop

(b) Learning curve considering the number of useful images

Figure 6: Impact of the sampling in proportion of useful images.

5.4.2 Proportion of useful images in samples

As pointed before, besides the $mAP@50$, we also wish to assess the performance of our method in sample useful images for training models, i.e., images with the presence of visible damage. Figure 6a shows the proportion of images with damage through the iterations. It can be seen that while the random sampling curve remains approximately constant, near the proportion of $\mathcal{U}$ of 77%, the active learning curve shows a rapid and sharp increase, achieving a peak proportion of 85% by iteration 9. This result is relevant for the reduction of annotation effort, as it means a reduction near 35% of the effort of a human annotator in discard non-useful images. After this iteration, the proportion starts to decrease, since the proportion of images with visible damage in the pool also decrease due its limited size. As the active learning samples more images with damages at the first iterations, it eventually depletes them of the data pool, leaving only images without damage to be sampled.

It could be argued that the difference in $mAP@50$ at each iteration between our method and random sampling (Figure 5a) is exclusively due the difference in proportion of usable images. To evaluate this, we compare the $mAP@50$ curve considering the number of usable images. This result is show in Figure 6b. It can be seen that even when has less useful images, our method still consistently surpass the random sampling. This means that our method not only has higher chance to sample useful images, but this images are more informative than the ones sampled randomly.

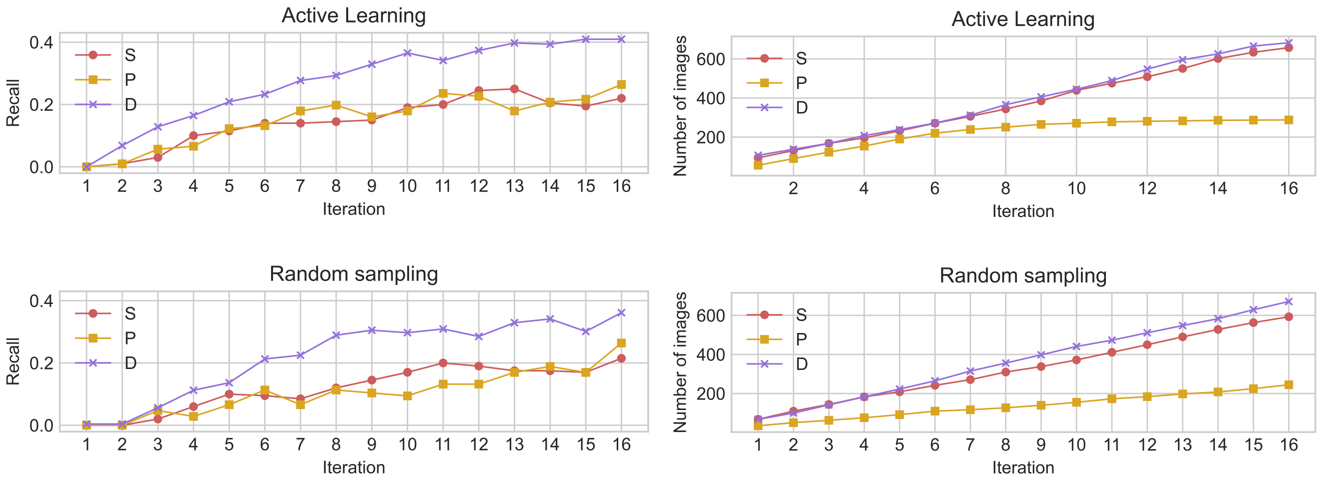
5.4.3 Recall analysis

Since our method aims to improve the capability of detection of the classes harder to detect, we also evaluate the evolution of recall by class in both loops. As showed before, the class P is minority in pool $\mathcal{U}_1$, which may result in worse detection performance. Figure 7a shows the evolution of recall by iteration. In general, the recall of all classes increases faster in the active learning loop, achieving better results in all iterations when compared with the random sampling loop. As stated before, the recall is a sensitive metric in this scenario and this improvement was intended by the use of the categorical uncertainty in Image Scoring stage. Particularly, the class P, has a minor gap in recall compared with other classes in the active learning loop than in the random sampling loop. While in active learning loop, the class P has a mean difference with the class S of 0.006, in the random sampling loop it is of 0.014.

The reduction of this recall gap is aimed by the image selection stage, when the number of images of each class is balanced in the sample. To better evaluate this effect, we present the quantity of images of each class by iteration in Figure 7b. In that case, the class of the image is assigned by the class that have the higher sum of area in the annotations of the image. It can be seen that in the random sampling loop the number of images of each class grows steady with the proportion of these images in the pool, with a gap of the number of images of P and the other classes. In the active learning loop, the number of images of the class P increases faster, reaching the quantity of other classes, as it was intended. The number of images of class P achieves a plateau around iteration 10, as all images in the pool $\mathcal{U}_1$ were sampled, which is a limitation of our experimental setup. It can be seen in Figure 7a that from this iteration, the recall of class P oscillates around the recall of the class S, when before this iteration it remained mostly superior.

5.5 Discussion

This results shows the effectiveness of our method in the presented conditions. However, as pointed out before, due the limited conditions of our experiment, i.e. the relatively small data pool $\mathcal{U}_1$, the full potential of this method is still uncleared. In a scenario where data is sampled of a large and diverse pool $\mathcal{U}$, where data is not sampled until exhaustion, the results could be



(a) Recall

(b) Images of each class

Figure 7: Evolution of recall and quantity of images of each class by iteration

even better. The difference in $mAP@50$ (Figures 5a and 5b) may take longer to converge and the proportion of useful images (Figure 6a) could reach a plateau instead of decrease. Nevertheless, the achieved results shows a promising evidence of the usefulness of this method for the task of CID detection in PCB. Furthermore, other tasks that share similar characteristics, (i.e. a large unlabeled data pool with non-informative images, a class accuracy imbalance and a focus on recall) could also benefit from this method.

One possible limitation of our method is that as it uses the uncertainty of predictions to sample images, it ignores images with false negatives. I.e., if the model fails to detect a damage in an image, making no prediction, this image will not be sampled, possibly resulting in the loss of relevant information. Further experiments must be made to assess the possible impact due this limitation.

6 Conclusion

We proposed and evaluated an model-agnostic active learning method for deep neural networks in the task of automatic detection of costumer induced damages in printed circuit boards, called Categorical Uncertainty Sampling Active Learning (CUSAL). Our method aims to solve some domain specific challenges, using uncertainty of predictions and the difficulty of detecting specific types of damage to sample the most informative images. The experimental results showed the capability of our method to improve the performance of a detector using with less data when compared to randomly sampled data. The results also showed that CUSAL can effectively sample more images with visible damage, reducing the effort of the human annotator in discard such images. Finally, the results showed the capacity of the method in improve the performance on the harder to detect types of damage, reducing class unbalance and the performance gap between classes.

As future work we intend to study alternatives to improve our method, in order to mitigate its deficiency in ignore images with false negatives, e.g. the combined use of diversity maximization [38] and the use of semi-supervised active learning [39]. We also intend to evaluate our method in a experimental setup where we sample directly from a large pool of images, labeling those images and comparing with random sampling. Finally, we intend to assess the possible generalization of our method for other domains and compare it with other active learning methods in the context of object detection.

7 Acknowledgement

This research was partially funded by Lenovo, as part of its R&D investment under Brazilian Informatics Law.

REFERENCES

[1] R. Heriansyah, S. A. R. Al-attas and M. M. Ahmad Zabidi. “Neural Network Paradigm for Classification of Defects on PCB”. *Jurnal Teknologi*, vol. 39, no. 1, pp. 87–104, Jan. 2012.

[2] V. A. Adibhatla, H.-C. Chih, C.-C. Hsu, J. Cheng, M. F. Abbod and J.-S. Shieh. “Defect Detection in Printed Circuit Boards Using You-Only-Look-Once Convolutional Neural Networks”. *Electronics*, vol. 9, no. 9, 2020.

[3] M. A. Alghassab. “Defect Detection in Printed Circuit Boards with Pre-Trained Feature Extraction Methodology with Convolution Neural Networks”. *Computers, Materials & Continua*, vol. 70, no. 1, pp. 637–652, 2022.

[4] V. A. Adibhatla, H.-C. Chih, C.-C. Hsu, J. Cheng, M. F. Abbod and J.-S. Shieh. "Applying deep learning to defect detection in printed circuit boards via a newest model of you-only-look-once". *Mathematical Biosciences and Engineering*, vol. 18, no. 4, pp. 4411, 2021.

[5] Y. Zhang, F. Xie, L. Huang, J. Shi, J. Yang and Z. Li. "A Lightweight One-Stage Defect Detection Network for Small Object Based on Dual Attention Mechanism and PAFPN". *Frontiers in Physics*, vol. 9, 2021.

[6] D. Alves, V. Farias, I. Chaves, R. Chao, J. P. Madeiro, J. P. Gomes and J. Machado. "Detecting Customer Induced Damages in Motherboards with Deep Neural Networks". In *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2022.

[7] A. Crispin and V. Rankov. "Automated inspection of PCB components using a genetic algorithm template-matching approach". *International Journal of Advanced Manufacturing Technology*, vol. 35, pp. 293–300, 12 2007.

[8] F. Raihan and W. Ce. "PCB defect detection USING OPENCV with image subtraction method". *2017 International Conference on Information Management and Technology (ICIMTech)*, pp. 204–209, 2017.

[9] J. Redmon, S. Divvala, R. Girshick and A. Farhadi. "You Only Look Once: Unified, Real-Time Object Detection", 2015.

[10] B. Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012.

[11] C.-A. Brust, C. Käding and J. Denzler. "Active learning for deep object detection". *arXiv preprint arXiv:1809.09875*, 2018.

[12] C. Yang, L. Huang and E. J. Crowley. "Plug and play active learning for object detection". *arXiv preprint arXiv:2211.11612*, 2022.

[13] J. Guo, Q. Wang, S. Su and Y. Li. "Informativeness-guided active learning for deep learning–based façade defects detection". *Computer-Aided Civil and Infrastructure Engineering*, 2023.

[14] X. Lv, F. Duan, J.-J. Jiang, X. Fu and L. Gan. "Deep active learning for surface defect detection". *Sensors*, vol. 20, no. 6, pp. 1650, 2020.

[15] Z. Ding, Y. Wu, X. Xiong and X. Zhu. "Chip Welding Short Circuit Defect Detection Based on Active Learning". In *2022 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, pp. 997–1000. IEEE, 2022.

[16] C. Szegedy, A. Toshev and D. Erhan. "Deep neural networks for object detection". *Advances in neural information processing systems*, vol. 26, 2013.

[17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie. "Feature pyramid networks for object detection". In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei. "Imagenet: A large-scale hierarchical image database". In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

[19] K. He, G. Gkioxari, P. DollÃ¡r and R. Girshick. "Mask R-CNN", 2018.

[20] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin and B. Guo. "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows", 2021.

[21] S. Ren, K. He, R. Girshick and J. Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", 2016.

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and C. L. Zitnick. "Microsoft coco: Common objects in context". In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

[23] D. Hakkani-Tür, G. Riccardi and A. Gorin. "Active learning for automatic speech recognition". In *2002 IEEE international conference on acoustics, speech, and signal processing*, volume 4, pp. IV–3904. IEEE, 2002.

[24] A. McCallum, K. Nigam *et al.*. "Employing EM and Pool-Based Active Learning for Text Classification." In *ICML*, volume 98, pp. 350–358. Citeseer, 1998.

[25] A. Holub, P. Perona and M. C. Burl. "Entropy-based active learning for object recognition". In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–8. IEEE, 2008.

[26] D. D. Lewis. "A sequential algorithm for training text classifiers: Corrigendum and additional data". In *Acm Sigir Forum*, volume 29, pp. 13–19. ACM New York, NY, USA, 1995.

[27] D. D. Lewis and J. Catlett. “Heterogeneous uncertainty sampling for supervised learning”. In *Machine learning proceedings 1994*, pp. 148–156. Elsevier, 1994.

[28] M.-F. Balcan, A. Broder and T. Zhang. “Margin based active learning”. In *International Conference on Computational Learning Theory*, pp. 35–50. Springer, 2007.

[29] K. Brinker. “Incorporating diversity in active learning with support vector machines”. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 59–66, 2003.

[30] T. T. A. Pham, D. K. T. Thoi, H. Choi and S. Park. “Defect Detection in Printed Circuit Boards Using Semi-Supervised Learning”. *Sensors*, vol. 23, no. 6, pp. 3246, 2023.

[31] F. He, S. Tang, S. Mehrkanoon, X. Huang and J. Yang. “A Real-time PCB Defect Detector Based on Supervised and Semi-supervised Learning.” In *ESANN*, pp. 527–532, 2020.

[32] E. Trajkova, J. M. Rožanec, P. Dam, B. Fortuna and D. Mladenić. “Active learning for automated visual inspection of manufactured products”. *arXiv preprint arXiv:2109.02469*, 2021.

[33] G. Van Rossum and F. L. Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[34] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai and S. Chintala. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In *Advances in Neural Information Processing Systems 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett, pp. 8024–8035. Curran Associates, Inc., 2019.

[35] NVIDIA, P. Vingelmann and F. H. Fitzek. “CUDA, release: 10.2.89”, 2020.

[36] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*. “MMDetection: Open mmlab detection toolbox and benchmark”. *arXiv preprint arXiv:1906.07155*, 2019.

[37] I. Loshchilov and F. Hutter. “Decoupled weight decay regularization”. *arXiv preprint arXiv:1711.05101*, 2017.

[38] Y. Yang, Z. Ma, F. Nie, X. Chang and A. G. Hauptmann. “Multi-class active learning by uncertainty sampling with diversity maximization”. *International Journal of Computer Vision*, vol. 113, pp. 113–127, 2015.

[39] S. Chen, Y. Yang and Y. Hua. “Semi-Supervised Active Learning for Object Detection”. *Electronics*, vol. 12, no. 2, pp. 375, 2023.