Image-based Methods for Inspection of Printed Circuit Boards

Nicholas Gardner1, Cooper Linsky1, Everardo FriasRios2, Nenad Nenadic1

1Golisano Institute for Sustainability, Rochester Institute of Technology, 1 Lomb Memorial Dr, Rochester, NY 14623

2CoreCentric Solutions, 191 E. North Avenue, Carol Stream, IL, 60188

**Primary Topic Area**: Remanufacturing & Material Reuse

**Secondary Topic Area**: Building a Sustainable Circular Economy for Materials & Products

# Abstract

The remanufacturing and reusing of printed circuit boards (PCBs) is an important component of the circular economy. Current practices on the remanufacturing floor employ several manual processing steps, including identification of the PCB type, localization and degradation assessment of various components, and keying data entries into the system. The repetitive nature of the manual processing steps places a heavy burden on technicians, who tire, make mistakes, and introduce subjectivity into the assessment process. Furthermore, these tedious tasks make employees unhappy, which leads to high turnover. Machine learning has become state-of-the-art for automating inspection tasks but typically requires a large amount of labeled data. We describe the process of introducing machine learning and computer vision for two tasks associated with the remanufacturing process: 1) part number identification and 2) localization of components with an assessment of their degradation. The components selected for the visual inspection were light-emitting diodes (LEDs) because the traditional assessment was based on manual visual inspection. The solution incorporated commercially-available solutions, newly-trained models, and novel approaches for relaxing requirements for machine learning development, all integrated into one development environment. Specifically, the part-number identification solution leveraged the Google Cloud Vision API for extracting character strings from images. The solution for the degradation assessment involved two steps, localizing components and classifying their health. The localization of components used a novel approach that employed classical, deterministic image processing and machine learning. The localized LED sub-images were classified using a custom-trained deep-learning model. Because labeling can also be time-consuming and expensive, we propose a localization scheme that leverages the efficacy of deep learning and significantly reduces the time required to label a dataset. LED localization and assessment performance showed a better than 97% detection rate on the validation data for a specific PCB when the false detection rate was held below 5%. In addition to software development, we explored and discussed trade-offs related to different options for image captures, industrial cameras, and smart devices relative to the use of the captured images.

# Introduction

*Cores* and *product returns* arrive on the manufacturing floor randomly, under unknown conditions. The sorting and handling of these cores require flexible processes, which currently depend on manual labor. Moreover, the variation of products is high, often featuring tens of thousands of classes of parts that must be identified and grouped by different applications and manufacturers. The receiving process starts with a manual teardown identifying the core with a *part number* identifier. With a large inbound of core, up to thousands of pieces per week, the sorting and receiving require speed, which often sacrifices the accuracy of the manual condition assessment. Errors in condition assessment create waste downstream because they failed to sort out cores cost-prohibitive to remanufacture. Moreover, failure to identify unusable cores wastes handling and storage capacity.

Training technicians takes 4-8 weeks and includes imparting the requisite domain knowledge, internalizing the knowledge, and practicing to attain the necessary efficiency. However, monotonous tasks lead to low employee retention in core receiving. Business motivation to adopt machine-learning-based automation includes efficiency improvement, increased accuracy, and a path to expanded capabilities.

Analysis of *light-emitting diode* (LED) degradation from the *printed circuit board* (PCB) images required two steps: 1) the localization of LED sub-images and 2) modeling degradation using the data of the sub-images. Two widely used data-driven localization models are *you-only-look-once* (YOLO) and U-Net. YOLO is a single-stage detector: it detects a bounding box and classifies the associated object in one shot (Diwan et al., 2022). YOLO has been continuously improving over the last six years (Redmon et al., 2016; Redmon & Farhadi, 2017, 2018; C.-Y. Wang et al., 2022; C. Y. Wang et al., 2020) and has been successfully applied to detect electronic components on PCBs (Li et al., 2019). However, YOLO’s bounding boxes are aligned with the image axes, which is a limitation for LED detection because PCB layouts employ both aligned and oblique LEDs relative to the PCB. A YOLO-based solution for slanted bounding boxes requires a post-processing step to standardize the diode layout for the assessment model. Initially developed for image segmentation of medical images, the U-Net model (Ronneberger et al., 2015) and its variants (Du et al., 2020; Siddique et al., 2021) have been successfully deployed in other fields, e.g., crack detection (Cheng et al., 2018; Hsieh & Tsai, 2020; Liu et al., 2019). Machine learning models like U-Net typically require large training datasets to learn. To train U-Net for diode localization, every diode on the PCBs must be labeled with bounding box locations. This is a highly time-consuming process, as there can be over a hundred diodes on each board and dozens of boards in the training set.

In addition to purely data-driven methods, classical image-processing approaches (Gonzalez & Wood, 2017) provide significant value in developing practical computer-vision solutions. For example, image registration is the task of transforming one image's coordinate system into another. It is necessary for computer vision tasks such as *simultaneous localization and mapping* (SLAM) (Bailey & Durrant-Whyte, 2006; Durrant-Whyte & Bailey, 2006), panoramic stitching (Brown & Lowe, 2007), and image alignment (Szeliski, 2007).

Feature detection is the task of localizing features in an image. A feature can be thought of as a point of interest. Features that let us recognize human faces are eyes, skin, hair, etc. After features are localized, a feature description is required to assign a numerical representation. Good features should not change under perspective or illumination changes and should be well localized. Several open-source or expired-patented algorithms are successful in doing so. SIFT (Lowe, 1999) relies on difference-of-Gaussian for detection and histogram of oriented gradients for description. ORB (Rublee et al., 2011) adds an orientation component to FAST corner detection (Rosten & Drummond, 2006) and modifies (Calonder et al., 2010) to be rotation invariant for description. BRISK (Leutenegger et al., 2011) also modifies FAST by searching for maxima in scale space in addition to the image plane. KAZE (Alcantarilla et al., 2012) detects and describes features in nonlinear scale space. Regarding the computational time to register two images, ORB is the fastest, followed by BRISK, with KAZE and SIFT significantly slower.

The degradation assessment can be treated either as a classification, with two (healthy and degraded) or more states (healthy and few specified discrete levels of degradation), or a regression where the output maps healthy to failed onto, e.g., the 0-1 range. Regardless if the supervised learner is a classifier or a regressor, the dominant approach for practical image-based machine learning is *transfer learning* (Pan & Yang, 2010), which adapts a pre-trained model, e.g., Inception (Szegedy et al., n.d.) to the desired task, using a major deep learning framework, such as Tensorflow (Abadi et al., 2016) with Keras (Chollet, 2017) or PyTorch (Paszke et al., 2019). Some highly-specialized images, e.g., maps, microstructures, or biological data, work better from custom models (Goodfellow et al., 2016).

Computer vision approaches found applications in solving manufacturing problems related to LEDs (Perng et al., 2011), leveraging heuristics to detect manufacturing issues such as mouse bites, missing components, or incorrect packing orientation. More recently, a deep learning model based on CNN layers was proposed for LEDs inspection to detect line blemishes and scratch marks (Lin et al., 2019). To our knowledge, the degradation of LED performance has not been studied.

# System-level approach to introducing machine-learning-based automation

This section describes the system-level development and implementation of computer vision and machine learning solutions for the remanufacturing floor. Specifically, the solution was developed for CoreCentric Solutions, a third-party after-market solutions provider, including returns management, in-warranty, out-of-warranty repairs, full product remanufacturing, and materials recycling. While the solution was developed for the specific remanufacturer, the methodology, components, and aspects should apply to similar problem settings commonly encountered by other organizations.

## System-level description

The system-level approach is depicted in Figure 1. It consisted of image capture, part number identification, and one or more detection algorithms, including LED degradation assessment. Part number identification and detection can operate on the same image, but only sometimes. For example, some PCBs containing LEDs have part number information on the opposite side of the PCB relative to the side which contains LEDs.



Image capture

Part number identification

LED degradation assessment

Detecting unrepairable PCBs

Figure 1 System block diagram

## Image capture

Image captures were mainly based on an industrial camera, as shown in Figure 1. However, images can also be taken by a smart device, a phone, or a tablet. This option is further explored in the section on part number identification.

In the context of the development and deployment of machine vision systems, image captures have multiple purposes: they are inputs for machine learning models described below, and they can also be used as information gathering that can unlock future opportunities. For example, Figure 2a shows the user interface for capturing images with ground truth in the form of the bounding box and failure mode description that the user specifies.

|  |  |
| --- | --- |
|  |  |
| (a) | (b) |

Figure 2: (a) User interface for capturing images with ground truth information on failures (b) Companion user interface for reviewing the annotated image captures

The information is stored in the accompanying data storage structure, which consists of a file with the ground truth table and a folder with the associated images. The ground truth table contains part numbers to allow the user to populate the table quickly and accurately and an extendible file that contains known failure modes, as shown in Table 1. The data in the table allows the reconstruction of the user’s image captures with annotation, as illustrated in Figure 2b, which shows a companion user interface for reviewing image captures. This information can be further edited by domain experts or used by the developers of machine learning solutions.

Table 1: Ground truth information associated with an image

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Image # | Part # | PCB side | File name | Timestamp | x | y | w | h | note |
| 1 | XYZ | Top | XYZ\_2\_Top\_degraded.jpg | 2022-Jan-21 17:03:27 | 1273 | 1575 | 640 | 466 | burned |
| 2 | XYZ | Bottom | XYZ\_2\_Bottom\_OK.jpg | 2022-Jan-21 17:03:56 | -1 | -1 | 0 | 0 | NaN |
| 3 | MNO | Top | MNO\_3\_Top\_degraded.jpg | 2022-Jan-24 09:22:57 | 1394 | 661 | 546 | 660 | crack |

## Part number identification

Part number identification was the second main block in Figure 1. The solution leveraged the Google Cloud Vision API. The block diagram of the system is displayed in Figure 3. It consists of four main blocks: 1) image update, 2) Google AI, 3) comparison of text and existing part numbers, and 4) display and visualization. The blocks are described in turn.

The image update block can either read a still image from the disk (still images) or the camera (live images). Live images can be captured using an industrial camera or smart devices via third-party apps. The system can decimate the captured image to accelerate cloud processing using the Google Cloud Vision API. The API takes the updated image and detects all text boxes it can find. It returns a structure that contains a collection of text strings and the associated bounding boxes in pixels.

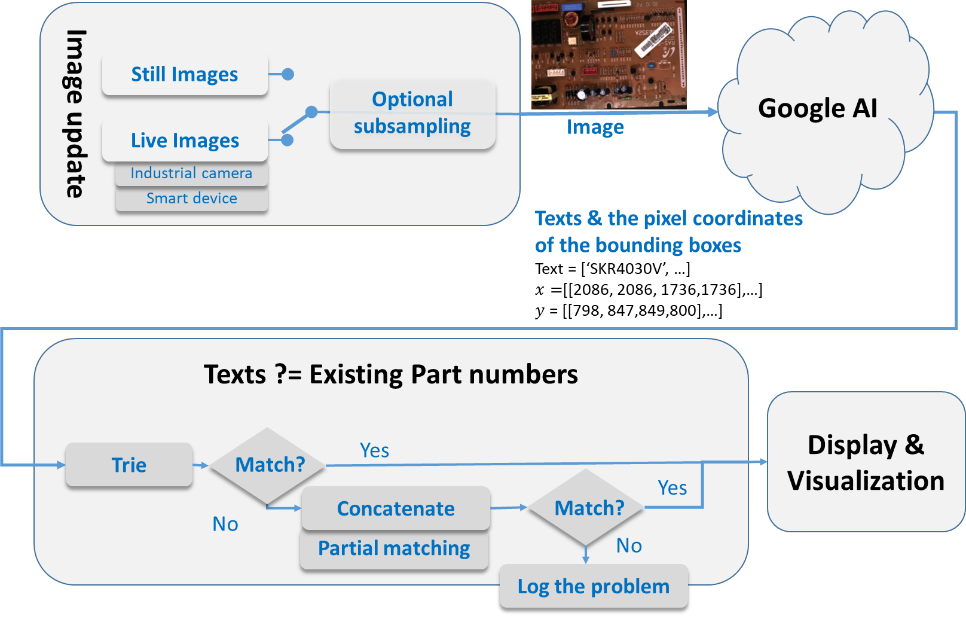


Figure 3 Block diagram of the part number identification algorithm

The next block attempts to find a match for any of the text strings extracted from the image to an existing part number. The processing employs a custom implementation of the Trie data structure. The initial algorithm testing identified two leading causes of possible failure match: 1) on occasion, the Google AI platform breaks the part number into substrings, and 2) sometimes, the Google AI platform misclassifies one or more characters within the part number string. Two solutions were developed to overcome the text identification issues. First, the substrings were concatenated using the information of the associated bounding boxes (they have to be aligned and adjacent). Second, the character misclassification was addressed with partial text string matching. In partial string matching, the existing stored part numbers were compared to text strings extracted from the PCB image by measuring the ratio of character agreements. The ratios and the associated strings were then sorted, and the top matches were shown to the user in descending order. If no suitable match is found, the problem is logged to enable future improvements. The logging consists of capturing the image of the PCB and an optional comment from the operator.

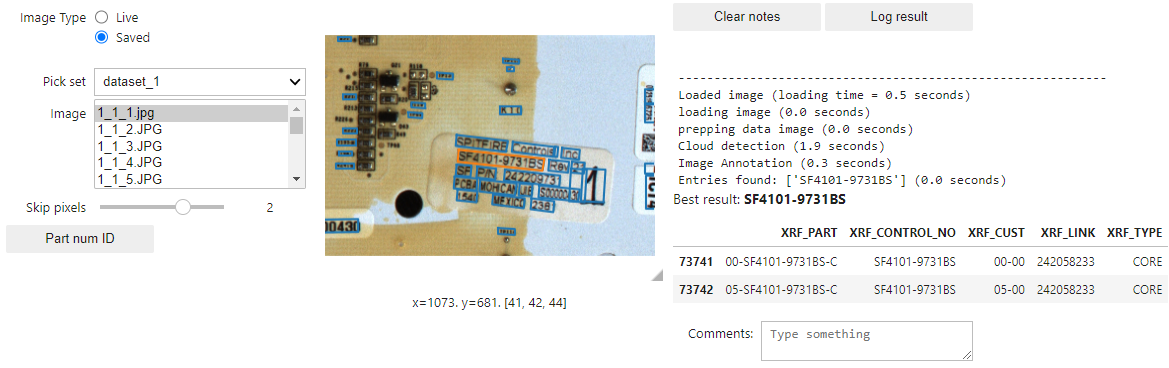


Figure 4 Integrated system of part number identification

The final block serves to display the results. It shows the rows of the reference table with identified parts.

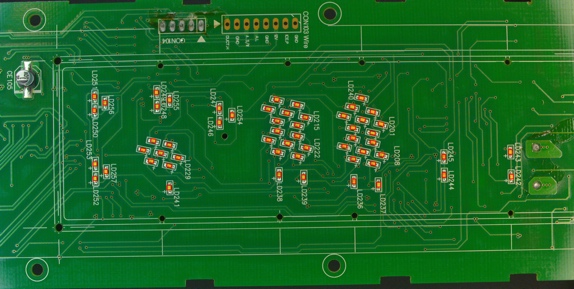
If the part is still undetected, the user can log the event for future troubleshooting by saving the associated image and providing an optional text message.

Figure 4 shows the user interface for the integrated part number identification. The interface can operate in *live mode* for operations on the remanufacturing floor or in *saved mode* for training purposes. The table on the right of the image shows the successfully identified part (the orange rectangle in the image). The interface also allows the user to log results and add notes when encountering problematic edge cases, enabling continuous improvement.

## LED degradation assessment

While machine learning models offer state-of-the-art results for computer vision problems, they typically require a large amount of carefully-labeled data. In the context of localizing electrical components, this entails meticulously drawing bounding boxes for each diode on every image captured of a PCB. Moreover, this must be done for every new PCB scheme requiring remanufacturing. To offset this impractical requirement, both for research and for potential on-site implementation, an automated approximate labeling system was leveraged. This system still requires manual labeling of one “template” board per board type but yields automatic, high-quality bounding boxes for all subsequent boards of the same type. This is accomplished by generating keypoints for the new board and relating them to the keypoints of the original board with a homography. The homography requires a minimum of four pairs of matching points between images and, ideally far more. We used the classical computer vision approach – feature detection and description – to find these points.

By making our localization scheme PCB-specific, the coordinate system of two PCB images can always be related. In other words, we find the transformation that maps the points of a PCB to the corresponding points of another. We model this transformation with a homography. Assuming a pin-hole camera model, two planes captured by a stationary camera can be related by a homography. Enforcing a stationary camera ensures that the two planes share the same projection center. Our capture system enforces a stationary camera, and even if the camera is rotated, two images can still be related. Therefore, no matter how a PCB is placed on the capture bed, whether translated or rotated, we can register its image to another. Specifically, the localization was based on deterministic image processing techniques that require a small manual setup associated with a new part number.



Label Bounding Boxes on Template Board

**One-time Initialization**

**Per Board Type**

Generate Keypoints and Descriptors for Template Board

**Automated Localization Per Board Image**

Generate Keypoints and Descriptors for New Board

Match Keypoints with Template Board

Transform Template Bounding Boxes onto New Board

Standardize Size and Rotation (+/- 180°)

Final Diode

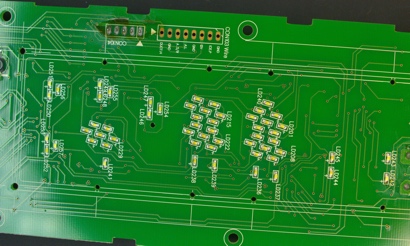
Sub-image

Manual Process

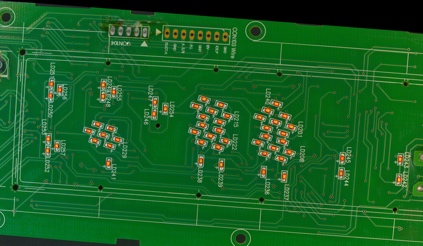
Keypoint Detection

Finalization

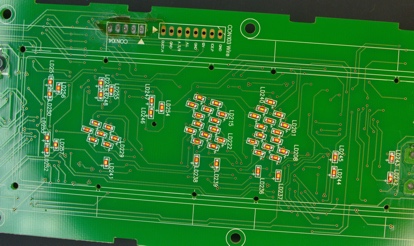
Labeled Template



Input PCB



Registered Template



Input PCB

Bounding boxes

Figure 5 Localization approach

The localization process is depicted in Figure 5. It starts with a one-time initialization that must be repeated for each new PCB design. The initial setup consists of manual labeling of LED bounding boxes on a template PCB, then generation of keypoints for that template PCB. The one-time initialization steps are grouped in the black rectangle on the left side of Figure 5. Once the initialization is complete, the system is ready for online localization, the localization of LEDs on new PCBs of the same type. Online localization starts with keypoint generation and matching, followed by transforming the bounding boxes of LEDs from the template board (see the top row in Figure 5). These steps represent the automated registration system. This system can also be utilized to extract the diode sub-images themselves. The final row represents the steps necessary to extend the automated registration system for this purpose. First, minor adjustments are performed to standardize diode orientation. Then, sub-images associated with individual LEDs are extracted and stored for assessment.

With the automated registration system complete and the dataset approximately labeled, our exploration could move on to machine learning models. With a training dataset consisting of three different PCBs and ten or more duplicate boards for each, training was conducted for U-Net. The results were mixed. The model displayed high bounding-box precision on validation boards but would routinely yield a false positive or false negative on each board. Tuning the confidence parameters was attempted but this effort quickly reached diminishing returns.

Additionally, generalization was quite limited. These mixed results are likely due in part to the limited dataset. As discussed earlier, machine learning models benefit from very large datasets. Due to practical considerations, it can be challenging to gather and image a sufficiently large dataset even with automated labeling. Despite its occasional performance flaws, the U-Net model was able to localize components very quickly. For an application with a larger amount of data and or a small number of unique board types to operate on, this could be the right solution.

Another path investigated localization using YOLO, specifically yolov5. Similarly to U-Net, performance was swift and effective on trained board types, but no meaningful generalization to new board types was observed. Different data splits and data augmentation hyperparameters were attempted, but none led to a general solution. Like U-Net, this model is an excellent solution for evaluating quickly and accurately on a small number of consistent board types. Additionally, with a sufficiently large dataset, it is possible that generalization to diode detection on any board would be possible.

Observing the lack of generalization from YOLO and U-Net, it was decided to examine the possibility of using the automated approximate labeling system for localization. This system generates almost perfect bounding boxes, with no missed or extra predicted diodes, and requires only labeling a single template board for each board type. This labeling was required for the machine learning models as well, as currently they must be trained on each new board type. This solution is not perfect either. For keypoint detectors, performance is dataset dependent. In this dataset, it is important that the algorithms are scale and rotation-invariant. We find that KAZE is most consistent at finding sufficient, high-quality keypoints on all boards in the dataset. However, unlike BRISK and ORB, it is quite slow. This method’s low overhead and simplicity makes it a good choice for general purpose applications and datasets containing many unique board types. Additionally, for datasets where faster keypoint detectors, like BRISK and ORB, are effective, this solution is likely the best one. Figure 6 shows a pseudocode that summarizes localization steps.

**Algorithm 1** Automated Localization (For Multiple Board Images)

**Require:** Board type template exists

template load(board\_type)

detector cv.KAZE.create()

t\_keypoints detector.detectAndCompute(template.image)

matcher cv.Matcher(params)

thresh

**for** images **in** board.images **do**

i.keypoints detector.detectAndCompute(image)

possible\_matches matcher.Match(i.keypoints, t\_keypoints)

valid\_matches []

**for** i.point, t.point **in** possible\_matches **do**

**if** i.point.distance thresh \* t\_point.distance **then**

valid.matches.append(match)

**end for**

homography cv.findHomography(t.valid\_matches, i.valid.matches)

bboxes []

**for** bbox **in** template **do**

new.bbox np.matmul(homography, bbox)

bboxes.append(new.bbox)

**end for**

**end for**

Figure 6 Automated localization pseudocode

LED assessment was based on sub-images of individual LEDs. The initial attempt to develop the model was based on transfer learning of pre-trained models. Inception-ResNet-v2 was chosen for its versatility and accuracy. The weights of this model were frozen, and a single dense layer was attached and trained to predict diode health based on the features extracted. Multiple tests were conducted, with poor performance when evaluated on the validation data. The lack of success suggested that small diode images seem to be too different from the images that Inception-ResNet-v2 had been trained on for naïve transfer learning to be effective. A future investigation into transfer learning for this application could examine the efficacy of partially unfreezing weights in the pre-trained model.



Flattened

512

Activation = Sigmoid

Input image

Degraded

Healthy

Max-Pooling

(22)

Max-Pooling

(22)

Sequence of Conv2D (33), padding = “same”

Activation = ReLU()

Figure 7 Topology of the LED assessment neural network with convolutional layers

The second, more successful approach was based on a custom-based convolutional neural network, trained on examples of sub-images of healthy and degraded LEDs. The general topology was based on standard neural network convolutional models (see, e.g., (Geron, 2019; LeCun et al., 2015)). The first part of the model consisted of two convolutional layers, with kernel size 5, separated by 2x2 max pooling and 25% dropout (Srivastava et al., 2014). After these layers, the output was flattened and sent through three decreasing linear layers, each separated with dropout and a ReLU activation, defined as . Finally, because the model was designed as a binary classification, the sigmoid was applied to the output, leaving a value to be compared to the health threshold chosen for the model. The necessary ground-truth data on the state of degradation was obtained by applying voltage on individual LEDs and observing the potential full or partial loss of brightness. The loss of brightness is not dramatic, the ground truth for assessment was subjective.

|  |
| --- |
| (a) (b) (c) |
| Figure 8 Performance of the LED degradation assessment for a PCB (a) image of the PCB(b) receivers operating characteristic (c) confusion matrix |

A typical performance on the validation data for a specific PCB (Figure 8a) was indicated in the *receiver operating characteristic* (ROC), which plots the estimated detection rate or true positive rate vs. estimated false alarm rate (Figure 8b), and the confusion matrix, which summarizes a specific point on the ROC curve in a tabular form (Figure 8c). The specific ROC point corresponded to the initial requirement that false alarm rate did not to exceed 5%. With this requirement, the model attained the estimated detection rate of 97.5%.

Graphical user interface, website

Description automatically generated

Figure 9 Integrated solution for LED assessment: image loading, localization, and assessment

Figure 9 shows the integrated solution the LED localization and assessment. Because some levels of degradation can be subjective, in addition to localization and assessment, the tool allows a subject-matter expert to review the assessments and provide feedback. The feedback consists of saved images and updated ground truth information, which enables the retraining of machine learning models and paves the path to continuous adaptive learning of the deployed system.

# Conclusions & Recommendations

Machine learning and computer vision provide a compelling path to automate and eliminate tedious, error-prone tasks on the remanufacturing floor. This article illustrated developing solutions for two carefully-selected tasks: part number identification and LED degradation assessment. Introducing machine learning to remanufacturing processes works particularly well when brought incrementally, starting from low-hanging fruit and building to higher-value solutions. Furthermore, solutions that do not cover the full range of input variations often provide sufficient value to be adopted.

Because the first task, part number identification, required solutions for text localization and character classification – both well-researched machine learning problems – we were able to leverage an existing commercial solution available through the Google Cloud Vision API. It is important to emphasize that the commercial solution, while very cost-effective, was not a turn-key solution and needed some data preprocessing and post-processing. For example, down-sampling images of very high resolution accelerated the part number identification. The post-processing steps included parsing the responses from the API and mapping them onto the data from previously registered part numbers. Sometimes detection broke part number strings into sub-strings that had to be concatenated during post-processing. Moreover, occasionally, matching the character strings to the existing part had to be probabilistic because individual characters were misclassified. Because of the large number of different part numbers, it was not practical to address all edge cases. Instead, the integrating solution was equipped with a mechanism to capture and store critical information when a problem arises to enable continuous improvement.

Images of LEDs are not commonly used for training large machine-learning models, and no commercial solutions exist. Moreover, they are sufficiently specialized that even transfer learning did not work well for the degradation assessment. Furthermore, popular deep-learning models for localization (YOLO and U-Net) showed limited generalization. In our experiments, the best solution for degradation assessment employed a CNN model trained from scratch. The most effective localization was primarily based on classical image-processing techniques. However, we combined classical methods and leveraged a more recent, popular segmentation model, U-Net, to accelerate the requisite localization ground truth labeling. The integrated solution for LEDs was equipped with the tools for careful inspection, which enabled the expert to overwrite individual assessments and create data for adaptive learning. This capability was especially useful because the assessment of mild loss of brightness was subjective. The models were developed for three different types of LEDs, but only one was presented in this manuscript. The next development step is to integrate LED degradation detection and localization.

The solution structure presented in this article can be extended to another component failure. The part identification system requires an updated part number database to be repurposed. The automated registration system drastically lowers the manual labor cost of building a relevant dataset. The assessment model can be trained to solve a different component failure problem with a new, labeled dataset. It is important to note: the best methods for localization and assessment will change depending on the dataset. However, this article has demonstrated a modular process flow that can be modified to suit new health assessment tasks. In addition to developing specific solutions, an organization significantly benefits from adopting *machine-learning philosophy* by capturing data associated with critical processes. This data represents an asset that can unlock additional solutions down the road, be used for training, and better characterize the inventory.

# Acknowledgments

The authors gratefully acknowledge our colleagues: Scott Nichols for developing the image capturing fixtures, Neel Surendra Sancheti for capturing the initial ground truth data on LED degradation, and Zane Kitchen Lipski for the initial development of the methods for concatenated bounding boxes.

This material is based upon work supported by the U.S. Department of Energy’s Office of Energy Efficiency and Renewable Energy (EERE) under the Advanced Manufacturing Office Award Number DE-EE0007897 awarded to the REMADE Institute, a division of Sustainable Manufacturing Innovation Alliance Corp. In addition, RIT received funding, in part, from New York State Empire State Development under Grant #AC118 that supported the work on this project.

The views expressed herein do not necessarily represent the views of the U.S. Department of Energy or the United States Government or New York State.

© 2023 Rochester Institute of Technology and CoreCentric Solutions Inc.

# References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., & others. (2016). Tensorflow: A system for large-scale machine learning. *12th Symposium on Operating Systems Design and Implementation (16)*, 265–283.

Alcantarilla, P. F., Bartoli, A., & Davison, A. J. (2012). KAZE features. *European Conference on Computer Vision*, 214–227.

Bailey, T., & Durrant-Whyte, H. (2006). Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics and Automation Magazine*, *13*(3), 108–117. https://doi.org/10.1109/MRA.2006.1678144

Brown, M., & Lowe, D. G. (2007). Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, *74*(1), 59–73. https://doi.org/10.1007/S11263-006-0002-3/METRICS

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). Brief: Binary robust independent elementary features. *European Conference on Computer Vision*, 778–792.

Cheng, J., Xiong, W., Chen, W., Gu, Y., & Li, Y. (2018). Pixel-level crack detection using U-net. *TENCON 2018-2018 IEEE Region 10 Conference*, 462–466.

Chollet, F. (2017). *Deep Learning with Python*.

Diwan, T., Anirudh, G., & Tembhurne, J. V. (2022). Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications*, 1–33.

Du, G., Cao, X., Liang, J., Chen, X., & Zhan, Y. (2020). Medical image segmentation based on u-net: A review. *Journal of Imaging Science and Technology*, *64*, 1–12.

Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localization and mapping: Part I. *IEEE Robotics and Automation Magazine*, *13*(2), 99–108. https://doi.org/10.1109/MRA.2006.1638022

Geron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras & Tensorflow* (2nd ed.). O’Reilly.

Gonzalez, R., & Wood, R. (2017). *Digital Image Processing* (4th ed.). Pearson.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Hsieh, Y.-A., & Tsai, Y. J. (2020). Machine learning for crack detection: Review and model performance comparison. *Journal of Computing in Civil Engineering*, *34*(5), 4020038.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). BRISK: Binary Robust invariant scalable keypoints. *Proceedings of the IEEE International Conference on Computer Vision*, 2548–2555. https://doi.org/10.1109/ICCV.2011.6126542

Li, J., Gu, J., Huang, Z., & Wen, J. (2019). Application Research of Improved YOLO V3 Algorithm in PCB Electronic Component Detection. *Applied Sciences 2019, Vol. 9, Page 3750*, *9*(18), 3750. https://doi.org/10.3390/APP9183750

Lin, H., Li, B., Wang, X., Shu, Y., & Niu, S. (2019). Automated defect inspection of LED chip using deep convolutional neural network. *Journal of Intelligent Manufacturing*, *30*(6), 2525–2534. https://doi.org/10.1007/S10845-018-1415-X/TABLES/6

Liu, Z., Cao, Y., Wang, Y., & Wang, W. (2019). Computer vision-based concrete crack detection using U-net fully convolutional networks. *Automation in Construction*, *104*, 129–139.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, *2*, 1150–1157.

Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury Google, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Xamla, A. K., Yang, E., Devito, Z., Raison Nabla, M., Tejani, A., Chilamkurthy, S., Ai, Q., Steiner, B., … Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, *32*.

Perng, D. B., Liu, H. W., & Chang, C. C. (2011). Automated SMD LED inspection using machine vision. *International Journal of Advanced Manufacturing Technology*, *57*(9–12), 1065–1077. https://doi.org/10.1007/S00170-011-3338-Y/METRICS

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection*. http://pjreddie.com/yolo/

Redmon, J., & Farhadi, A. (2017). *YOLO9000: Better, Faster, Stronger* (pp. 7263–7271). http://pjreddie.com/yolo9000/

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. https://doi.org/10.48550/arxiv.1804.02767

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015* (pp. 234–241). Springer International Publishing.

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. *European Conference on Computer Vision*, 430–443.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *2011 International Conference on Computer Vision*, 2564–2571.

Siddique, N., Paheding, S., Elkin, C. P., & Devabhaktuni, V. (2021). U-net and its variants for medical image segmentation: A review of theory and applications. *Ieee Access*, *9*, 82031–82057.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (n.d.). *Inception-v4, inception-resnet and the impact of residual connections on learning*.

Szeliski, R. (2007). Image Alignment and Stitching: A Tutorial. *Foundations and Trends® in Computer Graphics and Vision*, *2*(1), 1–104. https://doi.org/10.1561/0600000009

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2022). *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. https://doi.org/10.48550/arxiv.2207.02696

Wang, C. Y., Bochkovskiy, A., & Liao, H. Y. M. (2020). Scaled-YOLOv4: Scaling Cross Stage Partial Network. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 13024–13033. https://doi.org/10.48550/arxiv.2011.08036

# About the Authors

**Nicholas Gardner** is pursuing his B.S. and M.S. in Computer Science from Rochester Institute of Technology (RIT) and expects to graduate in 2023. His primary research interest is using machine learning to reduce manual labor and promote reusability.

**Cooper Linsky** is pursuing his B.S. from Rochester Institute of Technology (RIT) and expects to graduate in 2023. His interest lies in applied machine learning research on computer vision, cybersecurity, and knowledge graphs. He is currently researching how knowledge graphs can be used to more effectively model customer information and improve recommender systems.

**Everardo FriasRios** received his B.S. in Computer Engineering Technology from DeVry University in 2014. He joined CoreCentric Solutions Inc. in 2014, where he is currently Engineering Manager. He is responsible for the teams designing and developing remanufacturing procedures for consumer electronics.

**Nenad G. Nenadic** received his B.S. in Electrical Engineering from the University of Novi Sad in 1996 and his MS and Ph.D. in Electrical and Computer Engineering from the University of Rochester (Rochester, NY, USA) in 1998 and 2001, respectively. He has been at the Rochester Institute of Technology since 2005 and is currently a Research Associate Professor. His research interest includes sensors, systems, signal processing, and machine learning.