Journal of the Brazilian Computer Society, 2026, 29:1, doi: 10.5753/jbcs.2026.5323

Investigating the Impact of Image Quality Acquisition to Deep Object Detection Performance: a Case Study With PCB Damage Detection via Mobile Devices in the Wild

Lucas Cabral [**Systems and Databases Laboratory, Federal University of Ceará** | ***lucas.cabral@lsbd.ufc.br***]
João Pedro Santiago [**Systems and Databases Laboratory, Federal University of Ceará** | ***pedro.santiago@lsbd.ufc.br***]
Lucas Sena [**Systems and Databases Laboratory, Federal University of Ceará** | ***lucas.sena@lsbd.ufc.br***]
Joaquim Bento Cavalcante Neto [**Systems and Databases Laboratory, Federal University of Ceará** | ***joaquim.bento@lsbd.ufc.br***]
Yuri Lenon [**Systems and Databases Laboratory, Federal University of Ceará** | ***yuri.lenon@lsbd.ufc.br***]
Javam Machado [**Systems and Databases Laboratory, Federal University of Ceará** | ***javam.machado@lsbd.ufc.br***]

Computer Science Department, Federal University of Ceará (UFC), Campus do Pici - Bloco 952 - CEP 60440-900 - Fortaleza, CE - Brazil

Received: 13 December 2024 • **Accepted:** 05 September 2025 • **Published:** 16 YYYY

Abstract Mobile devices have revolutionized image acquisition, enabling diverse communication and image-processing applications. One promising application is automated damage detection in printed circuit boards (PCBs), crucial for quality control in electronics manufacturing. However, unlike controlled environments, mobile device image acquisition introduces challenges such as non-uniform lighting, background interference, and varying camera resolution, which can affect the accuracy of deep learning models. This paper presents a case study investigating the impact of domain-specific, no-reference image quality metrics on the performance of deep-learning object detection models for PCB damage detection. We evaluate nine metrics, including five novel contributions, using a semantic segmentation approach to measure foreground and background quality. Our study assesses how these metrics influence the performance of deep neural network architectures, determining optimal thresholds that separate high and low-quality images. Experiments are conducted on real-life images captured with smartphones in industrial settings. Our findings indicate that filtering poor-quality images based on these metrics could significantly improve detection performance, offering practical benefits for mobile damage detection applications.

Keywords: Image Quality Assessment, Mobile Image Acquisition, Deep Object Detection, PCB Damage Detection

1 Introduction

1.1 Motivation

Mobile devices such as smartphones and tablets are widespread and fully integrated into the daily lives of their users, enabling applications that range from communication and multimedia content to daily planning, gaming, and more. These devices typically feature high computational processing capabilities, high-resolution cameras, and internet connectivity. This combination of factors has led to deep learning-powered image processing applications gaining attention from both academia and industry [Morikawa *et al*., 2021].

Through such applications, users can capture photos or videos with their devices and send them to remote servers that provide Machine Learning as a Service (MLaaS), or even process them locally. Deep learning models can extract knowledge from images using methods such as image classification, object detection, or semantic segmentation [Voulodimos *et al*., 2018]. The processed results assist users by providing analytical insights, automating tasks, and reducing costs.

Examples of areas of application for mobile computer vision include agriculture [Picon *et al*., 2019; Nayak *et al*., 2023; Fu *et al*., 2018], healthcare [Nejati *et al*., 2016; Dallet *et al*., 2014], accessibility [Jin *et al*., 2016; Shukurov, 2024], infrastructure [Coca *et al*., 2021; Doğan and Ergen, 2022], augmented reality [Liu *et al*., 2019], and manufacturing [Lei *et al*., 2018]. Therefore, mobile computer vision applications combine the high accuracy of deep learning models with the flexibility, convenience, scalability, and lower costs provided by mobile devices, as they can be used anywhere and do not require expensive hardware or infrastructure for image acquisition [Oike, 2022].

In this context, an application that has high economic importance and has shown good results is automated damage detection in printed circuit boards [Du *et al*., 2023; Santiago *et al*., 2024; Liu *et al*., 2023]. A printed circuit board (PCB) is a board used in electronic devices to mechanically support and electrically connect various electronic components, mak-

ing it crucial to their internal structure. Given the significant role of PCBs in electronic devices, assessing this structure during the product quality assurance and control process is extremely important.

PCB diagnosis is predominantly performed manually by specialized technicians [Adibhatla *et al*., 2020]. This task is time-consuming and susceptible to errors, as defects can be challenging to identify. Consequently, there has been interest in automating PCB inspection, with some researchers focusing on developing automatic defect detection methods through computer vision techniques such as object detection [Alves *et al*., 2022]. A technician can take a photo of a PCB and obtain the position, delimited by bounding boxes, and classification of the damage categories present in the image, such as burns, cracks, scratches, and mold. Moreover, conducting this inspection through a mobile device provides the advantages of flexibility, scalability, and low cost. This assists both technicians and non-technicians in performing inspections in various scenarios with agility, from factory floors to field inspections, such as clients' homes, workplaces, and more.

Nevertheless, despite these many advantages, mobile computer vision applications face challenges in image acquisition that may negatively impact the performance of deep learning models [Henning *et al*., 2015]. Unlike controlled and standardized image acquisition settings, image acquisition on mobile devices highly depends on the user's skill in managing focus, position, and framing of the photo. Additionally, real-life acquisition conditions can also interfere due to the unconstrained environment, such as non-uniform lighting, background interference, and camera resolution. In particular, the task of PCB damage detection is subject to specific issues caused by improper photo capture methods, which impact the quality of the analysis by deep learning models [Cabral *et al*., 2023].

The distance of the camera from the PCB is an important factor. PCBs are composed of many small components, which means that many damages are also small. The further away the photo is taken from the PCB, the smaller the objects of interest will appear, leading to a loss of visual fine-grained details and information. The same issue can arise with low-resolution photos. Additionally, photo glare, caused by reflections from shiny surfaces, and blur, due to camera instability, motion, or defocus, are also common distortions. Finally, the background of the image—regions that are not the PCB—may contain textures or objects that visually resemble damage, leading the model to falsely detect damage on surfaces outside of the PCB. Understanding the impact of each of these factors on detection performance may guide the improvement of mobile applications through the development of assisted image-capturing tools and image preprocessing steps.

Many of the aforementioned distortions can be measured or estimated using Image Quality Assessment (IQA) metrics [Zhai and Min, 2020]. Over the years, numerous IQA metrics have been proposed to evaluate the perceptual quality of images. These metrics aim to produce an objective score that aligns closely with how the human visual system perceives image quality. However, it is still not fully understood how these metrics correlate with the performance of deep learning models.

Nonetheless, some gaps still exist in the literature. First, all the aforementioned works experimented with synthetic datasets, where visual distortions are artificially added to images in an attempt to reproduce natural image distortions. As such, their findings may not hold for natural image distortions encountered in real-world scenarios. Moreover, having both the original and distorted images allows for the computation of full-reference IQA metrics like SSIM, which is not possible with photos captured in the wild. Second, no-reference metrics such as BRISQUE are optimized for images of natural scenes, which come from a distribution that is very different from that of PCB images. Furthermore, these metrics are not interpretable and cannot be used to assist users of mobile applications in image capture. Finally, these studies do not address specific characteristics relevant to the problem of damage detection in PCBs, such as the distance of the board and background interference.

1.2 Proposed Approach

In this paper, we propose to investigate the impact of image acquisition on deep object detection, based on a case study investigating the impact of a set of domain-specific, no-reference image quality metrics on the performance of deep learning object detection models for damage detection in PCBs. We examine 9 metrics, of which 5 are introduced in this work. Specifically, we develop a subset of metrics obtained using a semantic segmentation approach, which allows us to measure both the background and the foreground. To assess the importance of each metric, we evaluate the difference in performance between images with high and low levels of quality issues—i.e., poor quality and good quality images according to the proposed metrics. We conduct our experiments on real-life images captured with smartphones in hardware repair centers at a large manufacturing company. To the best of our knowledge, this is the first such analysis.

1.3 Contributions

Our findings reveal that there are optimal thresholds for the evaluated metrics. Images with metrics below the optimal quality threshold showed noticeably lower performance compared to those above it. Approximately 30% of the test images were considered below the quality threshold. A practical implication of this is to discard such poor-quality images in a mobile damage detection app before they are sent for inference. This could provide feedback to users to capture better-quality photos, thereby improving detection performance.

Our key contributions include:

1. We propose a set of domain-specific, numeric, no-reference, and interpretable metrics to evaluate different aspects of image quality for damage detection. We investigate a total of 9 metrics, 5 of which are novel. These metrics can be used to provide feedback to users of a mobile application to capture better quality photos.
2. We evaluate the impact of each proposed metric on the performance of two well-known damage detection architectures, YOLOv6 and Mask-RCNN. We determine the optimal thresholds for these measures, i.e., the thresholds

that separate images with lower performance. Our evaluation is based on a real-life industry dataset of images captured with smartphones.

1.4 Outline

The remainder of this paper is organized as follows. Section 2 provides the necessary background information relevant to the study, including foundational concepts and terminologies. In Section 3, we review related works, highlighting previous research and existing approaches in the field. Section 4 details the quality issues and metrics in PCB Damage Detection. Section 5 details the methodology employed in our study, outlining the research design, data collection, and analysis procedures. The results of our study are presented and discussed in Section 6, where we analyze the findings and their implications. Finally, Section 7 concludes the paper with a summary of the key contributions, limitations of the study, and suggestions for future research directions.

2 Background

2.1 PCB Damage Detection

Damage detection in printed circuit boards (PCBs) is crucial for ensuring the reliability and longevity of electronic devices. Damage to PCBs can originate from various sources, including manufacturing defects, handling errors, and Customer Induced Damage (CID). Manufacturing defects can arise from improper soldering or component placement, such as missing fan screws, loose screws, incorrect screw types, and missing fan cables [Gong *et al*., 2023]. Handling errors, including electrostatic discharge or mechanical stress during assembly and transportation, can also compromise PCB integrity. CID encompasses damage caused by unauthorized individuals, including end-users, and can manifest as cracks, scratches, bent pins, liquid spills, mold, and burns [Alves *et al*., 2022]. Since CIDs result from different causes, their visual characteristics often differ from those of manufacturing defects. This work specifically focuses on CID detection.

The economic implications of PCB damage are significant. Early identification and diagnosis can prevent costly warranty claims and reduce the need for extensive repairs or replacements. Tools that aid technicians in identifying PCB damage, such as automated optical inspection systems, X-ray inspection, and thermal imaging, are essential for maintaining product quality and minimizing financial losses [Zhou *et al*., 2023]. These advanced diagnostic tools provide high-quality, high-speed, and stable image acquisition during the PCB inspection process, ensuring robust quality control and substantial long-term cost savings. However, such image acquisition equipment is expensive and not readily available for warranty programs, repair centers, or field inspections. PCB damage detection through mobile applications offers a cost-effective and flexible solution that can assist technicians in diagnosing PCB issues, helping manufacturing companies save money.

2.2 Computer Vision

Computer vision is a field of artificial intelligence that enables computers to interpret and make decisions based on visual data from the world [Szeliski, 2022]. This technology is crucial for a variety of applications, ranging from automated image analysis to advanced robotic vision systems. The advent of deep learning has significantly advanced the capabilities of computer vision, enabling more accurate and sophisticated analyses. Convolutional Neural Networks (CNNs) have been particularly instrumental in this progress due to their ability to capture spatial hierarchies in images through convolution operations [LeCun *et al*., 1998].

The main tasks in computer vision include image classification, semantic segmentation, and object detection. Image classification involves assigning a label to an image from a predefined set of categories. Semantic segmentation aims to classify each pixel in an image into a category, thereby providing detailed information about the image's regions. Object detection goes a step further by not only identifying objects within an image but also localizing them with bounding boxes. This task is critical for applications such as autonomous driving, where identifying and localizing objects like pedestrians and vehicles is essential for safe navigation [Grigorescu *et al*., 2020]. Other applications include surveillance systems [Zou *et al*., 2019], medical imaging for detecting tumors [Litjens *et al*., 2017], retail for inventory management, customer behavior analysis [Liu *et al*., 2020], and PCB damage detection [Alves *et al*., 2022; Santiago *et al*., 2024].

Object detection is more complex than image classification, as it involves not only recognizing objects but also determining their categories and positions within the image. Deep learning has transformed object detection through the use of CNNs, which can learn hierarchical features directly from the data. In object detection, bounding boxes are used to specify the location of objects. These boxes are typically defined by their coordinates in the image and are refined during training to achieve higher accuracy [Ren *et al*., 2015]. To train deep object detectors, images annotated with the coordinates of the bounding boxes for the objects of interest are required. A trained object detector processes an image and outputs the coordinates of the detected objects along with a probability distribution over the object classes associated with each bounding box. The highest probability of a predicted bounding box is referred to as the confidence score. Typically, only predictions with confidence scores above a predetermined threshold are considered.

Deep learning object detection methods generally fall into two primary categories: two-stage detectors and one-stage detectors. Two-stage detectors operate by first generating a set of region proposals that are likely to contain objects. In the second stage, these proposals are classified and refined. The Mask R-CNN architecture extends this approach by adding a branch for predicting segmentation masks, thus providing pixel-level object segmentation in addition to bounding box detection [He *et al*., 2017]. One-stage detectors, such as YOLO (You Only Look Once) [Redmon *et al*., 2016], bypass the region proposal step and predict bounding boxes and class probabilities directly from the full image in a single pass. This approach is significantly faster but can sometimes be

less accurate compared to two-stage detectors. YOLOv6, improves upon previous versions with an optimized network architecture, achieving state-of-the-art performance in real-time object detection scenarios [Li *et al*., 2022].

2.2.1 Metrics

The performance of object detectors is evaluated using several metrics.

Intersection over Union (IoU) Intersection over Union (IoU) is a measure of the overlap between the predicted bounding box B_p and the ground truth bounding box B_{gt}. It is defined as:

$$\text{IoU} = \frac{|B_p \cap B_{gt}|}{|B_p \cup B_{gt}|}, \tag{1}$$

where $|B_p \cap B_{gt}|$ is the area of overlap between the predicted and ground truth bounding boxes, and $|B_p \cup B_{gt}|$ is the area of their union.

Precision and Recall Precision and recall are fundamental metrics for evaluating the performance of an object detector. Precision measures the accuracy of the detected objects. It is defined as the ratio of true positive (TP) detections (correctly identified objects) to the total number of detected objects (true positives plus false positives, FP):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2}$$

High precision means that most of the detected objects are correct, with few false positive detections (incorrectly identified objects).

Recall in object detection measures the ability of the model to find all relevant objects in an image. It is defined as the ratio of TP detections (correctly identified objects) to the total number of actual objects present in the image (true positives plus false negatives, FN).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{3}$$

High recall means that the model successfully detects most of the actual objects in the image, with few false negative detections (missed objects).

Mean Average Precision (mAP) Mean Average Precision (mAP) is a comprehensive metric that summarizes the precision-recall curve for a given IoU threshold. To compute mAP, we follow these steps:

1. Compute the precision and recall at various thresholds of confidence scores.
2. Plot the precision-recall curve.
3. Compute the Average Precision (AP) for each class, which is the area under the precision-recall curve.
   The AP for a single class is calculated as:

$$\text{AP} = \sum_n (R_n - R_{n-1}) P_n, \tag{4}$$

   where R_n and R_{n-1} are the recall values at the n-th and $(n-1)$-th thresholds, respectively, and P_n is the precision at the n-th threshold.
4. Compute the mAP by averaging the APs over all classes:

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i, \tag{5}$$

   where N is the number of classes, and AP_i is the Average Precision for the i-th class.

The IoU threshold plays a critical role in determining true positives, which significantly affects precision, recall, and subsequently, the mAP. Typically, an IoU threshold of 0.5 is used, but more rigorous evaluations might use higher thresholds like 0.75 to measure the accuracy of object localization more strictly [Everingham *et al*., 2010]. In this paper, we refer to mAP with a threshold of 0.5.

2.3 Image Quality Assessment

Image Quality Assessment (IQA) is a field in image processing and computer vision that focuses on evaluating the perceptual quality of images. IQA methods are essential for various applications, including image compression, restoration, and enhancement, as well as for the development of new imaging technologies [Zhai *et al*., 2021]. These methods can be broadly categorized into Full-Reference Image Quality Assessment (FR-IQA) and No-Reference Image Quality Assessment (NR-IQA).

FR-IQA methods rely on the availability of a reference image, which is assumed to be of perfect quality. Commonly used metrics in FR-IQA include Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Feature Similarity Index (FSIM) [Wang *et al*., 2004; Zhang *et al*., 2011]. In contrast, NR-IQA methods do not require a reference image, making them more versatile and applicable in real-world scenarios where the reference image is unavailable. NR-IQA methods estimate image quality based solely on the distorted image. These methods leverage statistical models, machine learning, and deep learning techniques to predict perceptual quality. Recent advances in NR-IQA have seen the development of sophisticated models that can effectively predict image quality by learning from large datasets of human-rated images [Bosse *et al*., 2017; Mittal *et al*., 2012; Talebi and Milanfar, 2018; Zhu *et al*., 2020; Zhai *et al*., 2021]. NR-IQA has been applied to evaluate images used in deep learning systems, aiming to predict performance based on image quality [Kong *et al*., 2019; Bergstrom and Messinger, 2023]. These techniques can be used to select high-quality images for model training or to ensure that only good-quality images are fed to the models during inference.

However, image quality as judged by humans may differ from the quality determined by the performance of computer vision algorithms [Bergstrom and Messinger, 2023]. General IQA metrics are often difficult to interpret and do not clearly explain which visual distortions are present in images and how they impact model performance. Furthermore, the quality metrics that affect the performance of computer vision algorithms can be domain-specific.

2.4 Image Processing for Image Quality

Image processing encompasses a variety of techniques aimed at enhancing and analyzing visual data to extract meaningful information or improve image quality. Key tasks in image processing include edge detection, image segmentation, noise reduction, and image enhancement [Marengoni and Stringhini, 2009]. Each of these tasks plays a crucial role in addressing specific aspects of image quality and ensuring accurate interpretation of visual information.

Edge Detection is a fundamental task in image processing that involves identifying the boundaries or transitions in intensity within an image. This technique is essential for recognizing shapes, objects, and structures [Jing *et al*., 2022]. The Canny Edge Detector is a widely used method for this purpose, known for its robustness in detecting a broad range of edges while minimizing noise and false detections [Rong *et al*., 2014]. The process involves smoothing the image with a Gaussian filter [Basu, 2002], computing the gradient magnitude and direction, applying non-maximum suppression to thin the edges, and using hysteresis thresholding to confirm the presence of strong edges. Effective edge detection is crucial for assessing image sharpness and clarity, as blurred or indistinct edges can indicate problems such as improper focus or excessive noise.

Image Segmentation involves partitioning an image into distinct regions or segments to simplify or alter its representation. This task plays a critical role in evaluating image quality, particularly in detecting and analyzing defects or distortions. Segmentation techniques help isolate different regions of an image, making it easier to analyze and assess specific areas for quality issues [Kheradmandi and Mehranfar, 2022]. Techniques such as thresholding, clustering (e.g., k-means), and advanced methods like active contours are used for segmentation [Al-Rahlawee and Rahebi, 2021]. Thresholding divides the image based on pixel intensity, while clustering methods group pixels based on feature similarities. Active contours adjust boundaries dynamically to fit object shapes. Accurate segmentation aids in identifying and quantifying defects, analyzing structural features, and improving the overall quality of image analysis.

Noise Reduction is another critical task in image processing that aims to eliminate unwanted variations or distortions caused by noise. Techniques such as Gaussian filtering, median filtering, and wavelet denoising are used to smooth the image and enhance its quality. By reducing noise, these methods improve the clarity and accuracy of subsequent image processing tasks, such as edge detection and segmentation.

Together, these image processing tasks—edge detection, image segmentation, and noise reduction—form a comprehensive approach to analyzing and improving image quality. They address different aspects of image integrity and functionality, facilitating more accurate and effective interpretation of visual data.

3 Related Works

3.1 PCB Damage Detection

The problem of damage detection in PCBs through object detection has been extensively researched in academia [Zhou *et al*., 2023]. Most of the work focuses on manufacturing defect detection, which is part of quality control in industries. The studies by [Adibhatla *et al*., 2020; He *et al*., 2020; Tarvainen and Valpola, 2017] investigate the application of models from the YOLO family for real-time manufacturing defect detection. Approaches using two-stage detectors for manufacturing defect detection can be found in Hu and Wang [2020]; Li *et al*. [2020]. These works experimented with images captured in standardized settings, in controlled environments, and using high-quality cameras or automatic optical inspection machines during manufacturing quality control. Therefore, the images used do not contain quality issues resulting from mobile device acquisition, making it a less challenging task. The best results achieved an mAP as high as 0.985.

In contrast to manufacturing defects, Customer Induced Damage (CID) occurs after the product is purchased, and its detection is typically part of a warranty program inspection. The work of [Alves *et al*., 2022] investigates automatic CID detection using two-stage object detectors with a dataset of images collected in repair centers using smartphone cameras, as part of warranty program registration for legal support. The images were collected under diverse conditions, including varying illumination, PCB position and orientation, background, capture distance, inclination, and focus. The authors experimented with network architectures such as Mask R-CNN [He *et al*., 2017], Soft Teacher [Xu *et al*., 2021], and Swin Transformer [Liu *et al*., 2021] as backbones for the Mask R-CNN network for CID applications, achieving a mAP of 0.802.

The work of Santiago *et al*. [2024] uses a dataset collected under similar conditions and evaluates real-time CID detection using detectors from the YOLO family. Their best result, with a mAP of 0.331, was achieved with YOLOv6 [Li *et al*., 2022]. The work of Cabral *et al*. [2023] also deals with CID detection but focuses on an Active Learning method to select the most informative images from a pool of unlabeled images for annotation. It was demonstrated that the proposed method has a higher chance of selecting better quality images compared to random selection. However, none of the works mentioned directly investigated the relationship between damage detection performance and image quality.

3.2 Image Quality in Computer Vision Tasks

Some research has initiated the investigation of the correlation between image quality and computer vision tasks [Gummadi *et al*., 2023; Siddiqui *et al*., 2022; Kong *et al*., 2019; Hao *et al*., 2023]. The work of [Gummadi *et al*., 2023] finds a significant relationship between model performance and general-use metrics, such as the Structural Similarity Index Measure (SSIM), which compares a distortion-free reference image with the distorted image being evaluated [Wang *et al*., 2004], and the Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE), a no-reference image quality score for natural

scenes [Mittal *et al*., 2012]. The authors propose a correlation analysis of 17 traditional IQA metrics (both no-reference and full-reference) with the performance of object detectors in the context of autonomous driving. They conducted experiments on a modified version of the public benchmark KITTI MoSeg [Geiger *et al*., 2012] and applied different levels of compression to simulate variations in image quality. The results showed that several FR-IQA metrics had a strong positive correlation (exceeding correlation scores of 0.7) with average precision. In contrast, our work uses real-life images without artificially adding distortions or lowering the quality through image processing, meaning the image distortions are inherent to the acquisition process.

Other works, however, focus on developing novel metrics tailored for specific failure modes rather than correlating existing metrics with task performance. For instance, [Nam *et al*., 2025] proposes a No-Reference Image Quality Assessment (NR-IQA) model called BREMOLA, designed to monitor the health of camera sensors in autonomous driving systems by specifically measuring image blur, which they identify as the primary sign of sensor aging or deterioration. The goal is not to predict object detection performance but to provide a direct, real-time indicator of camera failure. Their method uses a Fourier transform to create a ”moving spectrum” that quantifies image sharpness, which is then normalized for environmental variations (like nighttime lighting or number of objects) using a Laplacian filter. To validate their model, they simulate different levels of camera degradation by applying an average filter with varying kernel sizes (from 1x1 to 20x20) to real-world driving images captured in diverse environments. The evaluation of BREMOLA focuses on its ability to robustly distinguish between the original, high-quality images and their artificially blurred counterparts, establishing a ”failure threshold” to determine when a sensor’s quality has degraded unacceptably. This approach centers on creating a standalone quality score for system health monitoring, distinct from assessing the performance of subsequent computer vision tasks.

Siddiqui *et al*. [2022] introduces benchmark datasets, RVL-CDIP-D and Tobacco3482-D, to assess the robustness of document image classifiers against real-world distortions. By augmenting well-known document image classification datasets with 21 different types of distortions (including blur, noise, pixelation, compression, geometric distortion, and surface distortion), the paper provides insights into the limitations of high-accuracy models in handling specific distortions, showcasing declines in classification accuracy in some cases.

Kong *et al*. [2019] proposes a regression model based on a bagging ensemble of trees to predict object detection performance on an image, aiming to evaluate image quality from the perspective of automatic analysis algorithms. The model utilizes easily extracted local and global features to achieve more accurate predictions of image quality for object detection compared to traditional IQA measures like PSNR and SSIM. Experiments were conducted on the Multiple Object Tracking (MOT) dataset and the Duke Multi-Target Multi-Camera Tracking (DM) dataset, with four types of artificial visual distortions: Gaussian blur, motion blur, imaging noise, and reduced spatial resolution. This work is related to ours by tailoring metrics to object detection performance, but the data used includes artificially added distortions.

Bergstrom and Messinger [2023] focused on evaluating the impact of image quality factors (resolution, blur, noise) on object detection performance using deep learning models. The study involved training models on high-quality images and fine-tuning them on lower-quality images to assess performance variations. It mapped image quality variables to the General Image Quality Equation (GIQE) terms and analyzed the suitability of GIQE for modeling object detector performance under significant image distortions.

Hao *et al*. [2023] evaluates the impact of image resolution variation and object distance on general object detection and proposes an architecture robust to varying resolutions, named RA-YOLO. The study demonstrated that closer objects are easier to detect than those farther away, emphasizing the need to consider object distance in optimizing object detection systems.

3.3 Discussion of Related Work

Table 1 positions our paper among related works. Summing up, our approach diverges from the works found in literature in the following points:

1. We use real-life images with natural quality issues that arise from mobile device acquisition, instead of artificially adding image distortion.
2. We evaluate domain-specific quality metrics for PCB damage detection. The metrics were chosen empirically from previous experiments. Most are well-known metrics of image processing, but some are proposed in our work. To the best of our knowledge, our work is the first to evaluate this set of metrics in relation with object detection performance.

4 Quality Issues and Metrics in PCB Damage Detection

Image quality in the context of PCB damage detection with mobile devices can be affected by different types of distortion or inappropriate usage that occur during image acquisition. We propose a set of metrics that give an approximate measure of the degree of the quality issues that are believed to impact PCB damage detection performance. Figures 1 and 2 illustrate how the selected quality issues induce detection errors. Table 2 summarizes the metrics.

4.1 Distance From Camera

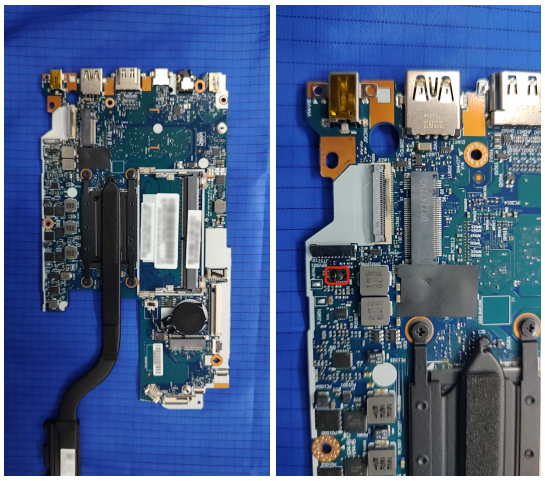
The relation between distance from camera to objects and tiny objects with performance decay of detection was demonstrated in several works [Hao *et al*., 2023; Noh *et al*., 2019; Bai *et al*., 2018]. PCBs have a lot of small components that can be damaged by mechanical causes, such as scratches, cracks, and pin bents, as due other factors such as burns due to overheating. These damages can be very small and hard to see from certain distances due the loss of resolution, as they are represented by just a few pixels. In previous experiments

Table 1. Comparison of related works

Work	**Task**	**Quality Issues**	**Quality Metrics**	**Analysis**
[Gummadi *et al*., 2023]	Object detection for autonomous driving	Artificially added compression	Traditional FR-IQA and NR-IQA	Correlation of quality metrics with detection performance
[Siddiqui *et al*., 2022]	Document classification	Artificially added 21 types of distortions such as blur, noise, pixelation, and compression	Degree of added distortions	Performance analysis in presence of distortions
[Kong *et al*., 2019]	Object tracking	Artificially added Gaussian blur, motion blur, imaging noise, and reduced spatial resolution	Tailored domain-specific metrics	Prediction of detection performance based on quality
[Bergstrom and Messinger, 2023]	General object detection	Artificially added blur, noise, and reduced spatial resolution	Degree of added distortions and General Image Quality Equation (GIQE)	Performance analysis in presence of distortions
[Hao *et al*., 2023]	General object detection	Distance from object and artificially added reduced spatial resolution	Degree of added distortions	Performance analysis in presence of distortions
[Nam *et al*., 2025]	Image quality assessment for autonomous driving	Artificially added blur	Proposed NR-IQA (BREMOLA) and traditional FR/NR-IQA	Robustness analysis of the proposed metric for failure detection
Ours	PCB damage detection	Naturally borne from mobile image acquisition, including blur, glare, perspective distortion, distance from object, and background noise	Tailored domain-specific metrics	Performance analysis between poor quality images and good quality images according to empirical measures

Table 2. Quality issues and metrics investigated in this work. The Reference column shows how the metric is addressed in this work. The Area Ratio metric is the only one that is higher the better.

Quality Issue	Metric	Reference	Direction of Improvement	Contribution of This Work
Distance From The Camera	Area Ratio	Area Ratio	Maximize	Ours
Noise	Fast Noise Estimation	Noise	Minimize	Others
	Signal-to-Noise Ratio	SNR	Minimize	Others
Background Interference	Background Variance	Var BG	Minimize	Ours
	Variance of Edges in Background	Var Canny BG	Minimize	Ours
	Background Noise	Noise BG	Minimize	Adapted from others
	Background SNR	SNR BG	Minimize	Adapted from others
Glare	Photometric Features	Glare	Minimize	Others
Blur	Laplacian Variance	Blur	Minimize	Others



(a) Area ratio: 40%. The superior left burn was not detected.

(b) Area ratio: 75%. The burn damage was successfully detected.

Figure 1. Damage detection in the same PCB at different distances. Information in labels was blurred due to confidentiality issues.

with a fixed image resolution, we found out that taking pictures of PCBs in distances far from 20 cm severely impacts the detection accuracy, reducing by 60% the recall. Figure 1 illustrates an exemplary case, where damage is not detected when the photograph is taken far from 20cm, where, the area of the PCB correspond to nearly 40% of the image. When a photograph was taken closely in the same PCB, where it corresponds to 75% of the image, the damage was correctly detected. Thus, it is essential that users take photographs with a appropriated framing to reach good quality images.

Metric: Area Ratio To estimate the distance of the PCB and, consequently, the resolution of objects, we devise a method based in semantic segmentation. The main idea is to distinguish the regions of the image that are the PCB, i.e., the foreground, from the regions that are not, i.e. the background, and then measure the foreground. We trained a U-Net++ [Zhou *et al.*, 2018] model using 168 images of manually segmented PCBs. This training set for the UNet++ model differs from the evaluation dataset used in our study. The model achieved a recall of 97%. We evaluate the adequacy of photograph distance by measuring the area of the segmented PCB divided by the total area of the image. The smaller this ratio, the more background area there is in the image, i.e. there are less useful information, as damages can only occur on PCBs surfaces. When this measure is near 1, it means that most or all of the image is composed of the PCB surface, which ensures the damages are at an acceptable resolution and there is less background interference. We highlight that for evaluating the impact of objects sizes, we could directly measure it from the labels, but we chose to use only metrics that can be computed with unlabeled images, so it could be applied during inference time as a feedback device to assist users.

4.2 Noise

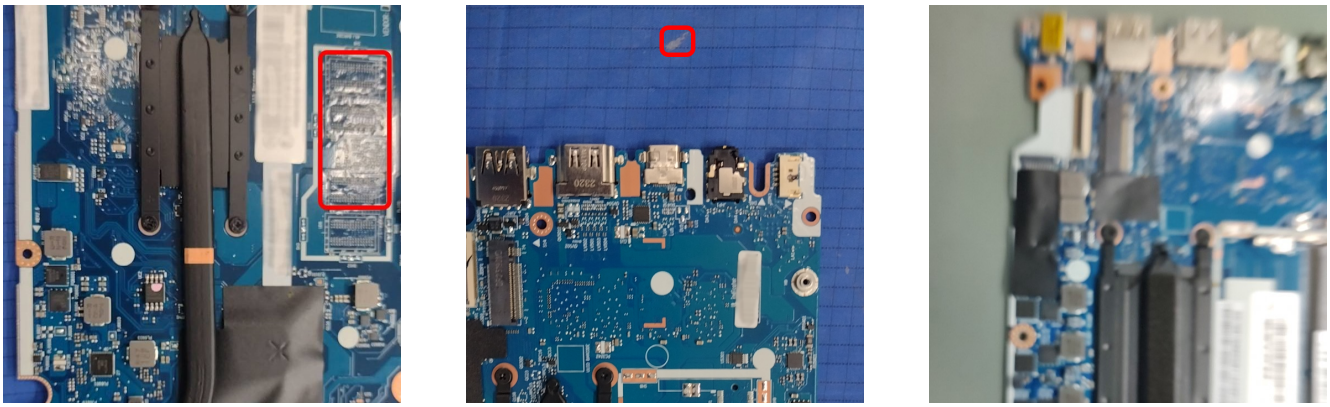
Image noise refers to random variations in brightness or color information in digital images. It appears as grainy or speckled patterns and can obscure fine details, making the image look less clear. Compared to DSLR cameras, images captured with mobile devices are more prone to higher levels of noise. This is due to the use of relatively low-cost sensors and lenses, especially in low-light conditions [Wang *et al.*, 2020]. Noisy regions of the image can make damages identification in PCBs difficult. Particularly, small damages are more affected by noise, as they have less available information and their corruption is more significative.

Metrics: Fast Noise Estimation and Signal-To-Noise Ratio To estimate the image noise, we adopt the classical approach proposed by [Immerkaer, 1996] called fast noise estimation. This method assumes that the noise is additive and Gaussian and relies on simple arithmetic operations and local neighborhood calculations, being computationally efficient. For each pixel, the local variance is determined using the differences between the pixel's value and its eight neighboring pixels in a 3x3 window. Specifically, the noise variance is approximated by averaging the squared differences between each central pixel and its neighbors, simplified by the Equation 6.

$$\sigma^2 \approx \frac{1}{6}\sum_{k=1}^{8}(I(i,j) - I(i_k, j_k))^2. \tag{6}$$

It is important to note that the divisor 6 does not come from the number of neighbors (8) or the $(n-1)$ correction of an unbiased sample variance estimator. Instead, it arises from the energy of the Laplacian-difference mask used in Immerkær's fast noise estimation method [Immerkaer, 1996]. This mask produces a response variance of $36\sigma^2$ for white Gaussian noise, leading to a standard deviation of 6σ. Thus, the factor 6 ensures consistency with the theoretical derivation, rather than reflecting degrees of freedom.

The final noise variance estimate is obtained by averaging these local variances across the entire image, providing a ro-



(a) Glare in the image induced an incorrect detection of mold.

(b) A scratch in the surface of the background was incorrectly detected as damage.

(c) A damage present in the PCB was not detected in the blurry photo.

Figure 2. Examples of quality issues of glare, background interference, and blur.

bust and computationally efficient way to estimate the overall noise level.

As a second metric of noise, we use the classical signal-to-noise ratio (SNR) in images. i.e., the logarithmic decibel ratio between the power of the signal and the power of the noise, estimated as the Equation 7.

$$\text{SNR} = 10 \log_{10} \left(\frac{\mu^2}{\sigma^2} \right), \tag{7}$$

Noise variance estimator

Following Immerkær Immerkaer [1996], let N denote the Laplacian-difference mask. For additive white Gaussian noise with standard deviation σ, the response variance is $36\sigma^2$, i.e.

$$\text{std}(I * N) = 6\sigma.$$

Using the Gaussian identity $E|X| = \sqrt{2/\pi}\,\text{std}(X)$ for a Gaussian variable X, we estimate:

$$\hat{\sigma} = \sqrt{\tfrac{\pi}{2}} \cdot \tfrac{1}{6} \cdot \frac{1}{(W-2)(H-2)} \sum_{x,y} \big|(I * N)(x,y)\big|.$$

The divisor 6 stems from the *kernel energy* of the filter N rather than from $(n-1)$ degrees of freedom as in unbiased variance estimation. We adopt the L_1 formulation to avoid multiplications while preserving accuracy.

where μ is the mean intensity of the image (representing the signal) and σ^2 is the variance of the noise. A higher SNR indicates that the signal (the useful information in the image) is much stronger than the noise (unwanted variations in pixel values), meaning the image quality is good with less noise. Conversely, a lower SNR indicates that the noise level is high relative to the signal, suggesting that the image quality is poor and noisy.

4.3 Background Interference

Regions of photographs that do not correspond to the surface of the PCB are in the background, while the regions of the PCB are in the foreground. As the mobile acquisition is unconstrained, there can be different settings of background. The background can contain objects or textures on surfaces that resemble damage in PCBs, leading the model to incorrectly detect damage in the background region. It is usual that PCBs in repair centers are handled on anti-static mantles, whose noisy texture can result in false detection, as illustrated in Figure 2b. It is relevant to mention that background interference is connected with the Area Ratio. The greater the Area Ratio, the lower the background surface is present in the image, lowering the chance of background interference.

Metrics: Variance, Edges, and Noise We evaluate the background interference by exploiting the segmentation model mentioned in the Subsection 4.1. We use the model to segment the PCB and apply an inverse mask, where the PCB regions are masked, displaying only the background. Then, we compute the following metrics on the masked images:

- **Background Variance**: we convert the image from RGB to grayscale and compute the variance of the values of the pixels of the background, i.e. the non-masked regions. The rationale is that the more variance in the background, the more misleading information can be present, as an ideal background would be a flat surface of a single color.
- **Variance of Edges in Background**: we apply the Canny edge detector [Rong *et al.*, 2014] on the masked image and compute the variance of the result. The more variance in the edges, the more objects and potentially misleading information are present in the background.
- **Background Noise**: We use the method of fast noise estimation in the masked image. The more noise in the background, the more misleading information can be present.

4.4 Glare

Glare in a photo refers to the phenomenon where bright spots or streaks of light appear in the image, often caused by intense and non-uniform light sources or reflections from shiny surfaces like polished metal [Esfahani and Wang, 2021]. The last cause is common in PCB images. This can reduce the overall quality of the photo by washing out details, reducing contrast, and creating unwanted visual distractions. Figure 2a illustrates an exemplary real case.

Metric: Glare Ratio We evaluate the level of glare by using an adaptation method proposed by Andalibi and Chandler [2017]. This method detects glare by computing local photometric features: intensity, saturation, and local contrast maps. Good candidates for glare regions exhibit high light intensity, low color saturation, and low luminance contrast. The glare map of a photo is computed as:

$$G_{photo}(x,y) = I(x,y) \times (1 - S(x,y)) \times (1 - C(x,y)), \quad (8)$$

where $I(x,y)$ is the local intensity, $S(x,y)$ is the local saturation in HSV color-space and $C(x,y)$ is the local contrast. More detailed information on how to calculate each component can be found in the original paper. The glare map G_{photo} is normalized to span the range $[0,1]$, and then it is applied a threshold of 0.99 resulting in a binary glare map. The global glare level is calculated as a ratio of the number of pixels with glare (above the threshold) over the total pixels of the image. In the original method the authors also used geometric information of the thresholded pixels and GPS data (azimuth and sun's elevation angles) to better estimate glare. In our work we use only the photometric features.

4.5 Blur

Blur in images refers to the loss of sharpness and clarity, resulting in a smoothed or smeared appearance. It can originate from several factors, often relating to movement and focus. One common cause is camera shake, where even a slight movement of the camera during a longer exposure can result in a blurry image. Additionally, improper focus settings, such as focusing on the wrong part of the PCB can cause parts of the image to appear blurry [Koik and Ibrahim, 2013]. The loss of detail caused by blur can impact the detection ability, as illustrated in Figure 2c.

Metric: Laplacian Variance We evaluate the global intensity of the blur on an image through the variance of the Laplacian, as proposed by Bansal *et al*. [2016], which produces a fast and reliable estimation of blur. The rationale behind this approach is that high variance in a typical, well-focused image indicates a wide range of responses, including both edge-like and non-edge-like features. Conversely, if the detected variance is low, it implies a narrow range of responses, suggesting that the image contains very few edges. Therefore, we can conclude that an image with very few edges is likely blurred.

5 Methodology

5.1 Experiments Overview

In this work, we focus on analyzing the impact of commonly observed image quality issues originating from mobile image acquisition on the performance of deep damage detection in PCBs. Each quality issue is approximately measured by one or more of the metrics detailed in Section 4.

To conduct this investigation, we train a set of models with well-known architectures in the object detection field, using the PCB image dataset described in Section 5.3. We randomly split the dataset into disjoint train and test sets, using the train set to train the models and evaluating them on the test set. For each image in the test set, we evaluate the performance metrics of the models along with the quality metrics.

We then conduct an iterative search for the optimal threshold of each metric, i.e., the threshold that best separates images with lower performance due to quality issues. By identifying these thresholds, we can evaluate the performance difference between the groups of poor-quality images and good-quality images.

Each metric's relevance is evaluated by considering the mean mAP difference between groups of images divided by the optimum threshold. If this difference exceeds 0.05 and the group of poor-quality images, as determined by this metric, comprises less than 30% of the total images, then the metric is considered relevant. In the context of object detection, a 0.05 improvement in mAP can significantly enhance practical performance, especially when the baseline scores are lower. The second condition ensures that the quality metric remains practical; it must not discard a large portion of the images, as this would set an unachievable standard.

After defining the optimal thresholds and relevant metrics, we use these criteria to separate all images and evaluate the differences in precision, recall, and mAP between the two groups, as well as the ratio of poor-quality images.

For each considered architecture, we follow the steps described below:

1. Train a model with the training set.
2. For each image in the test set, use the trained model to make inferences.
3. Compare the predictions with the ground truth for each image in the test set and evaluate the mAP, precision, and recall per image.
4. Compute the quality metrics proposed in Section 4 for each image in the test set.
5. For each metric distribution, iterate over the percentiles and use each percentile as a threshold to divide the images. Compute the difference in the mean mAP between the two groups and the proportion of poor-quality images.
6. The optimal threshold for each metric is the one that results in the highest difference in mAP between the groups. If the mAP difference is higher than 0.05 and the ratio of images below the threshold is lower than 30%, then the metric is considered relevant.
7. Evaluate the differences in recall, precision, and mAP between the groups of thresholded images for each metric.
8. After computing the optimal threshold for all metrics and identifying the relevant metrics, divide the images into groups where the relevant metrics are all equal to or above the thresholds (good quality images) and images with relevant metrics below the thresholds (poor-quality images). In the case of the Area Ratio metric, the good quality images are the ones with this value below the threshold.

With this procedure, we intend to identify which metrics from the proposed set can be useful in a damage detection

Table 3. Models used in the experiment.

Models	Family	Size (MB)	mAP in Test Set
YOLOv6	One-Stage	21.9	0.337
Mask-RCNN	Two-Stages	787.2	0.349

application. A practical application would be to develop a feedback system for capturing photos in a mobile damage detection application. This system would evaluate images from the camera in real-time, guiding users to take high-quality images, thereby improving detection performance.

5.2 Object Detector Architectures

To evaluate how the quality metrics impact the performance of damage detection, we conduct our experiments with two well-known architectures of object detectors, described in Section 2: YOLOv6 and Mask-RCNN.

YOLOv6 is a one-stage object detector that achieved the best results in the work of [Santiago *et al*., 2024], which compared different YOLO architectures for the task of CID detection. This model was trained using the open-source implementation from the Meituan Vision AI Department[1], specifically version N6, which has 10.4 million parameters. This is the second smallest architecture among the available versions of YOLOv6.

The **Mask-RCNN** is a two-stage object detector that achieved the best results in the work of [Alves *et al*., 2022], using the Swin Transformer network as the backbone. We used the Swin-S implementation provided by the MMDetection framework[2].

Thus, we experiment with object detectors from both families (one-stage and two-stage) to compare the consistency of the results. Both models were pre-trained with the COCO dataset and fine-tuned with our training dataset. They were trained using the standard hyperparameters of each framework. Table 3 describes the architectures in terms of family, size, and performance obtained on the test dataset. It can be seen that the detection capacity, measured by the mAP, is similar between both models despite the Mask-RCNN model being considerably larger.

5.3 Dataset

The dataset used in the experiments consists of real-world images collected from repair centers of a hardware manufacturer. These images are photos taken with smartphone cameras to document CIDs during the warranty inspection process. The photos were captured under diverse conditions of illumination, background, camera quality, distance, and position of PCBs. Experts analyzed the images and manually annotated the CIDs, indicating their locations with bounding boxes and categorizing them. The annotations follow the standard COCO dataset format [Lin *et al*., 2014].

There are five categories of CID in this dataset: break, scratch, pin damage, burn, and oxidation/mold. The dataset contains a total of 4,101 images and 8,736 annotated CIDs, with a nearly balanced number of examples in each class. To train and evaluate the models using a cross-validation setting, the dataset was randomly split into disjoint train and test sets. Table 4 summarizes the quantity of available data in each set.

Table 4. Dataset description.

Set	Images	Annotations	Scratch	Pin	Burn	Break	Oxidation
Train	3402	7486	1292	1439	843	1108	2804
Test	699	1250	281	279	104	312	274
Total	4101	8736	1573	1718	947	1420	3078

6 Results and Discussion

6.1 Metrics Analysis

After computing the metrics for all images in the test set, we analyze the distribution of values for each metric. Figure 3 shows these distributions, including those for performance metrics using the YOLO model. The Area Ratio metric is nearly uniformly distributed, with a mode of 1.0, indicating that these images consist entirely of the PCB surface. Other metrics exhibit right-skewed distributions, where most data points are concentrated at the lower end, but with a long tail extending toward higher values. This suggests that a few images are significantly degraded according to each metric.

To further examine redundancy among the proposed metrics, we computed a correlation matrix across all quality scores (Figure 8). The analysis revealed a very strong negative correlation between Area of Board Ratio and SNR Background ($\rho = -0.93$), suggesting that these two metrics capture overlapping information. A moderate negative correlation was also observed between Area of Board Ratio and Fast Noise Full ($\rho = -0.60$). In contrast, most of the remaining metrics, including Blur, Background Variance, and Vertical Tilt, showed only weak pairwise correlations, indicating that they provide complementary insights into image quality.

Although this redundancy analysis highlights overlaps between some features, we retained all metrics in the present study to maintain a comprehensive evaluation of interpretable, domain-specific indicators. Future work could explore dimensionality reduction techniques such as PCA, or feature selection methods to derive a more compact subset of the most informative metrics.

6.2 YOLO Model

The mAP achieved on the test set with the YOLO model is 0.388. Considering the YOLO model, after computing the optimal thresholds, we compare the distributions of mAP for good and bad quality images with respect to each metric. Figure 4 illustrates this comparison, with distributions approximated using the kernel density estimation (KDE) method. Each metric is plotted to show the distribution of mAP for both groups, including the means, the ratio of poor-quality images (referred to as the discard ratio), and the value of the optimal threshold. The metrics are arranged in ascending order based on the mean difference between the groups.

It is evident that the Area Ratio metric produces the highest mean difference in mAP, aligning with our initial intuition about the difficulty of detecting objects at smaller resolutions and specifically, removing images where the PCB occupies less than 15% of the total image area—corresponding to 7 images (1%) in this dataset—results in a mean gain of 0.21 in

[1]https://github.com/meituan/YOLOv6/

[2]https://github.com/open-mmlab/mmdetection/

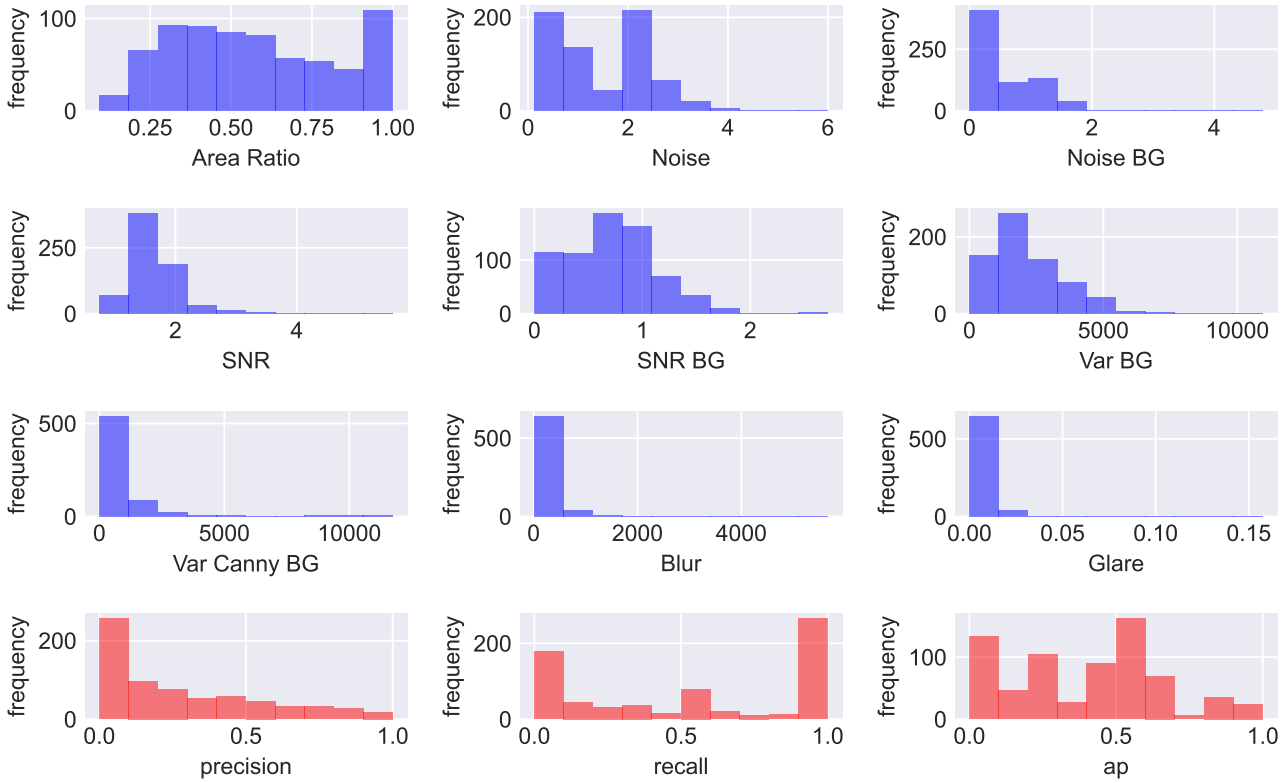


Figure 3. Distribution of quality metrics (in blue) and distributions of performance metrics (in red) in images of the test set.

mAP. This represents a significant improvement at a low cost. Additionally, images with this low Area Ratio are typically taken from a far distance, which is inappropriate for a damage detection application. The Noise metric also shows a similar trend but with a lower mean difference.

The Blur metric also produced a high mean difference, but with the high cost of discarding 25% of the images. Manual inspection of the dataset reveals that images blurred regions are very common. The metrics SNR BG, SNR, Var BG, Noise BG, and Var Canny BG showed similar results, discarding a small portion of images with moderate mean differences. These metrics are related to measures of noise, either across the entire image or specifically in the background. Finally, the Glare metric demonstrated no significant mean difference between distributions, indicating that glare does not have a substantial impact on performance in this scenario, contrary to our initial belief.

Based on the obtained results, we assess the relevance of the metrics using two criteria: a mean difference in mAP greater than 0.05 and a discarding ratio below 30%. All tested metrics are considered relevant except for Glare. However, since metrics can be correlated, we perform a further analysis to identify redundant metrics—those whose discarded image sets are entirely contained within the discarded image sets of other metrics. For that, we compute the intersection ratio between the discarded sets, computed as $\frac{|A \cap B|}{|A|}$. This ratio represents the proportion of elements in set A that are also in set B. The results are presented in Table 5, highlighting only those intersection ratios exceeding 15%.

Our analysis reveals that the set of images discarded by the Noise metric is fully contained within the set discarded by the Blur metric, indicating that Noise is redundant in this dataset. This finding is consistent with our observations, as the Blur metric discards a significantly larger portion of the data (25%, or 175 images) compared to the Noise metric, which discards only 7 images. Additionally, blurred images are often associated with noise issues, further justifying the redundancy of the Noise metric. Other noise-related metrics also show a large intersection ratio with the Blur discarded set, particularly those concerning background interference, such as SNR BG, Var BG, Noise BG, and Var Canny BG. Additionally, while the background metrics are related to their corresponding global metrics (e.g., SNR BG and SNR), the background metrics capture specific details that are valuable, making them worth retaining as relevant metrics.

Consequently, by applying this final set of non-redundant quality metrics to filter the dataset, we observed a significant impact on performance, as shown in Figure 5. Discarding 30% of the images based on these criteria yielded a mean gain of 0.1 in Precision, 0.14 in Recall, and 0.15 in mAP. Given that these metrics primarily identify artifacts such as blur and noise, this considerable improvement supports our hypothesis that such images can introduce noise or inaccuracies into the detection process. This filtering ensures that only high-quality, informative images are retained for training the detection algorithm.

This principle is aligned with the idea that higher-quality inputs lead to more reliable outputs. By removing lower-quality images, which could be cluttered with irrelevant details or artifacts, we ensures that the remaining dataset is more representative of the actual damage features. In practical terms, our metrics demonstrates that by improving the quality of

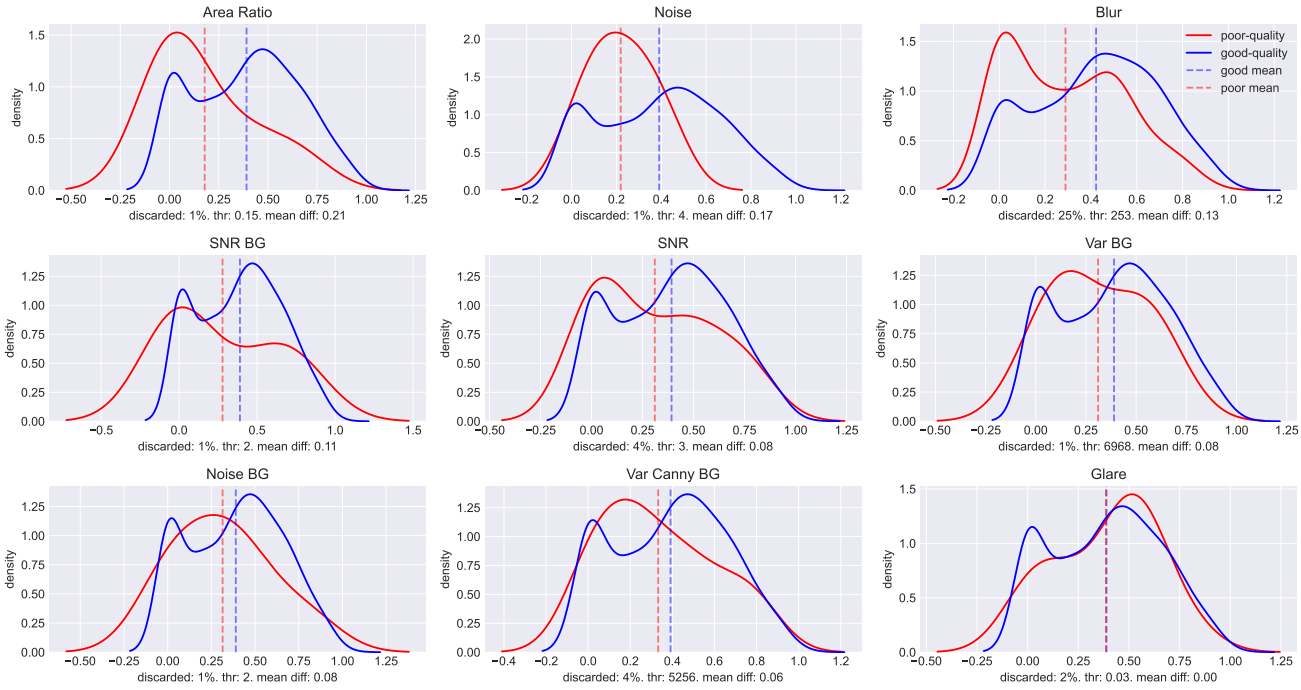


Figure 4. Difference between distribution of mAP in poor and good quality images by each metric, using the YOLO model. The metrics are ordered by the value of the mean difference in mAP. The percentage of poor-quality images (discarded), the threshold value (thr) and the mean difference (diff) between distributions are depicted for each metric.

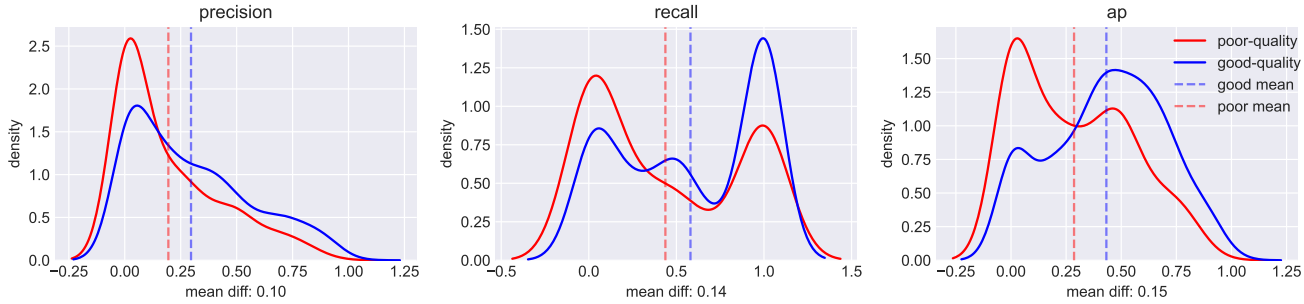


Figure 5. Difference between distributions of Precision, Recall, and mAP between poor and good quality images, considering all the relevant metrics for the YOLO model. The discarded set of poor-quality images constitutes 30% of all images in the test set.

the input data, one can achieve better performance metrics without necessarily increasing computational resources. This is particularly valuable in industrial applications where both accuracy and efficiency are critical.

6.3 Mask-RCNN Model

The mAP achieved on the test set with the Mask R-CNN model is 0.275, which is significantly lower than the mAP achieved with the YOLO model. However, the results concerning optimal thresholds and the distribution of mAP for good versus bad quality images are consistent with those obtained using the YOLO model. Figure 6 illustrates those results.

The metrics exhibiting the largest differences—Area Ratio, Noise, and Blur—showed similar thresholds, discarded ratios, and slightly lower mean differences compared to the YOLO model. The noise-related metrics also demonstrated comparable behavior. The set of relevant metrics was nearly identical to those of the YOLO model, with the exception of the Var BG metric, which showed a low mean difference in this model. The Glare metric was once again not considered relevant due to its low difference. Curiously, the discarded ratio of the Glare metric was 52%, which may be due to its lack of correlation with performance. Additionally, the intersection ratio between the discarded sets was very similar to that obtained with the YOLO model.

Figure 6 presents the results of discarding images using all relevant metrics. It indicates a mean difference of 0.12 in Precision, 0.16 in Recall, and 0.14 in mAP, achieved by discarding 30% of the images in the test set. These results are again comparable to those obtained with the YOLO model. The consistency between the results from models of the two major deep object detection families suggests that our proposed metrics are generally applicable to the problem of PCB damage detection.

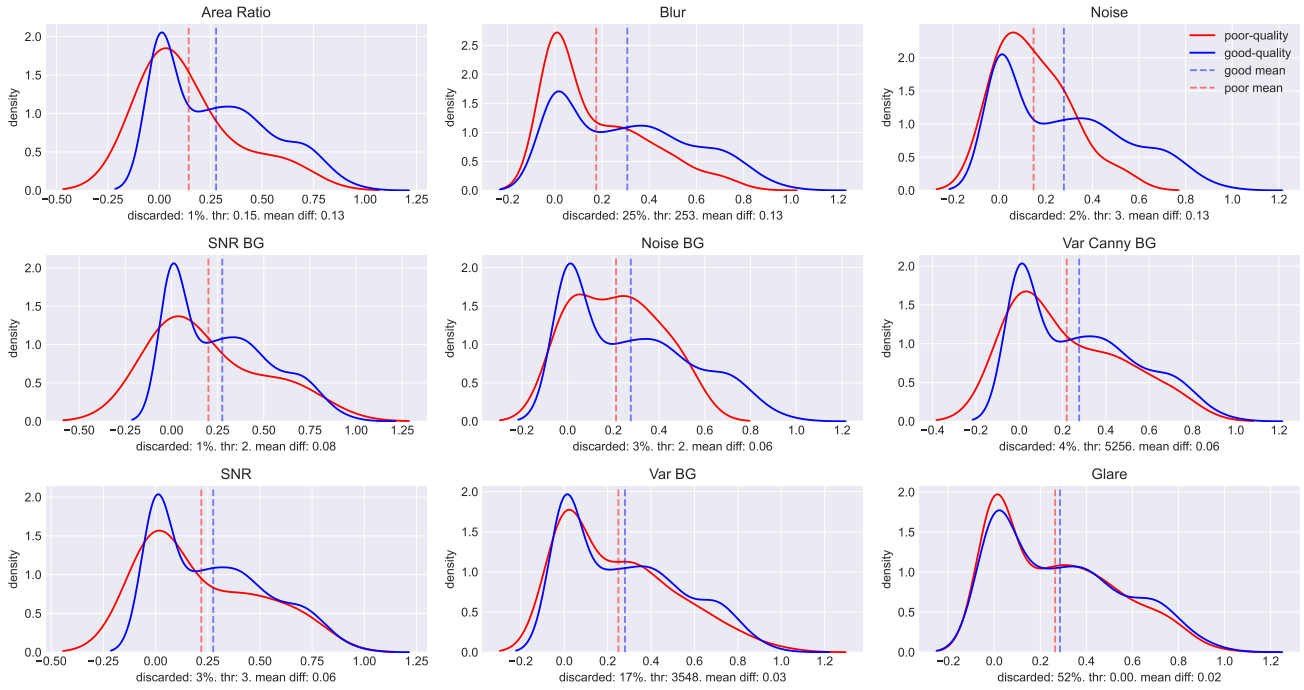


Figure 6. Difference between distribution of mAP in poor and good quality images by each metric, using the Mask-RCNN model. The metrics are ordered by the value of the mean difference in mAP.

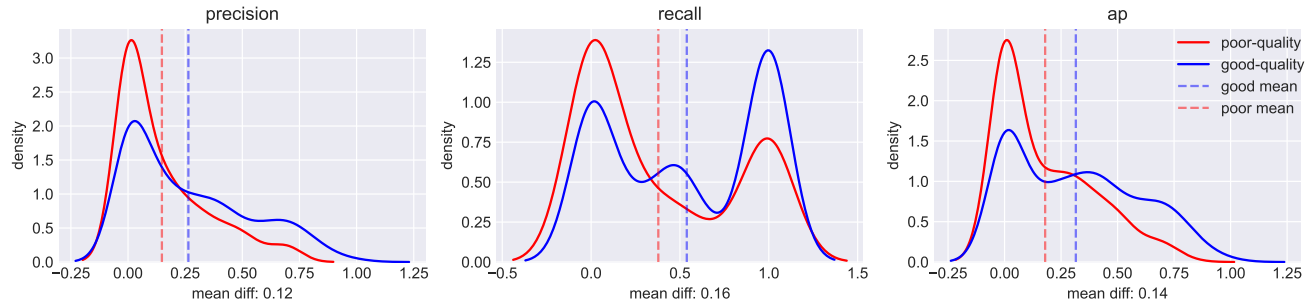


Figure 7. Difference between distributions of Precision, Recall, and mAP between poor and good-quality images, considering all the relevant metrics for the Mask-RCNN model. The discarded set of poor-quality images constitutes 30% of all images in the test set.

6.4 Discussion

The experimental results demonstrated that most of the proposed metrics significantly impact the model's performance in PCB deep damage detection, with the notable exception of the Glare metric. In particular, the Area Ratio, which measures the proportion of the image occupied by a PCB, had a remarkable effect. This aligns with our initial hypothesis that factors such as distance, size, and detail of the object, that are approximated by the Area Ratio, are crucial for performance. Metrics related to blur and noise also had substantial impacts. While noise can be challenging for users to control, blur and photo distance are more manageable depending on the user's skill.

We emphasize that our metrics can be computed for any image without requiring reference images and with relatively low computational cost. A practical application of these metrics in a mobile app could involve a feedback system that calculates relevant metrics in real time, providing users with immediate feedback to capture higher-quality photos prior to damage detection. As demonstrated by these experiments, this approach could significantly enhance detection capabilities without incurring the high costs associated with training larger models.

Regarding the Glare Ratio metric, although real examples demonstrated that glare could lead to incorrect damage detection, we found no significant correlation between the proposed glare metric and detection performance. Consequently, we conducted further investigations to explore this issue.

Firstly, as shown in Figure 3, the Glare Ratio, like most other metrics, follows a power law distribution with a heavy tail. The maximum glare value is 0.157, while the second-highest value is 0.06 — less than half of the maximum. The third quartile is 0.004, and the median is 0, indicating that most images exhibit no measurable glare under the chosen method. In Figure 9, we present the top three images with the highest glare values along with their respective mAP scores. The top two images in terms of glare had relatively high mAP scores of 0.6 and 0.5.

It is evident that what the chosen method identified as glare corresponds to reflections on the white background surface (see Figures 9a and 9b) and on a white label (see Figure 9c). It

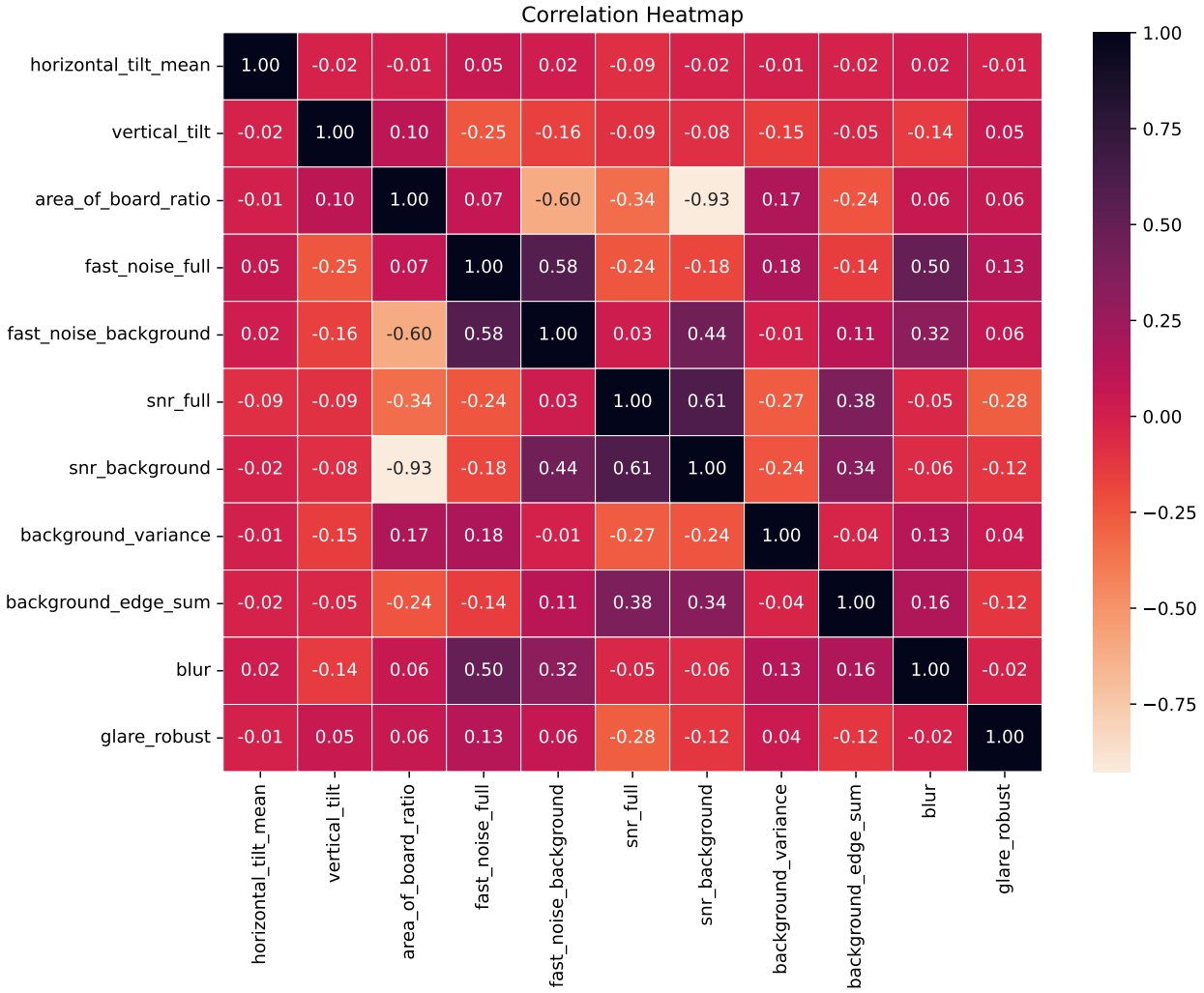


Figure 8. Correlation heatmap of the proposed image quality metrics. Strong negative correlation is observed between Area of Board Ratio and SNR Background ($\rho = -0.93$), while most other metrics exhibit weak pairwise correlations.

is important to emphasize that this metric considers the ratio of the identified glare area relative to the total image area. Therefore, the larger the area of identified glare, the higher the metric value will be.

Considering this evidence, we propose a hypothesis as to why the Glare metric could not be directly associated with detection performance. In our previous experiments, all observed cases where glare caused incorrect detection occurred due to reflections within the board, specifically on its reflective parts, as illustrated in Figure 2a. Since the detection models also rely on contextual information from the region where damage occurs, for a reflection to be misidentified as damage, such as mold, it would need to appear in an area of the PCB containing circuit elements.

To test this hypothesis, we performed an experiment where we measured glare only in the PCB region, using a semantic segmentation approach, similar to what we did with other metrics. However, the resulting distribution of this modified glare metric is nearly identical to the original one. The biggest differences were observed in images with a large white background area, such as those shown in Figures 9a and 9b, but these cases are outliers. Figure 10 illustrates the top three images with the highest glare on the PCB surface using semantic segmentation. Notably, the image ranked as the top-1 in glare on the PCB surface corresponds to the top-3 image when considering glare in the entire image.

Finally, we conducted the same analysis on the impact of the modified Glare metric on damage detection, identifying an optimal threshold and distinguishing poor-quality from good-quality images. The results are shown in Figure 11, where it becomes clear that this metric has no relationship with detection performance, even when glare is measured only on the PCB surface.

The glare regions identified in both Figures 9 and 10 are large, continuous, fully white areas that lack visual information resembling damage. This suggests that the factors causing glare to impact detection are its location, shape, and texture, rather than its size, which is what the current metric measures. In other words, the metric evaluates the global effect of glare, while what should be considered is a local evaluation of its shape, position, and texture. We can conclude that the chosen metric is not adequate for this use.

Interestingly, the Glare metric did not show a meaningful correlation with detection performance, a result that contradicts our initial hypothesis. We speculate that this is due to a fundamental misalignment between the standard photometric

Table 5. Relationship between discarded sets of poor quality images considering the YOLO model. The Intersection Ratio is computed as the ratio of the size of the intersection of two sets A and B to the size of the set A

Metric	Discarded Set Size	Intersection Ratio
Area Ratio	7	SNR BG: 43%, SNR: 43%
Noise	7	Blur: 100%, Noise BG: 29%, Var Canny BG: 29%
Blur	175	-
SNR BG	7	Area Ratio: 43%, Blur: 29%, SNR: 86%, Var Canny BG: 29%
SNR	28	Blur: 36%, SNR BG: 21%, Var Canny BG: 36%
Var BG	7	Blur: 43%
Noise BG	7	Noise: 29%, Blur: 71%, Var Canny BG: 57%
Var Canny BG	28	Blur: 79%, SNR: 36%

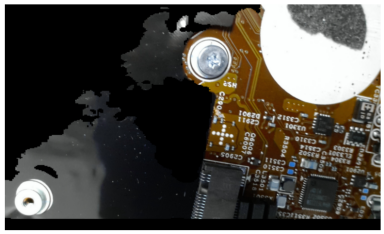
(a) Image with highest Glare (0.157). mAP of 0.6.

(b) Second-highest Glare (0.06). mAP of 0.5.

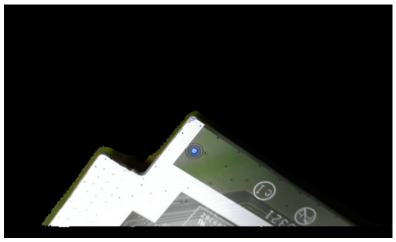
(c) Third highest Glare (0.059). mAP of 0.2.

Figure 9. Top-three images with the highest value of the Glare metric.

(a) Top-1 Glare on PCB (0.054). mAP of 0.2.



(b) Top-2 Glare on PCB (0.054). mAP of 0.6.



(c) Top-3 Glare on PCB (0.05). mAP of 0.8.

Figure 10. Top-three images with the highest value of the Glare metric only in the PCB surface, using semantic segmentation.

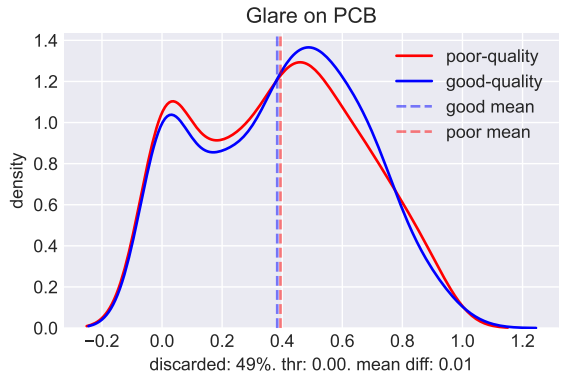


Figure 11. Distributions of mAP in images separated by the optimum threshold considering the Glare metric applied to the PCB surface.

definition of glare and its actual manifestation in PCB images. The adopted metric is designed to detect large, homogeneous glare regions typical of natural scenes. However, in PCBs, problematic glare often manifests as highly localized, intense reflections off small metallic components. These localized artifacts may not significantly impact the global image quality score, yet they can be severe enough to cause local detection errors, as illustrated in Figure 2a. Furthermore, our dataset may contain an insufficient number of extreme glare cases to statistically reveal its impact. Future work should investigate glare descriptors tailored to PCB characteristics, potentially combining photometric cues with geometric or contextual information (e.g., component location) to develop a more sensitive and domain-appropriate measure.

While metrics such as Area Ratio and Background Interference are specific to the geometry of printed circuit boards, the proposed methodology for developing interpretable, no-reference image quality metrics is generalizable. The process of identifying domain-specific sources of degradation (such as glare, blur, or occlusion) and mapping them to measurable features can be adapted to other fields. Therefore, even though the exact definitions of the metrics may not be directly transferable, the principle of tailoring the quality assessment to the operational environment remains applicable in contexts like automotive inspection, medical imaging, or robotic vision. This study, however, does not evaluate the generalization of these metrics beyond PCBs, and the conclusions should be interpreted within the boundaries of this application.

In this study, quality thresholds were selected empirically based on mAP gaps, which provided a straightforward and interpretable method to identify meaningful cutoffs for our analysis. However, we acknowledge that this static approach may not fully capture context-sensitive variability—such as different lighting conditions, camera models, or PCB types—nor guarantee robustness across diverse deployment scenarios. As a promising direction for future work, we propose investigating more sophisticated thresholding mechanisms. This could include dynamic thresholds that adapt based on real-time acquisition parameters or learning-based approaches where a model learns to predict the optimal quality cutoff for a given image context. Exploring such strategies could significantly enhance the adaptability and generalizability of quality-based filtering in real-world applications.

7 Conclusion

In this study, we explored the impact of domain-specific, no-reference image quality metrics on the performance of deep learning models for PCB damage detection. By evaluating 9 metrics, including 5 novel ones, we were able to provide valuable insights into how image quality influences detection performance. Our findings demonstrate that optimal thresholds for these metrics can significantly affect model accuracy, with mean differences of 0.15 in mAP, 0.14 in recall, and 0.1 in precision observed between high and low-quality images. Approximately 30% of the test images were found to be below the quality threshold, highlighting the importance of addressing image quality in practical applications.

The results suggest that incorporating these metrics into mobile damage detection applications can enhance performance by filtering out poor-quality images before inference. This approach not only improves detection accuracy but also offers actionable feedback to users, enabling them to capture higher-quality photos. Overall, our work provides a foundation for developing more effective mobile image processing tools and underscores the importance of image quality in real-world computer vision tasks.

In future research we intend to focus on automatic correction of quality issues through advanced image processing and deep learning techniques. For instance, techniques such as image denoising and deblurring could be employed. Additionally, methods such as image super-resolution could enhance the quality of low-resolution images. Integrating generative adversarial networks (GANs) or self-supervised learning approaches could also be explored to further improve image quality and ensure accurate defect detection. By advancing these methods, it may be possible to mitigate quality issues in real-time and enhance the overall efficacy of mobile damage detection systems.

Declarations

Acknowledgements

We thank Lenovo for the incentive to this research provided under the Brazilian Informatics Law. We thank all the participants in the empirical study and the Systems and Databases Laboratory (LSBD) members for their support.

Funding

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

Authors' Contributions

LC, BC, YL contributed to the conception of this study. LC, PS, and LS performed the experiments and analyzed the results. LC is the main contributor and writer of this manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Due to the confidential nature of the dataset and existing non-disclosure agreements, the data and code utilized in this study cannot be made publicly available.

References

Adibhatla, V. A., Chih, H.-C., Hsu, C.-C., Cheng, J., Abbod, M. F., and Shieh, J.-S. (2020). Defect detection in printed circuit boards using you-only-look-once convolutional neural networks. *Electronics*, 9(9):1547. DOI: 10.3390/electronics9091547.

Al-Rahlawee, A. T. H. and Rahebi, J. (2021). Multilevel thresholding of images with improved otsu thresholding by black widow optimization algorithm. *Multimedia Tools and Applications*, 80(18):28217–28243. DOI: 10.1007/s11042-021-10860-w.

Alves, D., Farias, V., Chaves, I., Chao, R., Madeiro, J. P., Gomes, J. P., and Machado, J. (2022). Detecting customer induced damages in motherboards with deep neural networks. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE. DOI: 10.1109/ijcnn55064.2022.9892047.

Andalibi, M. and Chandler, D. M. (2017). Automatic glare detection via photometric, geometric, and global positioning information. *Electronic Imaging*, 29:77–82. DOI: 10.2352/issn.2470-1173.2017.19.avm-024.

Bai, Y., Zhang, Y., Ding, M., and Ghanem, B. (2018). Sodmtgan: Small object detection via multi-task generative adversarial network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 206–221. DOI: 10.1007/978-3-030-01261-8$_1$3.

Bansal, R., Raj, G., and Choudhury, T. (2016). Blur image detection using laplacian operator and open-cv. In *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*, pages 63–67. IEEE. DOI: 10.1109/sysmart.2016.7894491.

Basu, M. (2002). Gaussian-based edge-detection methods-a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 32(3):252–260. DOI: 10.1109/tsmcc.2002.804448.

Bergstrom, A. C. and Messinger, D. W. (2023). Image quality and object detection performance of convolutional neural networks. In *Pattern Recognition and Tracking XXXIV*, volume 12527, pages 159–177. SPIE. DOI: 10.1117/12.2663779.

Bosse, S., Maniry, D., Müller, K.-R., Wiegand, T., and Samek, W. (2017). Deep neural networks for no-reference and full-reference image quality assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4321–4329. DOI: 10.1109/tip.2017.2760518.

Cabral, L., Farias, V., Sena, L., Chaves, I., Pordeus, J. P., Santiago, J. P., Sa, D., Machado, J., and Madeiro, J. P. (2023). An active learning approach for detecting customer induced damages in motherboards with deep neural networks. *Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC)*, 21:29–42. DOI: 10.21528/lnlm-vol21-no2-art3.

Coca, L.-G., Cusmuliuc, C. G., and Iftene, A. (2021). Automatic tarmac crack identification application. *Procedia Computer Science*, 192:478–486. DOI: 10.1016/j.procs.2021.08.049.

Dallet, C., Kareem, S., and Kale, I. (2014). Real time blood image processing application for malaria diagnosis using mobile phones. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2405–2408. IEEE. DOI: 10.1109/iscas.2014.6865657.

Doğan, G. and Ergen, B. (2022). A new mobile convolutional neural network-based approach for pixel-wise road surface crack detection. *Measurement*, 195:111119. DOI: 10.1016/j.measurement.2022.111119.

Du, B., Wan, F., Lei, G., Xu, L., Xu, C., and Xiong, Y. (2023). Yolo-mbbi: Pcb surface defect detection method based on enhanced yolov5. *Electronics*, 12(13):2821. DOI: 10.3390/electronics12132821.

Esfahani, M. A. and Wang, H. (2021). Robust glare detection: Review, analysis, and dataset release. *arXiv preprint arXiv:2110.06006*. DOI: 10.48550/arxiv.2110.06006.

Everingham, M. *et al*. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338. DOI: 10.1007/s11263-009-0275-4.

Fu, L., Liu, Z., Majeed, Y., and Cui, Y. (2018). Kiwifruit yield estimation using image processing by an android mobile phone. *IFAC-PapersOnLine*, 51(17):185–190. DOI: 10.1016/j.ifacol.2018.08.137.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE. DOI: 10.1109/cvpr.2012.6248074.

Gong, T., Jiang, Y., He, L., Wu, X., and Liu, J. (2023). Pmddnet: A novel one-stage lightweight network for multi-category defect inspection of pc motherboards. *Available at SSRN 4403073*. DOI: 10.2139/ssrn.4403073.

Grigorescu, S., Trasnea, B., Cocias, A., and Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37(3):362–386. DOI: 10.1002/rob.21918.

Gummadi, D., Chan, P. H., Wang, H., and Donzella, V. (2023). Correlating traditional image quality metrics and dnn-based object detection: a case study with compressed camera data. *Authorea Preprints*. DOI: 10.36227/techrxiv.24566371.

Hao, Y., Pei, H., Lyu, Y., Yuan, Z., Rizzo, J.-R., Wang, Y., and Fang, Y. (2023). Understanding the impact of image quality and distance of objects to object detection performance. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11436–11442. IEEE. DOI: 10.1109/iros55552.2023.10342139.

He, F., Tang, S., Mehrkanoon, S., Huang, X., and Yang, J. (2020). A real-time pcb defect detector based on supervised and semi-supervised learning. In *ESANN*, pages 527–532. Available at:`https://cris.maastrichtuniversity.nl/en/publications/a-real-time-pcb-defect-detector-based-on-supervised-a`

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international*

conference on computer vision, pages 2961–2969. DOI: 10.1109/iccv.2017.322.

Henning, K.-F., Fritze, A., Gillich, E., Mönks, U., and Lohweg, V. (2015). Stable image acquisition for mobile image processing applications. In *Digital Photography XI*, volume 9404, pages 177–186. SPIE. DOI: 10.1117/12.2076146.

Hu, B. and Wang, J. (2020). Detection of pcb surface defects with improved faster-rcnn and feature pyramid network. *Ieee Access*, 8:108335–108345. DOI: 10.1109/access.2020.3001349.

Immerkaer, J. (1996). Fast noise variance estimation. *Computer vision and image understanding*, 64(2):300–302. Available at:`https://scispace.com/papers/fast-noise-variance-estimation-29fk5iihql`.

Jin, C. M., Omar, Z., and Jaward, M. H. (2016). A mobile application of american sign language translation via image processing algorithms. In *2016 IEEE Region 10 Symposium (TENSYMP)*, pages 104–109. IEEE. DOI: 10.1109/tenconspring.2016.7519386.

Jing, J., Liu, S., Wang, G., Zhang, W., and Sun, C. (2022). Recent advances on image edge detection: A comprehensive review. *Neurocomputing*, 503:259–271. DOI: 10.1016/j.neucom.2022.06.083.

Kheradmandi, N. and Mehranfar, V. (2022). A critical review and comparative study on image segmentation-based techniques for pavement crack detection. *Construction and Building Materials*, 321:126162. DOI: 10.1016/j.conbuildmat.2021.126162.

Koik, B. T. and Ibrahim, H. (2013). A literature survey on blur detection algorithms for digital imaging. In *2013 1st International Conference on Artificial Intelligence, Modelling and Simulation*, pages 272–277. IEEE. DOI: 10.1109/aims.2013.50.

Kong, L., Ikusan, A., Dai, R., and Zhu, J. (2019). Blind image quality prediction for object detection. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 216–221. DOI: 10.1109/MIPR.2019.00046.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. DOI: 10.1109/5.726791.

Lei, J., Gao, X., Feng, Z., Qiu, H., and Song, M. (2018). Scale insensitive and focus driven mobile screen defect detection in industry. *Neurocomputing*, 294:72–81. DOI: 10.1016/j.neucom.2018.03.013.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., *et al*. (2022). Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*. DOI: 10.48550/arxiv.2209.02976.

Li, Y.-T., Kuo, P., and Guo, J.-I. (2020). Automatic industry pcb board dip process defect detection with deep ensemble method. In *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, pages 453–459. IEEE. DOI: 10.1109/isie45063.2020.9152533.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer. DOI: 10.1007/978-3-319-10602-1_48.

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., and Sanchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88. DOI: 10.1016/j.media.2017.07.005.

Liu, L., Ke, C., and Lin, H. (2023). Mobile-deep based pcb image segmentation algorithm research. *Computers, Materials & Continua*, 77(2). DOI: 10.32604/cmc.2023.042582.

Liu, L., Li, H., and Gruteser, M. (2019). Edge assisted real-time object detection for mobile augmented reality. In *The 25th annual international conference on mobile computing and networking*, pages 1–16. DOI: 10.1145/3300061.3300116.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., and Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. *International journal of computer vision*, 128(2):261–318. DOI: 10.1007/s11263-019-01247-4.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022. DOI: 10.1109/iccv48922.2021.00986.

Marengoni, M. and Stringhini, S. (2009). Tutorial: Introdução à visão computacional usando opencv. *Revista de Informática Teórica e Aplicada*, 16(1):125–160. Available at:`https://seer.ufrgs.br/index.php/rita/article/view/rita_v16_n1_p125`.

Mittal, A., Moorthy, A. K., and Bovik, A. C. (2012). No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708. DOI: 10.1109/tip.2012.2214050.

Morikawa, C., Kobayashi, M., Satoh, M., Kuroda, Y., Inomata, T., Matsuo, H., Miura, T., and Hilaga, M. (2021). Image and video processing on mobile devices: a survey. *the visual Computer*, 37(12):2931–2949. DOI: 10.1007/s00371-021-02200-8.

Nam, W., Youn, T., and Ha, C. (2025). No-reference image quality assessment with moving spectrum and laplacian filter for autonomous driving environment. *Vehicles*, 7(1):8. DOI: 10.3390/vehicles7010008.

Nayak, A., Chakraborty, S., and Swain, D. K. (2023). Application of smartphone-image processing and transfer learning for rice disease and nutrient deficiency detection. *Smart Agricultural Technology*, 4:100195. DOI: 10.1016/j.atech.2023.100195.

Nejati, H., Pomponiu, V., Do, T.-T., Zhou, Y., Iravani, S., and Cheung, N.-M. (2016). Smartphone and mobile image processing for assisted living: Health-monitoring apps powered by advanced mobile imaging algorithms. *IEEE Signal Processing Magazine*, 33(4):30–48. DOI: 10.1109/msp.2016.2549996.

Noh, J., Bae, W., Lee, W., Seo, J., and Kim, G. (2019).

Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9725–9734. DOI: 10.1109/iccv.2019.00982.

Oike, Y. (2022). Expanding human potential through imaging and sensing technologies. In *2022 International Electron Devices Meeting (IEDM)*, pages 1–2. IEEE. DOI: 10.1109/iedm45625.2022.10019533.

Picon, A., Alvarez-Gila, A., Seitz, M., Ortiz-Barredo, A., Echazarra, J., and Johannes, A. (2019). Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. *Computers and Electronics in Agriculture*, 161:280–290. DOI: 10.1016/j.compag.2018.04.002.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788. DOI: 10.1109/cvpr.2016.91.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28. DOI: 10.1109/tpami.2016.2577031.

Rong, W., Li, Z., Zhang, W., and Sun, L. (2014). An improved canny edge detection algorithm. In *2014 IEEE international conference on mechatronics and automation*, pages 577–582. IEEE. DOI: 10.1109/icma.2014.6885761.

Santiago, J. P., Farias, V., Sena, L., Pordeus, J. P., and Machado, J. (2024). Real-time detection of customer-induced damage in printed circuit boards using mobile devices and yolo detectors. *Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC)*, 22:17–31. DOI: 10.21528/lnlm-vol22-no1-art2.

Shukurov, J. (2024). Improve accessibility for low vision and blind people using machine learning and computer vision. *arXiv preprint arXiv:2404.00043*. DOI: 10.48550/arxiv.2404.00043.

Siddiqui, S. A., Agne, S., Dengel, A., Ahmed, S., *et al*. (2022). Are deep models robust against real distortions? a case study on document image classification. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 1628–1635. IEEE. DOI: 10.1109/icpr56361.2022.9956167.

Szeliski, R. (2022). *Computer vision: algorithms and applications*. Springer Nature. DOI: 10.5860/choice.48-5140.

Talebi, H. and Milanfar, P. (2018). Nima: Neural image assessment. *IEEE Transactions on Image Processing*, 27(8):3998–4011. DOI: 10.1109/tip.2018.2831899.

Tarvainen, A. and Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30. DOI: 10.48550/arxiv.1703.01780.

Voulodimos, A., Doulamis, N., Doulamis, A., and Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018. DOI: 10.1155/2018/7068349.

Wang, Y., Huang, H., Xu, Q., Liu, J., Liu, Y., and Wang, J. (2020). Practical deep raw image denoising on mobile devices. In *European Conference on Computer Vision*, pages 1–16. Springer. DOI: 10.1007/978-3-030-58539-6$_1$.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612. DOI: 10.1109/tip.2003.819861.

Xu, M., Zhang, Z., Hu, H., Wang, J., Wang, L., Wei, F., Bai, X., and Liu, Z. (2021). End-to-end semi-supervised object detection with soft teacher. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3060–3069. DOI: 10.1109/iccv48922.2021.00305.

Zhai, G. and Min, X. (2020). Perceptual image quality assessment: a survey. *Science China Information Sciences*, 63:1–52. DOI: 10.1007/s11432-019-2757-1.

Zhai, G., Min, X., Gu, K., and Yang, X. (2021). Perceptual image quality assessment: A survey. *arXiv preprint arXiv:2109.04380*. DOI: 10.1007/s11432-019-2757-1.

Zhang, L., Zhang, L., Mou, X., and Zhang, D. (2011). Fsim: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386. DOI: 10.1109/tip.2011.2109730.

Zhou, Y., Yuan, M., Zhang, J., Ding, G., and Qin, S. (2023). Review of vision-based defect detection research and its perspectives for printed circuit board. *Journal of Manufacturing Systems*, 70:557–578. DOI: 10.1016/j.jmsy.2023.08.019.

Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., and Liang, J. (2018). Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, pages 3–11. Springer. DOI: 10.1007/978-3-030-00889-5$_1$.

Zhu, P., Zhai, G., Min, X., Gu, K., and Liu, Z. (2020). Metaiqa: Deep meta-learning for no-reference image quality assessment. *IEEE Transactions on Image Processing*, 29:6080–6095. DOI: 10.1109/cvpr42600.2020.01415.

Zou, Z., Shi, Z., Guo, Y., Ye, J., and Others (2019). Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*. DOI: 10.1109/jproc.2023.3238524.