Towards Deep Computer Vision for In-Line Defect Detection in Polymer Electrolyte Membrane Fuel Cell Materials

Alfred Yan[a], Peter Rupnowski[b], Nalinrat Guba[a], Ambarish Nag[a]

a. Computational Science Center, National Renewable Energy Laboratory, 15013 Denver West Pkwy, Golden, CO 80401, USA
b. Material Science Center, National Renewable Energy Laboratory, 15013 Denver West Pkwy, Golden, CO 80401, USA

ABSTRACT

Polymer Electrolyte Membranes (PEM) fuel cells are a promising source of alternative energy. However, their production is limited by a lack of well-established methods for quality control of their constituent materials like the electrode and PEM during roll-to-roll manufacturing. One potential solution is the implementation of deep learning methods to detect unwanted defects through their detection in scanned images. We explore the detection of defects like scratches, pinholes, and scuffs in a sample dataset of PEM optical images using two deep learning algorithms: Patch Distribution Modeling (PaDiM) for unsupervised anomaly detection and Faster-RCNN for supervised object detection. Both methods achieve scores on performance metrics (ROC-AUC and PRO-AUC for PaDiM and AP for Faster-RCNN) that are comparable to their scores on benchmark datasets and show potential for localizing relevant defects of interest. Overall, deep learning methods show promise at detecting defects and has the potential to achieve real-time defect detection.

Keywords: anomaly detection, PEM, object detection, pinhole, defect detection, electrode

1. INTRODUCTION

The rising threat of climate change and growing worldwide energy demand necessitates the development of clean energy technologies that have a minimal carbon footprint and environmental impact. Polymer Exchange Membrane (PEM) fuel cells constitute a promising example of such a clean energy technology with the potential for a wide range of applications. However, one barrier to their deployment is the need for quality control methods for relevant materials during manufacturing, such as the PEM, electrode, and gas diffusion layer. [1] In particular, the existence of harmful defects, which include pinholes[2][3][4][5][6], cracks[7][8][9][10], bubbles[11][12][13], delaminations [14][15][16], and coating/thickness irregularities[12][17][18] is well-documented. Research on such defects has included a mix of investigating their impacts on material performance, their underlying causes [19][20][21][14][15][22], as well as methods for detecting and localizing them. It is particularly desired to develop in-line detection methods for roll-to-roll manufacturing, which can non-invasively detect defects in 'real-time', or quickly enough as to be almost instantaneous and thus allowing the processing of large volumes of material. Previous work in this area mostly involves methods including infrared (IR) thermography[23][24] [25][26], pressure drop measurements[24], and optical inspection through a camera feed. IR thermography methods utilize a gas that reacts with the material when there is a defect, and has been used to detect defects like pinholes[26] and bare spots on the gas diffusion electrode catalyst layer [23][25]. For example, Ulsh et al. [26] have detected artificially-induced pinholes in membrane-electrode assemblies (MEAs) through pulsing hydrogen-containing gas at the membrane.

This study focuses on optical inspection of PEM materials. Optical inspection includes X-ray fluorescence, which has demonstrated potential to monitor the chemical composition and film thickness of the catalyst layer [25][27]. However, such methods are not under serious investigation for its slow acquisition times, and this was one of the factors driving research into IR thermography. More recently, optical methods include multispectral imaging, in which incident light generates a measured thermal response from a material. This has applications in thickness mapping of Li-ion battery electrodes [28] and

for general purpose membrane inspection[29][30][31]. Another type of optical defect detection involves the rolling of the material across a camera feed, so that images of the membrane can be processed by computer vision (CV) algorithms that can locate and identify defects [32][33]. Such optical methods have been investigated for the detection of defects in PEM materials, many of which involve pre-determining a set of defects in the optical image, and the application of CV algorithms like intensity thresholding to isolate anomalously bright or dark spots in the image [34][35][36][37]. For example, in Rupnowski et al.[35], defects are artificially induced to two sample membranes: one membrane with scratches and scuffs, and the other with debris sprinkled onto the surface. CV algorithms based on binary thresholding show potential at identifying sprinkled debris but are not investigated for detecting scratches and scuffs due to their complex morphology. In Johnson [36], a thresholding algorithm is used to find potential objects of interest on a membrane, and a neural network is used to classify the objects based on their characteristics like shape and brightness. Another method used on polymer films by Tolba et al. [38] calculates the similarity of each region of an image on a rectangular grid with a defect-free baseline, where similarity is calculated by comparing the luminance and contrast between the image region and baseline as well as their covariances. While high scores are obtained using this method, a potential flaw is that its performance may be sensitive to the selection of the baseline image. An assumption made is that one small sub-image can entirely describe a defect-free membrane, when there may be high variance in the appearance of a defect-free image.

The study by Johnson marks the beginning of a trend of applying machine learning for optical defect detection methods. However, the machine learning methods used still leave open challenges: in the study by Johnson, careful feature selection is required before the neural network can be trained, which may be cumbersome, and the neural network can only classify defects that can be detected by a binary thresholding algorithm. Challenges also exist in the work by Rupnowski et al. [35], in that the binary thresholding algorithms were not successful in detecting some faint artificial slits and scuffs. Overall, optical methods that depend on manually designed CV algorithms may be unable to generalize to unforeseen defects. For example, an algorithm trained to detect defects by filtering regions of interest by size, shape, or brightness may miss other defects that do not fit the parameters specified in the CV algorithms.

With this problem, the task of optical defect detection lends itself to the possibility of applying more advanced deep learning methods. Machine learning already has many applications in PEM fuel cells [39] ranging from on-line fault diagnosis[40][41][42][43][44][45], aiding in fuel cell design [46][47][48], and predicting optimal operating conditions [49][50][51]. However, to the authors' knowledge, there have been no studies using machine learning, besides that by Johnson, centered on detecting defects from membrane images during the roll-to-roll manufacturing process. There is, nevertheless, a variety of studies investigating deep learning for defect detection in other materials that serve as inspiration [52][53][54][55][56][57][58]. Unsupervised anomaly detection is a deep learning framework that is well applied in this area. Unsupervised anomaly detection models only require baseline anomaly-free images for training and a relatively small test dataset of defective and defect-free images for selecting an anomaly threshold. After training, models have the ability highlight any anomalous pixels or regions in an image. Without feature selection, there is great potential to generalize to unforeseen defects. The amount of image preprocessing is minimized, as the model can account for much of the noise and irregularities that may appear in the baseline image. Thus, anomaly detection methods have significant defect detection applications, having been applied in detecting defects in a variety of objects like textiles and metals [59]. An example of an anomaly detection algorithm is Patch Distribution Modeling (PaDiM) developed by Defard et al. [60]. PaDiM is evaluated on benchmark datasets including the MVTec-AD dataset, and achieves superior performance compared to previous existing methods as evidenced by its high AUC-ROC and PRO-AUC scores [60]. Another type of deep learning model is the object detection model. In object detection, a dataset of images labeled with bounding boxes identifying certain objects in the images is used to train a model to detect and identify the same objects in new images. Object detection is a popular machine learning problem, with many different model architectures created like Faster-RCNN [61] and YOLO [62]. Object detection is also particularly appealing because of existing methods for 'few-

shot' detection, where a small dataset showing very few instances of a particular object is used to train a network. In this study, PaDiM and a Faster-RNN object detection model are trained and tested on a small dataset of PEM optical images, and their relative advantages and disadvantages are discussed.

2. METHODS

The following section will separately outline the procedure used to train PaDiM for anomaly detection and Faster-RCNN for object detection.

2.1 Detection of defects as anomalies using PaDiM

The full procedure for PaDiM, outlined in Defard et al. [60], is summarized as follows: each defect-free image in the training dataset is input into an image classification convolutional neural network (CNN) pretrained on ImageNet [63] and embedding vectors for each image are created by extracting activation vectors from different layers of the CNN. In the implementation of PaDiM used in this study, the pretrained model used was EfficientNet-B4, a CNN developed by Tan et al. [64][65]. Activation vectors were extracted from the 3^{rd}, 7^{th}, and 17^{th} blocks, which allowed various semantic levels to contribute to the feature vector. After extracting features, a $W \times H$ (the largest resolution of the activation maps that features are extracted from) map of embedding vectors is created for each image, where each embedding vector at position $(i,j) \in [1, W] \times [1,H]$ on the map contains only values in the extracted features that are spatially connected to position (i,j). Finally, a vector mean and covariance is calculated for the vectors in each position $(i,j) \in [1, W] \times [1,H]$ over all the training images, to create an $W \times H$ map of vector normal distributions. In the inference step, embedding maps are created using the same method for test images, and for each position (i,j) on the map, an anomaly score is assigned by calculating the Mahalanobis distance [66] between the embedding and the learned normal distribution. Finally, the $W \times H$ map is resized to the original image resolution to yield a pixel-level anomaly map for the original test image. To segment anomalies, pixels with values in the anomaly map higher than a chosen anomaly threshold are highlighted. In our implementation of PaDiM, 2 GPUs were used to parallelize the training and testing processes.

2.1.1 Detection and Localization of Scratches and Scruffs

The first image dataset investigated in this study featured scratches and scuffs that were artificially induced on some electrodes. To create the scratches/scuffs dataset, four 8×11'' fuel cell electrode specimens were used, which were originally featured in Rupnowski et al. [35]. Two of these specimens contained artificially induced defects in the forms of scratches and scuffs arranged in a 6x3 grid, of which one had defects induced prior to tacking a polymer electrolyte membrane, and the other had defects induced after. Two specimens were identified as defect-free baselines, of which one had a polymer electrolyte membrane, and the other did not. A large high-resolution image was taken of each membrane using reflectance mapping [35], and labels around the artificially induced defects were manually drawn.

These images were investigated at different resolutions. To create input images to the model, each membrane image was split into a $6c \times 3c$ grid of smaller sub-images, where c was an integer constant, and each sub-image was resized to 224×224 pixels. Three values of c (1,3,5) were tested. As c increased, each sub-image was shrunk less relative to its original size during resizing, and thus image resolution increased.

For c=1, each of the four membrane specimens was each split into a 6x3 grid of sub-images. This yielded 36 images containing defects and 36 defect-free images. Because the defects were originally created in a grid on the membrane, each defective image contained 1 defect with little overlap. Two different procedures were used to organize the images into training and test data and evaluate the model. The first, which will be referred to as the "standard" procedure, followed Defard et al. [60]. The model was trained on the defect-free training data, and then evaluated on the test dataset containing both defective and defect-free images to yield final accuracy scores. 18 of the total 36 defect-free images were

chosen randomly to create the training set, and the remaining 18 were used as test data. To combat class imbalance arising from the different number of defect-free images in the test and training data, each of the 18 defect-free test images was duplicated once to create 36 defect-free test images in total. Finally, after all images were reduced from size 1359-1365 pixels × 2041-2124 pixels to size 224 pixels × 224 pixels to be processed by the PaDiM model, each image in the training dataset was augmented through rotations of 90, 180, and 270 degrees clockwise, as well as random flips (horizontal, vertical, and both horizontal and vertical). Finally, the model is trained on the defect-free training images and evaluated on the test images using the average ROC-AUC, PRO-AUC score, and threshold PRO. The PRO-AUC is the area under the per-region overlap curve, which is described in Bergmann et al. [67] and Shi et al. [68]. The threshold PRO is the average per-region overlap ranging from 0 to 1 calculated by segmenting anomaly-containing pixels using a chosen threshold, which is described in Bergmann et al. [59]. In this study, the chosen threshold was one that maximized the F1 score, which is defined in Lipton et al. [69] as follows:

$$F1 = \frac{2 * tp}{2 * tp + fp + fn}$$

Here, *tp* is the number of true positives, *fp* is the number of false positives, and *fn* is the number of false negatives. To address potential instability resulting from the small dataset sizes in this study, 36 repeated trials were taken with different random samples of 18 defect-free images from the original set of 36 sub-images to use as training data, and the average of the scores were reported at a 95% confidence interval. These trials were conducted once without reducing the embedding vector size and once where the embedding vector size was reduced through randomly selecting 100 features, as performed in Defard et al. [60].

The standard evaluation method closely resembles the evaluation method used by Defard et al. [60] and allows for direct comparison with the current study. However, the sizes of the test and training datasets are smaller than those seen in the benchmark MVTec-AD dataset [59], where there can be hundreds of different training images and more than a hundred test images for one object class. In contrast, there are 18 unique images to be used for training and 54 unique images used for testing. Thus, it is worth investigating whether the small dataset sizes prevent the final model and anomaly thresholds from generalizing to defects outside the test dataset. This can be done by quantifying the variance in model performance across the different images in the test set. To do this, another procedure was used to evaluate the models, to be referred to as leave-one-out cross-validation (LOOCV). 36 runs were repeated like in the standard procedure with c=1, except in each run, a different defective image was removed from the rest of the images. Then, all the 53 remaining images were used to train and evaluate the model to obtain accuracy scores and select a pixel-level anomaly threshold, just like in the standard procedure. This process will be referred to as the "inner loop." The calculated accuracy metrics were the ROC-AUC, PRO-AUC, and threshold PRO. Afterwards, the threshold PRO was evaluated on the removed image alone, using the threshold and trained model obtained in the inner loop, a process which will be referred to as the "outer loop." This is repeated 36 times, each time with a different defective image taken out to be evaluated in the outer loop. Finally, the average of all 36 outer loop threshold PRO scores is calculated at a 95% confidence interval and compared to the average threshold PRO scores obtained on the inner loop to quantify the model's performance degradation. Different random train/test splits of the defect-free membrane sub-images were used in the 36 repeated runs. This procedure was also tested without dimensionality reduction as well as with 100 randomly selected features. A schematic of this cross-validation procedure is shown in Figure 1.

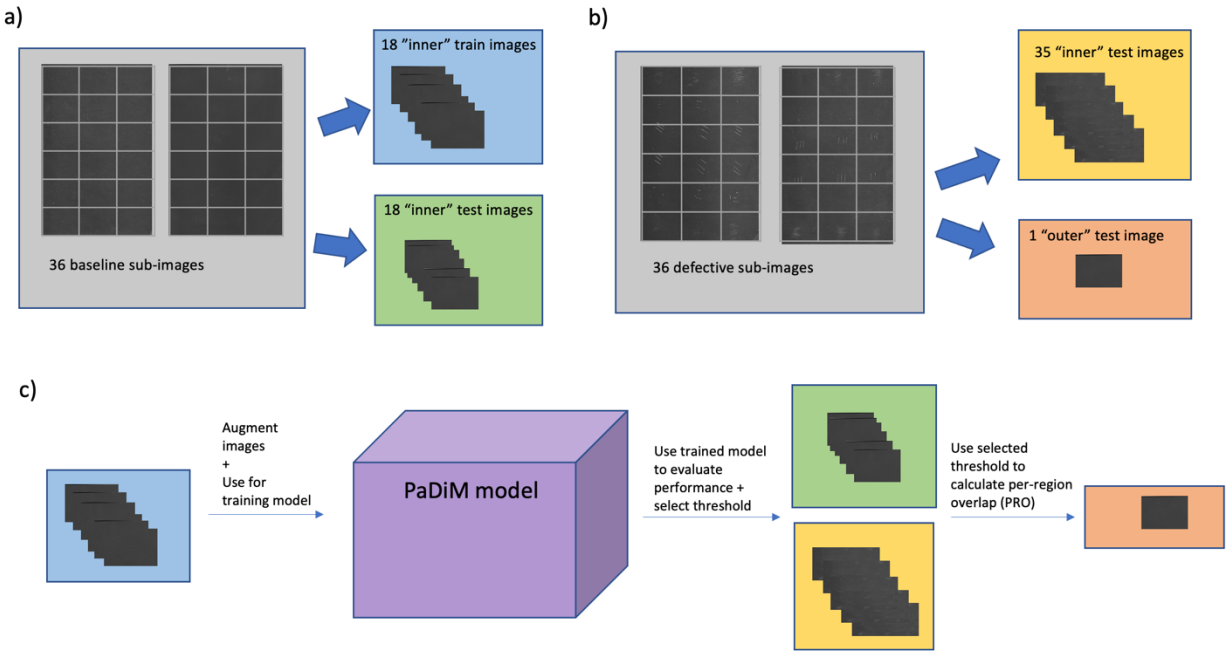


Figure 1: A schematic of the leave-one-out cross-validation (LOOCV) procedure is displayed. In each run, all the sub-images are split into train and test datasets: the baseline sub-images are split into train and test images (1a), and one defective sub-image is held out from the rest of the defective images (1b). In 1c, the model is trained on the baseline images and tested on some baseline and defective sub-images, which are also used to choose an anomaly threshold by optimizing the F1 score. Finally, the model and chosen threshold are tested on the held-out image by evaluating the per-region overlap (PRO) score on the held-out image. This process is repeated 36 times, each time with a different defective sub-image held out.

For c=3 and c=5, the resolution was higher and more sub-images were created from the larger membrane images. Thus, all the sub-images created from splitting the defect-free baseline membranes into a 6c × 3c grid were used for training data, and all the sub-images created from splitting the defect-containing membranes were used for test data. Data augmentation through rotating and flipping the images was not used because a larger number of sub-images was available, close to the number of images per class in the MVTec-AD benchmark dataset [59]. Only the standard procedure was used (no LOOCV).

2.1.2 Detection and Localization of Pinholes

PaDiM was also investigated in its ability to detect and localize pinholes on the membrane. The pinhole dataset was created from 4 8 × 11" electrode samples, of which one was described in Rupnowski et al. [35], each containing 9 artificial pinholes. The smallest pinholes were approximately 150 µm. These images were also divided into a grid of sub-images, except instead of dividing a whole membrane image into a grid of images with a preset dimension (e.g., 6c × 3c), the membrane image was kept at its original resolution and divided into a large grid of 224 × 224 pixel sub-images, with no resizing, after each dimension in each membrane was slightly cropped to become a multiple of 224. This was because a far larger resolution was required for the small pinholes to become visible. 39 sub-images containing pinholes were then manually isolated and segmented to create a test dataset. There were more images than the number of pinholes because some pinholes were located on the boundary between two sub-images, causing two images to contain the same pinhole. 36 random images from the set of pinhole-free images were also added to the test set, so that the total size of the test set was 75. Both the standard and LOOCV evaluation method were used, each once with no dimensionality reduction, and another with 100 randomly selected features. In the standard procedure, 20 runs were repeated. Data augmentation on the training data was not used because of its larger size.

2.2. Detection of defects and objects using Faster-RCNN

Object detection models with the Faster-RCNN [61] architecture were trained to detect defects as well. While anomaly detection methods many already be tailored for detecting defects, object detection methods may be suitable for when defects need to be classified. The methods used in this study are particularly inspired by recent advances in few-shot object detection. Few-shot object detection is a framework where models are optimized to detect objects after being trained on very few images. Some recent work on few-shot detection includes a study by Wang et al. [70], in which a "two stage fine-tuning" approach (TFA) is developed. In this two-stage approach, a Faster-RCNN model is first trained on a large base dataset and then fine-tuned on a smaller novel dataset. The base dataset contains many different object classes, each of which contains many images. The novel dataset contains all the object classes in the base dataset as well as some new classes, but there are only K examples of each class in the novel dataset, where K is a very small number like 1,5,7, etc. This method, while illuminating some effective strategies for learning from small datasets, cannot be optimally implemented in this study because it requires a large base dataset during the first stage of training, which should ideally be similar to the novel dataset. However, aside from the small hand-labeled datasets of scratches, scuffs, and pinholes used in the PaDiM study, there is no obvious candidate for a closely related base dataset that is well-labeled. Thus, it is not immediately obvious how to exactly replicate the few-shot object detection method by Wang et al. [70]. Nevertheless, we can borrow some insights by using a more rudimentary single-stage fine-tuning approach. By using a technique that is similar to but less optimal than TFA, the results from this study can establish a baseline performance for object detection in PEMs that can be expected to improve as more data becomes available to fully implement TFA.

To train a model to detect defects, a rudimentary fine-tuning approach inspired by TFA is used. A model with the Faster-RCNN [61] architecture pre-trained on a section of the PascalVOC benchmark dataset is taken from an online model repository [70]. The ResNet backbone, region proposal network, and feature extractor are frozen according to TFA, and only the box classifier and regressor are given randomized weights and then trained on a novel dataset that contained only images of defects. Through this process, knowledge from the PascalVOC training dataset could be used for transfer learning to improve prediction performance on detecting defects. A cosine similarity-based box classifier with $\alpha = 20$ was used [70]. In this study, two novel datasets were used: the pinholes and scratches/scuffs. Model architectures and training hyperparameters were the same as those used to train on the Pascal-VOC novel classes in Wang et al. [70], which can be found in their publicly available code repository [70]. During testing, 1 GPU was used to parallelize the inference step.

2.2.1 *Detection of Pinholes*

To create an image dataset from the sample images containing pinholes, a 66 × 66 rectangular boundary was drawn around each pinhole, and the image inside each boundary was resized to 800 × 800, which is the input resolution for Faster-RCNN, and then added to the training dataset. Box labels were then manually drawn around the pinhole in each image. To acquire baseline images, each sample was divided into a grid of 66 × 66 sub-images. Gaussian adaptive thresholding was used to extract baseline sub-images that contained bright spots and discard the rest. Each baseline sub-image was also resized to size 800 × 800, and sub-images containing a pinhole were discarded. It should be noted that the drastic resizing was done to make the pinhole appear larger in each sub-image, as object detection models are known to perform worse at detecting small objects [71]. Finally, to perform 5-shot and 10-shot learning, either 5 or 10 pinhole images were randomly chosen from the 36 total pinhole images to be used as training images, the rest were used for testing, and 36 baseline images that contained bright spots were also randomly selected to be added to the test dataset. For each shot (5 or 10), stable results were obtained by repeating runs 30 times with different random seeds. At each new random seed, a new random sample of 5 or 10 images was taken from the total set of defective images to serve as the training data.

2.2.2 Detection of Scratches and Scuffs

For the scratches/scuffs sample images, the same procedure was used with a few modifications. A lower resolution was used: each defect-containing membrane was split into a 6 × 3 grid of sub-images, such that approximately each sub-image contained 1 defect, and each defect was manually labeled. In the original study [35], 12 scuffs and 24 scratches were reported to be created on the membrane, and ideally there should be the same number of images containing scratches and scuffs. However, some images were labeled with both, which will be explained in the Analysis section of this paper. One of the baseline membranes (membrane B2 in Rupnowski et al. [35]) was split into 18 sub-images as well, to be added to the test set in each run. Only 5-shot learning was used for the scratches and scuffs because of the lower data availability.

3. RESULTS

For PaDiM, various performance metrics obtained from implementing the standard procedure are displayed for both the scratches/scuffs dataset in Table 1 and the pinhole dataset in Table 2. All margins of error are rounded up to the nearest hundredth. For all the accuracy metrics, averages were rounded down to the nearest hundredth, while the margins of error were rounded up to the nearest hundredth. For c=3,5, time metrics were rounded to the nearest hundredth and other performance metrics were rounded down to the next hundredth. The pixel-level ROC-AUC score summarizes the model's ability to localize defects (segment pixels that are part of a defect), and the image-level ROC-AUC score describes the model's ability to detect them (classify whether an image contains a defect overall) [60]. Additionally, stages of the testing process were timed to investigate the feasibility of real-time in-line quality inspection, and the results are shown in Tables 1 and 2. For these time efficiency metrics, the confidence intervals were rounded up to the nearest hundredth and the averages were rounded to the nearest hundredth (up or down). The total inference time is the amount of time elapsed during the inference step on all the test images, which is when Mahalanobis distances are calculated for all the patches' feature vectors in all the test images [60]. The total testing processing time describes the time elapsed from when features start being extracted for the test images to when the inference step finishes. This metric most realistically represents the total time it takes for the model to make a prediction on an image, starting from when the images are input and ending when the model outputs an anomaly map for each image. The inference times in this study are shorter than those reported in the original PaDiM study due to our parallel GPU implementation. For example, the original study reports an inference time of 0.23 seconds per image using a CPU, while for the pinhole dataset, the inference time is on average 0.42 seconds for 75 images, or around .01 second/image.

Performance metrics obtained from implementing the LOOCV procedure are displayed in Table 3 for the scratches/scuffs dataset and in Table 4 for the pinholes, and runtimes for the LOOCV procedure were not evaluated to avoid redundancy.

Features	c	Image ROC-AUC	Pixel ROC-AUC	PRO-AUC score	Threshold PRO score	Inference time (s)	Total Processing time (s)
100	1	0.9 ± 0.02	0.95 ± 0.01	0.82 ± 0.01	0.47 ± 0.01	0.42 ± 0.01	1.09 ± 0.02
248	1	0.9 ± 0.01	0.95 ± 0.01	0.83 ± 0.01	0.48 ± 0.01	0.62 ± 0.02	1.5 ± 0.06
100	3	0.85	0.94	0.84	0.53	0.96	3.99

Features	c	Image ROC-AUC	Pixel ROC-AUC	PRO-AUC score	Threshold PRO score	Inference time (s)	Total Processing time (s)
248	3	0.82	0.94	0.84	0.53	1.32	4.51
100	5	0.81	0.95	0.86	0.52	2.1	10.78
248	5	0.79	0.95	0.86	0.52	2.74	11.27

Table 1: Performance metrics using PaDiM for anomaly segmentation are recorded on the scratches and scuffs dataset, using the standard evaluation procedure. Values are rounded to the nearest hundredth. For c=1, metrics are averaged over 36 repeated trials and are displayed at a 95% confidence interval. For c=3 and c=5, only 1 run is taken because the dataset sizes were larger, so there were no confidence intervals recorded.

Features	Image ROC-AUC	Pixel ROC-AUC	PRO-AUC score	Threshold PRO score	Inference time (s)	Total Processing Time (s)
100	0.99 ± 0.01	0.99 ± 0.0	0.99 ± 0.0	0.8 ± 0.01	0.42 ± 0.01	1.37 ± 0.11
248	0.99 ± 0.01	0.99 ± 0.0	0.99 ± 0.01	0.81 ± 0.01	0.64 ± 0.02	1.64 ± 0.1

Table 2: Performance metrics using PaDiM for anomaly segmentation are recorded on the pinhole dataset using the standard evaluation procedure. 20 runs were repeated.

Features	Inner Image ROC-AUC	Inner Pixel ROC-AUC	Inner PRO-AUC score	Inner threshold PRO	Mean outer threshold PRO
100	0.89 ± 0.02	0.95 ± 0.01	0.83 ± 0.01	0.46 ± 0.01	0.63 ± 0.12
248	0.9 ± 0.01	0.95 ± 0.01	0.83 ± 0.01	0.48 ± 0.01	0.65 ± 0.11

Table 3: Performance metrics using PaDiM for anomaly segmentation are recorded on the scratches and scuffs dataset using the LOOCV procedure. Scores are averaged over 36 images at a 95% confidence interval.

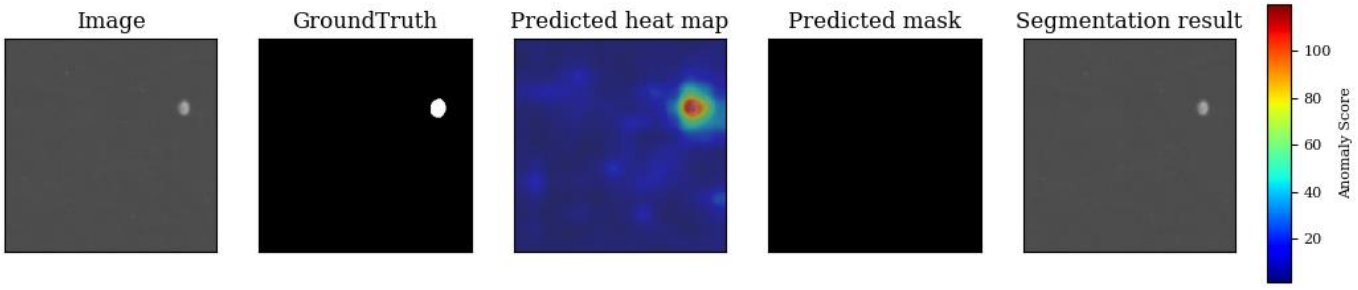
Features	Inner Image ROC-AUC	Inner Pixel ROC-AUC	Inner PRO-AUC score	Inner threshold PRO	Mean outer threshold PRO
100	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.8 ± 0.01	0.8 ± 0.09

Features	Inner Image ROC-AUC	Inner Pixel ROC-AUC	Inner PRO-AUC score	Inner threshold PRO	Mean outer threshold PRO
248	0.99 ± 0.01	0.99 ± 0.01	0.99 ± 0.01	0.81 ± 0.01	0.8 ± 0.08

Table 4: Performance metrics using PaDiM for anomaly segmentation are recorded on the pinhole dataset using the LOOCV procedure. Scores are averaged over 39 images at a 95% confidence interval.

The model performs the best on the pinhole dataset. The high image-level ROC-AUC scores indicate that the model can effectively distinguish whether an image contains a pinhole. However, the high pixel-level ROC-AUC scores are difficult to interpret because very few pixels belong to a pinhole compared to the number of pixels belonging to the baseline, leading to class imbalance in the pixel labels. The PRO-AUC accounts for the imbalance by ignoring areas that are not labeled as defective [59], but the PRO-AUC scores are very high as well. The only scores that are not saturated (very close to 1.0) are the threshold PRO scores. The outer threshold PRO scores from LOOCV are close to the threshold PRO scores from the standard procedure and inner LOOCV loop. This suggests that the model generalizes well when localizing new pinholes. However, as evidenced by the large confidence interval in the outer threshold PRO, there can be high variance in model performance when making new predictions. Inspecting the model pixel predictions on the pinhole images evaluated in the outer loop (with 100 randomly selected features), one pinhole is completely missed. This pinhole and the pinhole with the smallest nonzero relative area segmented are shown in Figure 2a and 2b. They have visibly plain appearances, without the concentric circles characteristic of other pinholes [35], which may cause the model to overlook them. While visual inspection of their anomaly heat maps seems to indicate that the anomaly scores of their pixels are above average, their scores are simply not high enough to surpass the chosen anomaly threshold. The distribution of various threshold PRO scores for different pinhole images obtained during the cross-validation outer loop is shown in a histogram plot in Figure 3.

a)



b)

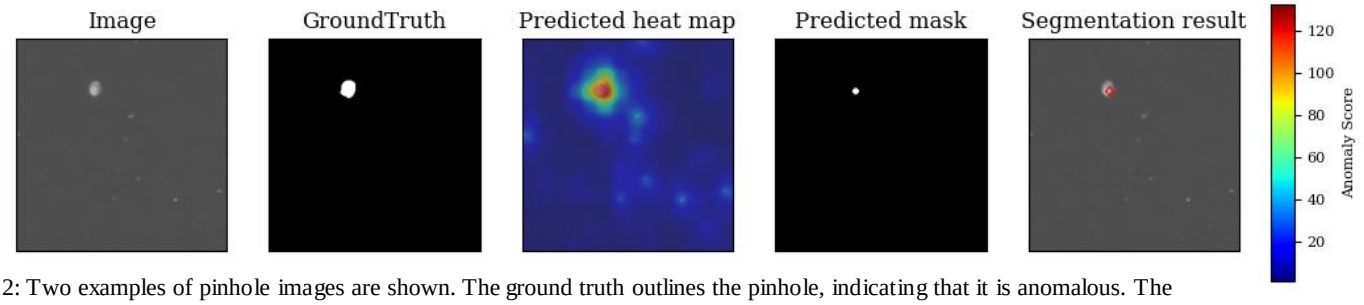


Figure 2: Two examples of pinhole images are shown. The ground truth outlines the pinhole, indicating that it is anomalous. The predicted heat map shows a heat map of anomaly scores output by the model, where red indicates high anomaly scores. The

segmentation result is created by circling regions that have predicted anomaly scores greater than a chosen anomaly threshold. Here, the anomaly threshold is chosen by maximizing the pixel classification F1 score. 2a) shows a pinhole that was completely missed by the algorithm, although its heatmap showed relatively high anomaly scores. 2b) shows another pinhole. The model recognizes the pinhole as anomalous, but only 16% of its pixels have a score higher than the anomaly threshold.

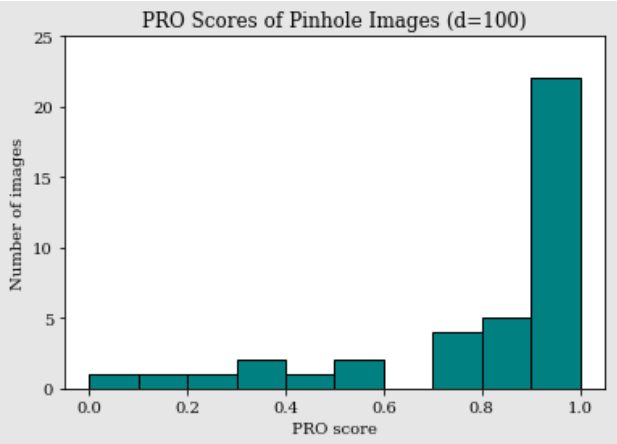


Figure 3: Histogram distribution of PRO scores for each image containing pinholes. 1 pinhole image had a value of 0, but the rest obtain a nonzero PRO. Scores are obtained in trials where 100 random features were selected (d=100).

Inspecting the results from a separate standard run with dimensionality reduction, there are three false positives in the baseline pinhole image predictions. Some seem to be from debris like lint, which still may be worth detecting. These false positives are shown in Figure 4. There are no other false positives in that one run specifically, so the model appears to have good specificity.

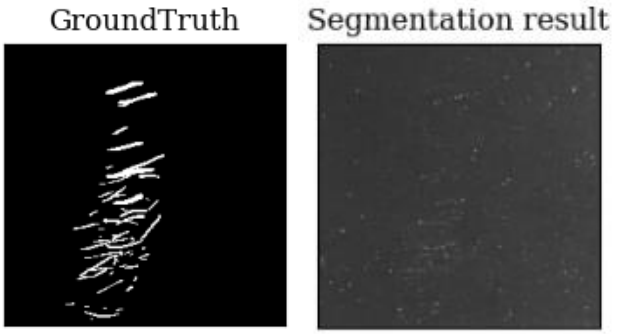
Figure 4: False positives predicted by PaDiM on some baseline images from the pinhole dataset. The first appears to be from a small bright spot, and the second two appear to be from pieces of foreign debris. These are the only false positives that are predicted among all the 75 images.

Model performance on the scratches and scuffs dataset is weaker. Inspecting results from the LOOCV procedure with dimensionality reduction, some low-contrast defects that were anticipated in Rupnowski et al. [35] to be challenging to detect were indeed missed by the model. However, as the resolution increases, the low-contrast defects become more conspicuous and can be localized by the model. For example, three defects were not localized during LOOCV, but were localized in the runs with c=3. These defects are displayed at low resolution (c=1) in Figure 5, and at higher resolution (c=3) in Figure 6.

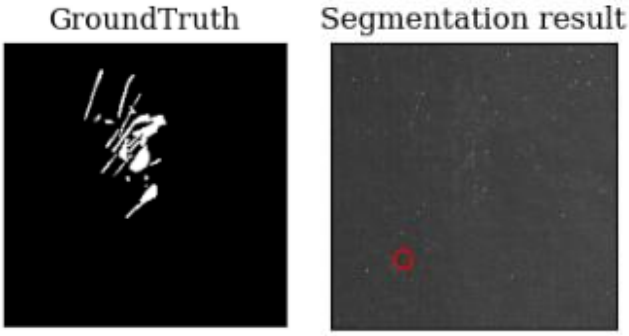
While higher resolution allows faint defects are more easily segmented, it also causes the model to be more susceptible to segmenting false positives. Small bright spots are ubiquitous in the sample images, and as resolution increases, they become more visible and are more likely seen as defects by the model. Other false positives are large and localizable at all resolutions and may represent significant defects, but their true nature is unknown. For example, a triangular section of the electrode has a lighter tone from the rest of the sample, which might be from a fold in the sample causing the section to have a

higher elevation during image scanning. The real reason for the discoloration is unknown, and whether it represents a truly harmful defect could be subject to further investigation. This triangular discoloration is displayed in Figure 7 at varying resolutions, as well as another low resolution (c=1) image containing scratches that also contains false positives.

a)



b)



e)

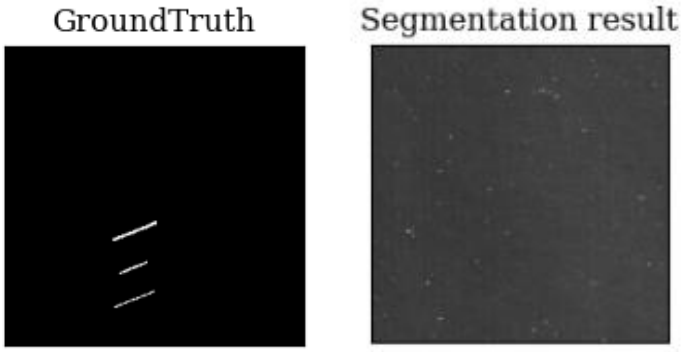
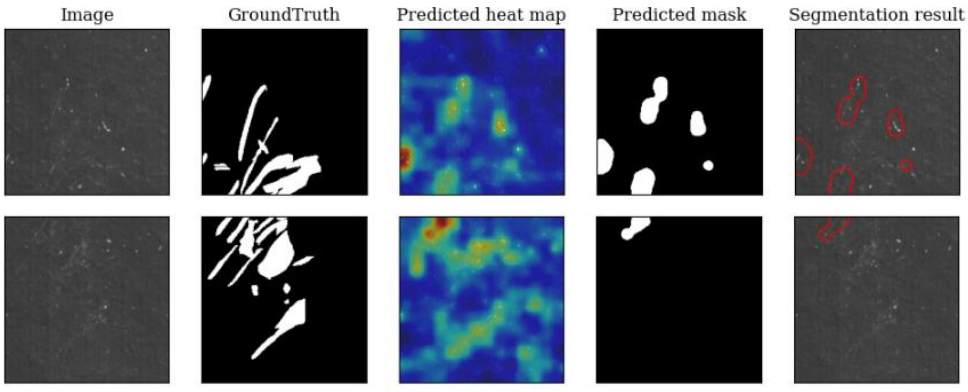
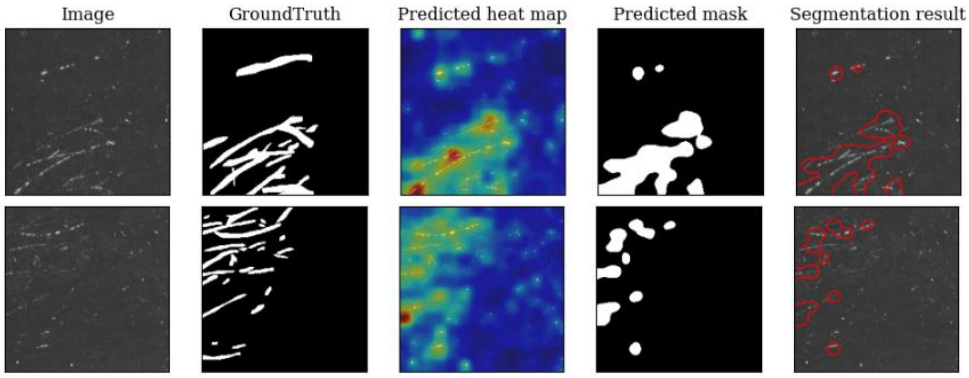


Figure 5: 5a) to 5c) show the 3 different defects that were completely unsegmented by the algorithm. The defects may be difficult to spot by eye in this diagram, due to the low image resolution (c=1).

a)



b)



c)

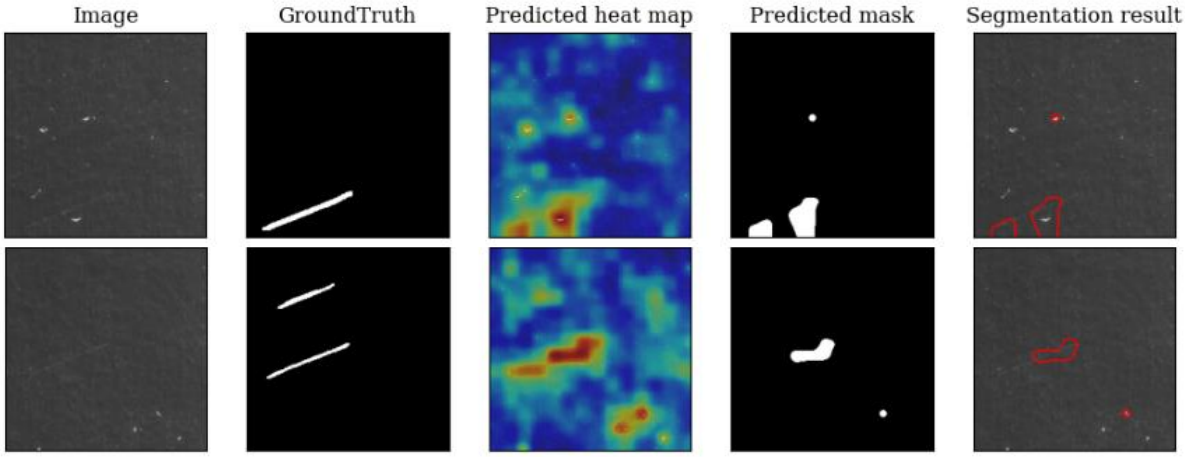


Figure 6: Three defects that are not localized during the LOOCV procedure with c=1 are displayed from a trial run at a higher image resolution (c=3). Although the defects are very faint, they can be segmented by the algorithm.

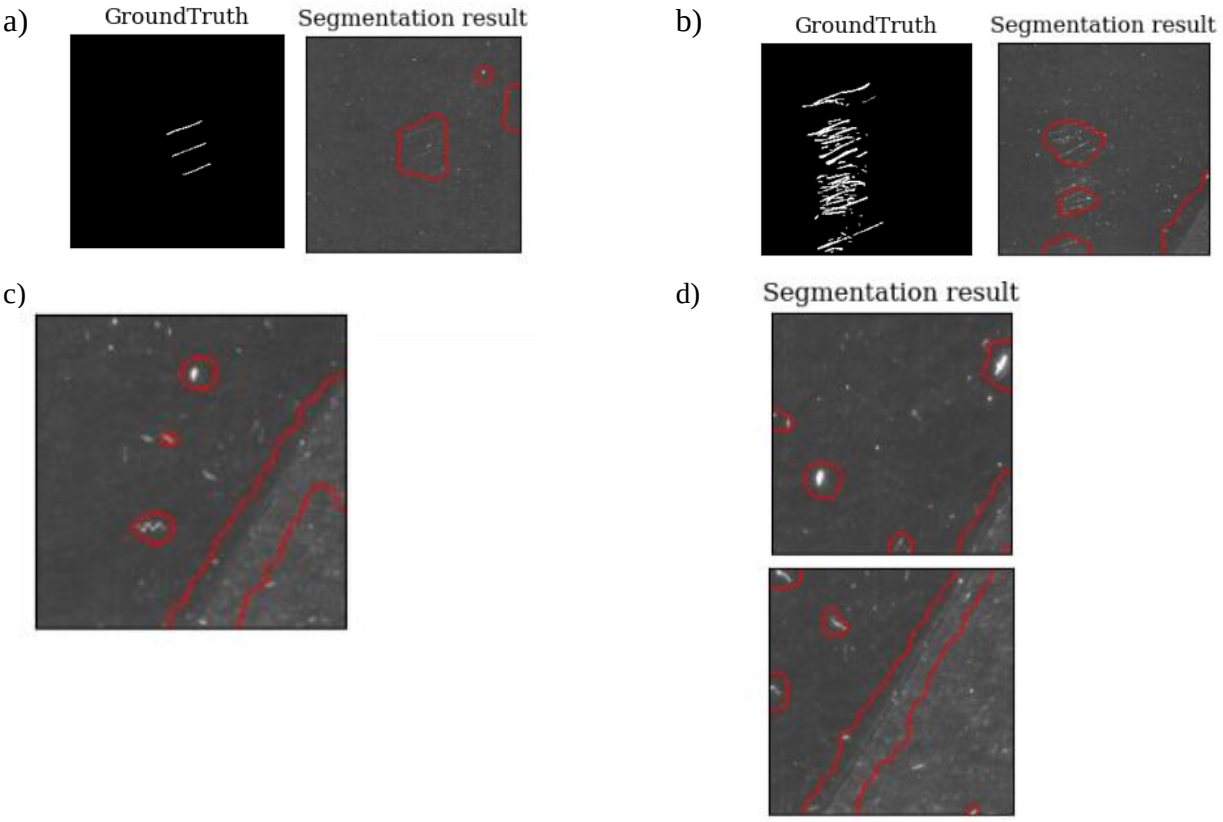


Figure 7: Two examples of false positives are shown in 6a) and 6b), respectively. These are taken after the LOOCV (leave-one-out cross-validation) procedure was used and 100 features were randomly selected in each run. 6a) shows a small bright spot (upper right) and a slight discoloration (center right) that were predicted as anomalous. 6b) shows a lighter triangular discoloration in the lower right corner, which was not labeled as anomalous in the ground truth. 6c) shows the same discoloration as that in 6b), except it is taken from training the model at a higher resolution where the scaling factor c=3. 5d) shows the same discoloration at c=5. As resolution increases, more random bright spots become incorrectly predicted as anomalous.

Overall, as c increases, smaller, fainter defects can be localized at the expense of more false positives, which likely explains the slight increase in pixel-level ROC-AUC as c increases. However, the image-level ROC-AUC appears to decrease as the scaling factor c increases. This is probably because as c increases, the entire membrane image is divided into smaller sub-images, and more defects are cut-off and spread out among multiple sub-images. Some sub-images may contain a very small portion of a defect that becomes unrecognizable from the original defect, which may confuse the model when trying to classify the image as a whole. An example is shown in Figure 8, which is a sub-image created from splitting the membrane image with c=3, which shows a small sliver of a scratch defect.

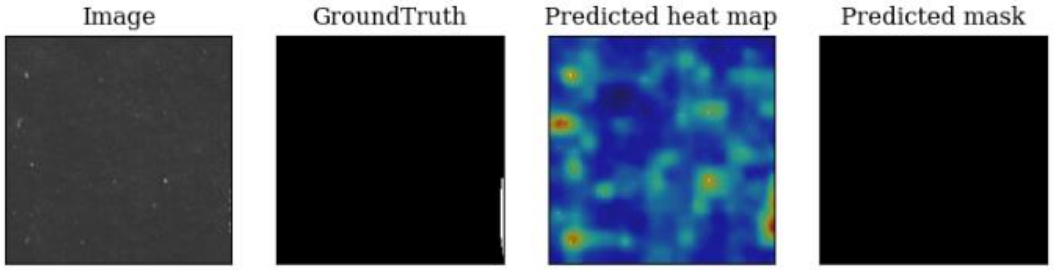


Figure 8: A very small part of a scratch appears in the bottom right side of the image, but it is almost unrecognizable and may make the image difficult to predict correctly. The original image is shown, with the ground truth highlighting the area as anomalous; a heat map of pixel anomaly scores, where higher anomaly scores are colored in red and yellow; and the predicted mask which segments areas that have an anomaly score greater than the chosen anomaly threshold. The mask shows that the model does not localize any defects in the sample.

The outer threshold PRO-scores of the scratches and scuffs dataset are lower than those of the pinholes and there is more variation in performance. A histogram of threshold PRO scores for each image evaluated in the outer loop during LOOCV with dimensionality reduction is shown in Figure 9—3 defects have a value of 0 (were missed entirely).

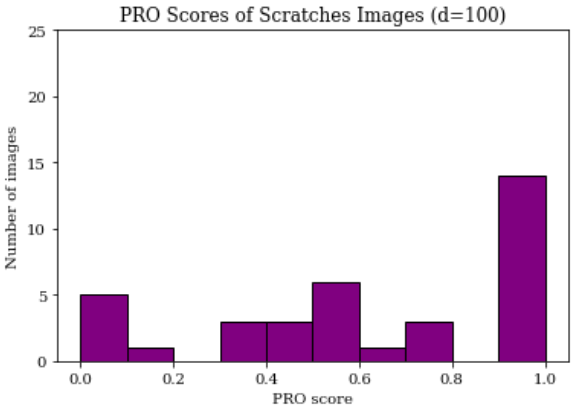


Figure 9: A histogram distribution of PRO scores for each image of scratches or scuffs is shown. 3 scratches/scuffs images had a value of 0, but the rest have nonzero PRO scores. Scores are obtained in trials where 100 random features were selected (d=100).

As is evident from a comparison of Tables 1 and 3, the outer LOOCV threshold PRO score is higher than both the inner LOOCV threshold PRO score and the threshold PRO score from the standard procedure for the scratches and scuffs dataset. This is caused by differences in calculating the different scores. As described in Bergmann et al. [59], the threshold PRO score is calculated for a dataset by looping through all the connected components in the ground truth masks and calculating the relative area of each connected component that is segmented by the model at a chosen threshold. Then, the relative areas are averaged over all the connected components. In this study, the outer threshold PRO score is found by calculating an individual threshold PRO score for each defective image and then averaging over all the defective images' PRO scores uniformly. Thus, each defect has equal impact on the final score. contrast, the inner and standard threshold PRO scores are calculated by averaging the relative areas of all the connected components in all the defect images at once. Because each connected component is weighted equally, defects with many connected components have a greater impact on the final score. Here, the scuffs, consisting of a cluster of small scratches and discolorations, have far more connected components than the scratches, so if the model performs poorly on a scuff defect, the inner and standard threshold PRO as well as the PRO-AUC are lowered disproportionately compared to scratch defects. So far, there is no reason to use a score that gives the scuffs more weight when evaluating model performance, because although they have more connected components, the components are generally fainter and smaller than those of the scratches, so the scuffs are not necessarily more serious defects than the scratches. Thus, it can be argued that the outer threshold PRO score is a more useful indicator of performance than the threshold PRO score obtained in the standard procedure. It should be noted that his issue does not occur with the pinhole dataset, as noted in Tables 2 and 4 where all the threshold PRO scores are very close. This is because the ground truth mask for each defect only consists of one connected component so each defect has equal weight when calculating the inner and standard threshold PRO.

The object detection metrics evaluated were AP, AP50, and AP75, which are calculated by finding the area under the precision-recall curve at a specified Intersection over Union (IoU) threshold [72]. Mean scores across the 30 repeated runs are recorded for pinholes in Table 5 and scratches and scuffs in Table 6 at a 95% confidence interval. All averages and confidence intervals were rounded to the nearest tenth. The inference time averaged over the 90 total trials is displayed in Figure 7 at a 95% confidence interval. Lastly, some success and failure cases are shown in Figure 10.

Shot	AP	AP50	AP75
5	33.6 ± 2.8	65.3 ± 5.1	30.7 ± 3.6
10	40.3 ± 1.3	72.1 ± 1.8	40.3 ± 2.9

Table 5: Performance metrics including AP, AP50, and AP75, obtained by fine-tuning a pre-trained Faster-RCNN model on the pinhole dataset, are displayed. 5-shot and 10-shot learning were implemented, and results are averaged at a 95% confidence interval over 30 different runs.

Object	AP	AP50	AP75
scratches	8.0 ± 0.9	17.7 ± 1.7	6.0 ± 1.3
scuffs	3.9 ± 1.2	22.2 ± 5.6	2.0 ± 1.6

Table 6: Performance metrics including AP, AP50, and AP75, obtained by fine-tuning a pre-trained Faster-RCNN model on the scratches and scuffs dataset, are displayed. Only 5-shot learning was implemented due to the small dataset size, and results are averaged at a 95% confidence interval over 30 different runs.

Inference time (milliseconds)
$39.8 \pm 4390 \times 10^{-5}$

Table 7: Inference time of the Faster-RCNN model, using 1 GPU, averaged over the recorded inference times from each of the 90 trials. The inference time is recorded at a 95% confidence interval.

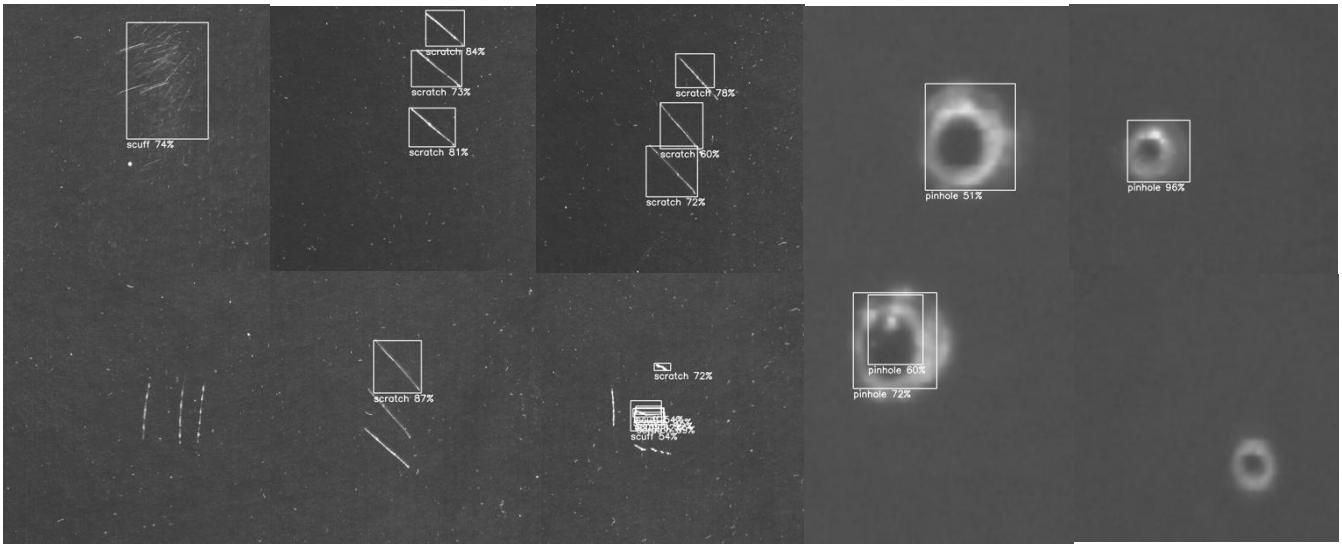


Figure 10: The top row shows success cases of training the Faster-RCNN. The first three images in each row show examples from the images of scratches and scuffs, while the last two show examples of pinholes. Failure cases include not predicting a bounding box with sufficiently high confidence and overlapping predictions. For example, a scratch can be predicted to be both a scratch and scuff.

The confidence intervals are large when compared to the intervals seen in Wang et al. [70]. This is likely due to the small size of the test data. For example, 12 images of scuffs were available in total, so 7 images were used for testing in each run (as 5 images were used for training). Such a small number of images causes individual differences in each image to make a larger relative impact on the overall AP score, leading to higher variance. Furthermore, a different sample was taken from the images to serve as test data in each repeated run, so different test datasets also lead to higher variance. The mean AP scores also give mixed results. For the pinholes, they are relatively high, and in fact surpass the scores seen on the benchmarks seen in Wang et al. [70]. This is unexpected, because a rudimentary fine-tuning approach was used, keeping the backbone frozen the entire time without following the two-stage fine-tuning procedure described in Wang et al. [70] The source dataset used to train the model, which was the Pascal VOC training data containing color photographs, was largely dissimilar to the grayscale 2-D pinhole images, which could have also decreased the effectiveness of transfer learning. However, the high pinhole performance is unsurprising, as most of the pinholes look very similar, usually being either a large white dot or a large black/gray dot with a whiter border. Also, pinholes were the only object class, so there was less room for misclassification error.

The models underperform on the scratches and scuffs datasets, where its strongest score is AP50, with lower AP and AP75. This likely indicates that the model struggles to predict bounding boxes that precisely overlap with the ground truth boxes. These AP scores are not too surprising, however, as the TFA method by Wang et al. achieves AP scores on the COCO dataset that also appear to be in the single digits [70].

4. ANALYSIS

Overall, the models trained on the pinhole dataset shows great promise, with very high ROC-AUC and PRO-AUC scores, while the models trained on the scratches/scuffs dataset need much improvement. The main cause for this difference is the degree of uniformity in the baseline. Through visually inspecting the images qualitatively, the scratches/scuffs membranes were rougher and had more bright spots, which lead to more false positives. It is unlikely that these bright spots are real defects because in Rupnowski et al. [35], the they are postulated to be a result of specular reflection or scattering resulting from slight variations in surface roughness, and were not seriously treated as defects. Thus, it is possible that alternative methods for optical imaging of the membrane could reduce scattering and lead to fewer problematic bright spots. Using computer vision algorithms to remove them may be risky because they could accidentally remove small but real defects.

Very similar issues arise for the object detection model, that also lead to worse performance on the scratches and scuffs when compared to the pinholes. As noted previously, the membrane images containing the artificially induced scratches and scuffs are ‘noisier’ with more bright spots in the background, many of which look like the artificially induced scratches and scuffs, which may confuse the model. Some of those spots may in fact be scratches and scuffs from other sources, but it is impossible to know the causes for all these bright spots. Following the convention outlined in the original study in Rupnowski et al. [35], most objects that resembled scratches and scuffs but whose nature were unknown and didn’t seem artificially created were not labeled--only artificially created defects were counted. Avoiding drawing a box arbitrarily around any object that resembled a defect minimized human bias during the labeling process. However, some defects’ origins were unknown so heuristic judgement had to be used. For example, there were long curved streaks on a membrane that were labeled as scratches. They were not grouped with other scratches and were shaped significantly different from the other scratches. They were also located near a scuff, so they may have been created accidentally when the scuff was being created. Nevertheless, they were still labeled as scratches because they greatly stood out in the image, even if they are unlikely to be significant according to Rupnowski et al. [35]. To avoid such confusion in future studies, it would be helpful to increase collaboration with experimentalists with domain expertise

who could assist with labeling in the training data. The image containing these defects is shown in Figure 11.

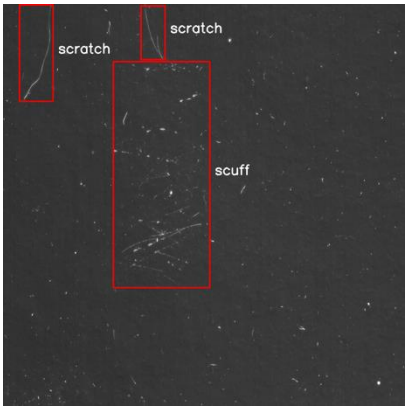


Figure 11: A section of the membrane is shown with a scuff in the center that was artificially created. There are also white streaks that were labeled as scratches, but their origin is not fully known.

Another obstacle is the overlap in the morphology of scuffs and scratches. Although the scratches and scuffs were created differently, many scuffs resemble a large, tight cluster of scratches, and a group of scratches may also be seen as a very small, sparse scuff. If the scratches inside the scuff defects are not labeled as scratches, the model is discouraged from being able to identify scratches based on appearance alone, and its performance degrades.

Other obstacles include many scratches being carved close to each other, so that overlap between different bounding boxes around scratches likely degraded performance. Lastly, many scuffs, being a loose cluster of scratches, had ambiguous bordering such that drawing bounding boxes required heuristic judgement. This likely caused their low AP and AP75 scores as the model may predict a box that surrounds a large portion of a scuff, but does not intersect with the ground truth bounding box.

Finally, detection times are estimated to help assess viability for real-time defect detection. For deep learning optical methods to be viable, they need to process large volumes of a membrane quickly: facilities may have a coating speed of 10 feet per minute [35], which need to be matched by the algorithm to find defects in real time. Without a manufacturing line to test our models, the detection time is roughly estimated using the time-complexity metrics for PaDiM and the inference times for Faster-RCNN as reported in the Results section.

For PaDiM, we focus on the pinhole dataset, which should be more time-consuming to detect than the scratches as higher resolutions were needed. For the membranes with pinholes, each 8 × 11” membrane image was divided into 224 × 244 pixel sub-images for processing by the PaDiM algorithm. This corresponds to 3924 sub-images that need to be inspected to find defects on the four membrane specimens. However, this was after cropping the scanned image so that its height and width were a multiple of 224 pixels, so a small part of the membrane on the border was removed. If the border was included and the membrane was not cropped, then splitting the membranes into 224 × 224-sized sub-images would result in 4181 sub-images total. This amount more realistically represents the number of tiles that need to be processed so that the entire membrane can be inspected. It should be noted that this is not an exact multiple of 4 because the different scanned membrane images had slightly different sizes. The time taken to extract features for all these images and create an anomaly map is roughly estimated using Equation 1:

$$T_{total} \approx \frac{N_{total}}{N_{testing}} T_{testing}^{processing}$$

Here, T_{total} represents the total amount of time taken to create an anomaly map for N_{total} sub-images, which is assumed to be 4181; $N_{testing}$ is the number of images in the test dataset (75) and $T_{testing}^{processing}$ is the processing time for the test dataset as described in the Results. This estimation assumes that the time duration is linear in the amount images to be processed. For the pinholes, the total processing time for the test dataset was 1.6 seconds with no dimensionality reduction. Thus, the amount of time T_{total} to inspect the four membrane specimens is calculated as:

$$T_{total} \approx \frac{4181}{75} 1.6 \approx 89\ seconds$$

89 seconds is a long time and does not account for other steps like image calibration; to reach the goal of 10 ft/minute, the four 11" long membranes in this study would need to be inspected in around 22 seconds. The reactive excitative method developed by Ulsh et al. [26] can find pinholes in 7.5 × 7.5 cm samples in a little over 5 seconds. A direct comparison between the two methods is difficult because studies have not yet been performed on how time duration of the reactive excitative method would scale up as sample area increases to the sizes of the membranes used in this study. However, it can be estimated that a 7.5 × 7.5 cm portion of the pinhole membrane would correspond to approximately 104 sub-images, so the amount of time to inspect it would be approximately:

$$T_{total} \approx \frac{104}{75} 1.6 \approx 2.2\ seconds$$

Thus, PaDiM's inspection time seems comparable to some IR thermography methods.

Several other measures can be taken to improve processing speed. Other more recent unsupervised anomaly detection algorithms like Fastflow [73] and CFLOW-AD [74] might yield improved accuracy and inference time over PaDiM. However, the parallelized PaDiM implementation in this study is also a magnitude faster than the original serial implementation in Defard et al. [60], so Fastflow and CFLOW-AD are not guaranteed to be faster. Future research could also investigate parallelizing the inspection process with multiple cameras and/or computers, which could divide the processing time by 2 or more. Minimizing the real-time image pre-processing needed by training the PaDiM models on raw un-processed images could also optimize processing time. For example, Rupnowski et al. [35] describes how sample images may need to be calibrated due to differences in light intensity at different angles from the light source. Training the models to on uncalibrated images instead may negate the need for calibration.

A similar calculation on processing time is made for object detection algorithms. In this study, Faster-RCNN has a reported inference time of 0.04 seconds for 1 image. The images used to train and test the model were created from dividing each of the four pinhole-containing membrane images into size 66 × 66 sub-images, which resulted in 46,219 66 × 66 pixel sub-images total. For the inference stage, it would take approximately 30.8 minutes to make a prediction for all the 46,219 sub-images, which is too slow to achieve real-time defect detection of pinholes. For the scratches and scuffs, however, each membrane was divided into 18 sub-images, so it would take approximately 7.2 seconds to perform inference on all the sub-images from one membrane. It is clear that increasing the image resolution greatly slows down the inspection time.

5. CONCLUSION

The application of deep learning for in-line defect detection is a promising area overall. The anomaly detection model PaDiM shows more promise than the Faster-RCNN model, being orders of magnitude faster. PaDiM in this study could detect faint scratches, scuffs, and artificially induced pinholes, the same defects that are usually found using infrared thermography [26]. Furthermore, because feature selection was not used, PaDiM will likely generalize well to other types of visible defects besides

pinholes, scratches, and scuffs. However, the utility of deep computer vision is limited to visible defects that can be seen on camera. Invisible defects, like catalyst layer thickness irregularities on gas diffusion electrodes [25], will be less straightforward to detect through deep computer vision.

The speed of PaDiM is within the magnitude of real-time detection speed, even when detecting pinholes near a microscopic scale. However, the Faster-RCNN models are too slow. In the future, object detection can be improved by investigating more recent object detection architectures like YOLOv7 with reported inference times as fast as 286 FPS [75]. Performance at lower resolutions could also be investigated.

The main obstacle for detecting visible defects is "noisy" baselines: sample images may have very bright spots which are not supposed to be defects and can cause false positives. Although they are not of interest as harmful defects, they are nevertheless "anomalous" in a sense that their location and occurrence can be unpredictable. It was found that even if training data contained random bright spots, other bright spots in the test data were still labeled as anomalous. A lower frequency of bright spots likely allowed the model to perform well on the pinhole-containing samples, and a high frequency caused the model to perform more poorly on the scratches/scuffs-containing samples.

It may be worthwhile investigating more hybrid approaches combining image preprocessing using computer vision algorithms and deep learning. For example, adjusting image contrast may assist in detecting faint defects, and smoothing algorithms may help remove background noise. However, researchers should be cautious to ensure that such preprocessing generalizes well and does not accidentally improve detection for some defects at the cost of obscuring other defects.

Further studies could do a more direct comparison between the optical methods used in this study and infrared thermography methods, such as analyzing the energy and resource consumption of the two. Infrared thermography methods need a steady supply of reactive gas like H2, energy to pump the gas through the material and purge it, as well an IR camera. On the other hand, optical detection methods in this study need a camera and light source to capture the images, and a computer with GPUs for the model to make calculations. While deep learning negates the need for reactive gas, the many calculations needed might be energy intensive.

Finally, collaboration between machine learning engineers and experimentalists that have developed infrared thermography methods would be beneficial: expertise in the characterization and the effects of different defects would help create realistic, accurate datasets. Additionally, a tight feedback loop between experimentalists and programmers could greatly enable the rapid acquisition of new membrane samples to be used as training data. Overall, the application of deep learning for defect detection in PEMs will likely require more collaborative and interdisciplinary efforts. However, if successful, the applicability of optical methods for detecting defects could be greatly enhanced.

Acknowledgements

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Energy Efficiency and Renewable Energy and the Hydrogen and Fuel Cell Technologies Office. This work was also sponsored by the Office of Science and the Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internship (SULI) program. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

References

[1] X.-Z. Yuan, C. Nayoze-Coynel, N. Shaigan, D. Fisher, N. Zhao, N. Zamel, P. Gazdzicki, M. Ulsh, K.A. Friedrich, F. Girard, U. Groos, A review of functions, attributes, properties and

measurements for the quality control of proton exchange membrane fuel cell components, J. Power Sources. 491 (2021) 229540. https://doi.org/10.1016/j.jpowsour.2021.229540.
[2] Y.-H. Cho, H.-S. Park, J. Kim, Y.-H. Cho, S.W. Cha, Y.-E. Sung, The Operation Characteristics of MEAs with Pinholes for Polymer Electrolyte Membrane Fuel Cells, Electrochem. Solid-State Lett. 11 (2008) B153. https://doi.org/10.1149/1.2937450.
[3] A.M. Niroumand, H. Homayouni, G. Goransson, M. Olfert, M. Eikerling, In-situ diagnostic tools for hydrogen transfer leak characterization in PEM fuel cell stacks part III: Manufacturing applications, J. Power Sources. 448 (2020) 227359. https://doi.org/10.1016/j.jpowsour.2019.227359.
[4] S. Kreitmeier, M. Michiardi, A. Wokaun, F.N. Büchi, Factors determining the gas crossover through pinholes in polymer electrolyte fuel cell membranes, Electrochimica Acta. 80 (2012) 240–247. https://doi.org/10.1016/j.electacta.2012.07.013.
[5] W. Zheng, L. Xu, Z. Hu, Y. Ding, J. Li, M. Ouyang, Dynamic modeling of chemical membrane degradation in polymer electrolyte fuel cells: Effect of pinhole formation, J. Power Sources. 487 (2021) 229367. https://doi.org/10.1016/j.jpowsour.2020.229367.
[6] V.M. Ehlinger, A. Kusoglu, A.Z. Weber, Modeling Coupled Durability and Performance in Polymer-Electrolyte Fuel Cells: Membrane Effects, J. Electrochem. Soc. 166 (2019) F3255. https://doi.org/10.1149/2.0281907jes.
[7] M. p. Arcot, K. Zheng, J. McGrory, M. w. Fowler, M. d. Pritzker, Investigation of catalyst layer defects in catalyst-coated membrane for PEMFC application: Non-destructive method, Int. J. Energy Res. 42 (2018) 3615–3632. https://doi.org/10.1002/er.4107.
[8] J. Shi, Z. Zhan, D. Zhang, Y. Yu, X. Yang, L. He, M. Pan, Effects of Cracks on the Mass Transfer of Polymer Electrolyte Membrane Fuel Cell with High Performance Membrane Electrode Assembly, J. Wuhan Univ. Technol.-Mater Sci Ed. 36 (2021) 318–330. https://doi.org/10.1007/s11595-021-2412-z.
[9] A.P. Soleymani, J. Chen, M. Ricketts, J. Waldecker, J. Jankovic, Failure Analysis and Defects Characterization of Polymer Electrolyte Membrane Fuel Cells after Relative Humidity Cycling, ECS Trans. 98 (2020) 109. https://doi.org/10.1149/09809.0109ecst.
[10] Q. Tang, B. Li, D. Yang, P. Ming, C. Zhang, Y. Wang, Review of hydrogen crossover through the polymer electrolyte membrane, Int. J. Hydrog. Energy. 46 (2021) 22040–22061. https://doi.org/10.1016/j.ijhydene.2021.04.050.
[11] A. Phillips, M. Ulsh, J. Mackay, T. Harris, N. Shrivastava, A. Chatterjee, J. Porter, G. Bender, The Effect of Membrane Casting Irregularities on Initial Fuel Cell Performance, Fuel Cells. 20 (2020) 60–69. https://doi.org/10.1002/fuce.201900149.
[12] M. Wang, G. Rome, S. Medina, J.R. Pfeilsticker, Z. Kang, S. Pylypenko, M. Ulsh, G. Bender, Impact of electrode thick spot irregularities on polymer electrolyte membrane fuel cell initial performance, J. Power Sources. 466 (2020) 228344. https://doi.org/10.1016/j.jpowsour.2020.228344.
[13] S. Shi, X. Sun, Q. Lin, J. Chen, Y. Fu, X. Hong, C. Li, X. Guo, G. Chen, X. Chen, Fatigue crack propagation behavior of fuel cell membranes after chemical degradation, Int. J. Hydrog. Energy. 45 (2020) 27653–27664. https://doi.org/10.1016/j.ijhydene.2020.07.113.
[14] S. Ma, Y. Qin, Y. Liu, L. Sun, Q. Guo, Y. Yin, Delamination evolution of PEM fuel cell membrane/CL interface under asymmetric RH cycling and CL crack location, Appl. Energy. 310 (2022) 118551. https://doi.org/10.1016/j.apenergy.2022.118551.

[15] R. Banan, J. Zu, A. Bazylak, Humidity and Temperature Cycling Effects on Cracks and Delaminations in PEMFCs, Fuel Cells. 15 (2015) 327–336. https://doi.org/10.1002/fuce.201400118.
[16] A. Tavassoli, C. Lim, J. Kolodziej, M. Lauritzen, S. Knights, G.G. Wang, E. Kjeang, Effect of catalyst layer defects on local membrane degradation in polymer electrolyte fuel cells, J. Power Sources. 322 (2016) 17–25. https://doi.org/10.1016/j.jpowsour.2016.05.016.
[17] A. Phillips, M. Ulsh, J. Porter, G. Bender, Utilizing a Segmented Fuel Cell to Study the Effects of Electrode Coating Irregularities on PEM Fuel Cell Initial Performance, Fuel Cells. 17 (2017) 288–298. https://doi.org/10.1002/fuce.201600214.
[18] A. Phillips, M. Ulsh, K.C. Neyerlin, J. Porter, G. Bender, Impacts of electrode coating irregularities on polymer electrolyte membrane fuel cell lifetime using quasi in-situ infrared thermography and accelerated stress testing, Int. J. Hydrog. Energy. 43 (2018) 6390–6399. https://doi.org/10.1016/j.ijhydene.2018.02.050.
[19] S.R. Choi, W.Y. An, S. Choi, K. Park, S.-D. Yim, J.-Y. Park, Assessing the degradation pattern and mechanism of membranes in polymer electrolyte membrane fuel cells using open-circuit voltage hold and humidity cycle test protocols, Mater. Sci. Energy Technol. 5 (2022) 66–73. https://doi.org/10.1016/j.mset.2021.12.001.
[20] Q. Lin, S. Shi, L. Wang, X. Chen, G. Chen, Biaxial fatigue crack propagation behavior of perfluorosulfonic-acid membranes, J. Power Sources. 384 (2018) 58–65. https://doi.org/10.1016/j.jpowsour.2018.02.002.
[21] S. Shi, J. Li, B. Gao, H. Dai, Q. Lin, X. Chen, Effect of catalyst layer on fatigue life and fracture mechanisms of fuel cell membrane, Fatigue Fract. Eng. Mater. Struct. 45 (2022) 687–700. https://doi.org/10.1111/ffe.13626.
[22] M. Wang, S. Medina, J. Ochoa-Lozano, S. Mauger, S. Pylypenko, M. Ulsh, G. Bender, Visualization, understanding, and mitigation of process-induced-membrane irregularities in gas diffusion electrode-based polymer electrolyte membrane fuel cells, Int. J. Hydrog. Energy. 46 (2021) 14699–14712. https://doi.org/10.1016/j.ijhydene.2021.01.186.
[23] M. Ulsh, J.M. Porter, D.C. Bittinat, G. Bender, Defect Detection in Fuel Cell Gas Diffusion Electrodes Using Infrared Thermography, Fuel Cells. 16 (2016) 170–178. https://doi.org/10.1002/fuce.201500137.
[24] M. Obermaier, K. Jozwiak, M. Rauber, A. Bauer, C. Scheu, Comparative study of pinhole detection methods for automotive fuel cell degradation analysis, J. Power Sources. 488 (2021) 229405. https://doi.org/10.1016/j.jpowsour.2020.229405.
[25] I.V. Zenyuk, N. Englund, G. Bender, A.Z. Weber, M. Ulsh, Reactive impinging-flow technique for polymer-electrolyte-fuel-cell electrode-defect detection, J. Power Sources. 332 (2016) 372–382. https://doi.org/10.1016/j.jpowsour.2016.09.109.
[26] M. Ulsh, A. DeBari, J.M. Berliner, I.V. Zenyuk, P. Rupnowski, L. Matvichuk, A.Z. Weber, G. Bender, "The development of a through-plane reactive excitation technique for detection of pinholes in membrane-containing MEA sub-assemblies," Int. J. Hydrog. Energy. 44 (2019) 8533–8547. https://doi.org/10.1016/j.ijhydene.2018.12.181.
[27] R. Su, M. Kirillin, E.W. Chang, E. Sergeeva, S.H. Yun, L. Mattsson, Perspectives of mid-infrared optical coherence tomography for inspection and micrometrology of industrial ceramics, Opt. Express. 22 (2014) 15804–15819. https://doi.org/10.1364/OE.22.015804.
[28] P. Rupnowski, M. Ulsh, B. Sopori, B.G. Green, D.L. Wood, J. Li, Y. Sheng, In-line monitoring of Li-ion battery electrode porosity and areal loading using active thermal

scanning - modeling and initial experiment, J. Power Sources. 375 (2018) 138–148. https://doi.org/10.1016/j.jpowsour.2017.07.084.
[29] B.L. Sopori, M.J. Ulsh, P. Rupnowski, G. Bender, M.M. Penev, J. Li, D.L.W. III, C. Daniel, Batch and continuous methods for evaluating the physical and thermal properties of films, US10684128B2, 2020. https://patents.google.com/patent/US10684128B2/en (accessed May 16, 2022).
[30] P. Rupnowski, M.J. Ulsh, Thickness mapping using multispectral imaging, US10480935B2, 2019. https://patents.google.com/patent/US10480935B2/en (accessed May 12, 2022).
[31] B. Sopori, P. Rupnowski, M. Ulsh, On-line, continuous monitoring in solar cell and fuel cell manufacturing using spectral reflectance imaging, National Renewable Energy Lab. (NREL), Golden, CO (United States), 2016. https://www.osti.gov/doepatents/biblio/1234545-line-continuous-monitoring-solar-cell-fuel-cell-manufacturing-using-spectral-reflectance-imaging (accessed May 12, 2022).
[32] B. Sopori, M. Ulsh, P. Rupnowski, Optical techniques for monitoring continuous manufacturing of proton exchange membrane fuel cell components, US20130226330A1, 2013. https://patents.google.com/patent/US20130226330A1/en (accessed May 16, 2022).
[33] D. Choi, Y.-J. Jeon, S.H. Kim, S. Moon, J.P. Yun, S.W. Kim, Detection of Pinholes in Steel Slabs Using Gabor Filter Combination and Morphological Features, ISIJ Int. 57 (2017) 1045–1053. https://doi.org/10.2355/isijinternational.ISIJINT-2016-160.
[34] W.-T. Chen, R.A. Chowdhury, J. Youngblood, G.T.-C. Chiu, A Two-step Thresholding Algorithm for Image-based Defect Detection in Roll-to-Roll Coating of Cellulose Nanocrystal Films, in: 2018 Annu. Am. Control Conf. ACC, 2018: pp. 4440–4445. https://doi.org/10.23919/ACC.2018.8431596.
[35] P. Rupnowski, M. Ulsh, B. Sopori, High Throughput and High Resolution In-Line Monitoring of PEMFC Materials by Means of Visible Light Diffuse Reflectance Imaging and Computer Vision, in: American Society of Mechanical Engineers Digital Collection, 2015. https://doi.org/10.1115/FUELCELL2015-49212.
[36] johnson_jay_t_200912_mast.pdf, (n.d.). https://smartech.gatech.edu/bitstream/handle/1853/37234/johnson_jay_t_200912_mast.pdf (accessed May 11, 2022).
[37] M.T.N. Truong, S. Kim, Automatic image thresholding using Otsu's method and entropy weighting scheme for surface defect detection, Soft Comput. 22 (2018) 4197–4203. https://doi.org/10.1007/s00500-017-2709-1.
[38] A.S. Tolba, H.M. Raafat, Multiscale image quality measures for defect detection in thin films, Int. J. Adv. Manuf. Technol. 79 (2015) 113–122. https://doi.org/10.1007/s00170-014-6758-7.
[39] Y. Wang, B. Seo, B. Wang, N. Zamel, K. Jiao, X.C. Adroher, Fundamentals, materials, and machine learning of polymer electrolyte membrane fuel cell technology, Energy AI. 1 (2020) 100014. https://doi.org/10.1016/j.egyai.2020.100014.
[40] S. Zhou, P.R. Shearing, D.J.L. Brett, R. Jervis, Machine learning as an online diagnostic tool for proton exchange membrane fuel cells, Curr. Opin. Electrochem. 31 (2022) 100867. https://doi.org/10.1016/j.coelec.2021.100867.
[41] Y. Xing, B. Wang, Z. Gong, Z. Hou, F. Xi, G. Mou, Q. Du, F. Gao, K. Jiao, Data-driven Fault Diagnosis for PEM Fuel Cell System Using Sensor Pre-Selection Method and Artificial

Neural Network Model, IEEE Trans. Energy Convers. (2022) 1–1. https://doi.org/10.1109/TEC.2022.3143163.
[42] J.Y. Park, I.S. Lim, E.J. Choi, M.S. Kim, Fault diagnosis of thermal management system in a polymer electrolyte membrane fuel cell, Energy. 214 (2021) 119062. https://doi.org/10.1016/j.energy.2020.119062.
[43] J. Liu, Q. Li, W. Chen, Y. Yan, X. Wang, A Fast Fault Diagnosis Method of the PEMFC System Based on Extreme Learning Machine and Dempster–Shafer Evidence Theory, IEEE Trans. Transp. Electrification. 5 (2019) 271–284. https://doi.org/10.1109/TTE.2018.2886153.
[44] S. Zhou, Y. Lu, D. Bao, K. Wang, J. Shan, Z. Hou, Real-time data-driven fault diagnosis of proton exchange membrane fuel cell system based on binary encoding convolutional neural network, Int. J. Hydrog. Energy. 47 (2022) 10976–10989. https://doi.org/10.1016/j.ijhydene.2022.01.145.
[45] J. Xie, C. Wang, W. Zhu, H. Yuan, A Multi-Stage Fault Diagnosis Method for Proton Exchange Membrane Fuel Cell Based on Support Vector Machine with Binary Tree, Energies. 14 (2021) 6526. https://doi.org/10.3390/en14206526.
[46] W. Fan, B. Xu, H. Li, G. Lu, Z. Liu, A novel surrogate model for channel geometry optimization of PEM fuel cell based on Bagging-SVM Ensemble Regression, Int. J. Hydrog. Energy. 47 (2022) 14971–14982. https://doi.org/10.1016/j.ijhydene.2022.02.239.
[47] L.A. Briceno-Mena, G. Venugopalan, J.A. Romagnoli, C.G. Arges, Machine learning for guiding high-temperature PEM fuel cells with greater power density, Patterns. 2 (2021) 100187. https://doi.org/10.1016/j.patter.2020.100187.
[48] B. Wang, B. Xie, J. Xuan, K. Jiao, AI-based optimization of PEM fuel cell catalyst layers for maximum power density via data-driven surrogate modeling, Energy Convers. Manag. 205 (2020) 112460. https://doi.org/10.1016/j.enconman.2019.112460.
[49] M. Derbeli, C. Napole, O. Barambones, Machine Learning Approach for Modeling and Control of a Commercial Heliocentris FC50 PEM Fuel Cell System, Mathematics. 9 (2021) 2068. https://doi.org/10.3390/math9172068.
[50] J. Wang, R. Ding, F. Cao, J. Li, H. Dong, T. Shi, L. Xing, J. Liu, Comparison of state-of-the-art machine learning algorithms and data-driven optimization methods for mitigating nitrogen crossover in PEM fuel cells, Chem. Eng. J. 442 (2022) 136064. https://doi.org/10.1016/j.cej.2022.136064.
[51] A. Morán-Durán, A. Martínez-Sibaja, J.P. Rodríguez-Jarquin, R. Posada-Gómez, O.S. González, PEM Fuel Cell Voltage Neural Control Based on Hydrogen Pressure Regulation, Processes. 7 (2019) 434. https://doi.org/10.3390/pr7070434.
[52] T. Czimmermann, G. Ciuti, M. Milazzo, M. Chiurazzi, S. Roccella, C.M. Oddo, P. Dario, Visual-Based Defect Detection and Classification Approaches for Industrial Applications—A SURVEY, Sensors. 20 (2020) 1459. https://doi.org/10.3390/s20051459.
[53] T. Wang, Y. Chen, M. Qiao, H. Snoussi, A fast and robust convolutional neural network-based defect detection model in product quality control, Int. J. Adv. Manuf. Technol. 94 (2018) 3465–3471. https://doi.org/10.1007/s00170-017-0882-0.
[54] X. Fang, Q. Luo, B. Zhou, C. Li, L. Tian, Research Progress of Automated Visual Surface Defect Detection for Industrial Metal Planar Materials, Sensors. 20 (2020) 5136. https://doi.org/10.3390/s20185136.

[55] S. Shahrabadi, Y. Castilla, M. Guevara, L.G. Magalhães, D. Gonzalez, T. Adão, Defect detection in the textile industry using image-based machine learning methods: a brief review, J. Phys. Conf. Ser. 2224 (2022) 012010. https://doi.org/10.1088/1742-6596/2224/1/012010.
[56] E. Westphal, H. Seitz, A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks, Addit. Manuf. 41 (2021) 101965. https://doi.org/10.1016/j.addma.2021.101965.
[57] P.M. Bhatt, R.K. Malhan, P. Rajendran, B.C. Shah, S. Thakar, Y.J. Yoon, S.K. Gupta, Image-Based Surface Defect Detection Using Deep Learning: A Review, J. Comput. Inf. Sci. Eng. 21 (2021). https://doi.org/10.1115/1.4049535.
[58] B. Si, M. Yasengjiang, H. Wu, Deep learning-based defect detection for hot-rolled strip steel, J. Phys. Conf. Ser. 2246 (2022) 012073. https://doi.org/10.1088/1742-6596/2246/1/012073.
[59] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, C. Steger, The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection, Int. J. Comput. Vis. 129 (2021) 1038–1059. https://doi.org/10.1007/s11263-020-01400-4.
[60] T. Defard, A. Setkov, A. Loesch, R. Audigier, PaDiM: a Patch Distribution Modeling Framework for Anomaly Detection and Localization, (2020). https://doi.org/10.48550/arXiv.2011.08785.
[61] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, arXiv, 2016. https://doi.org/10.48550/arXiv.1506.01497.
[62] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, arXiv, 2016. https://doi.org/10.48550/arXiv.1506.02640.
[63] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conf. Comput. Vis. Pattern Recognit., 2009: pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848.
[64] M. Tan, Q.V. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, ArXiv190511946 Cs Stat. (2020). http://arxiv.org/abs/1905.11946 (accessed May 12, 2022).
[65] ingbeeedd, PaDiM-EfficientNet, 2022. https://github.com/ingbeeedd/PaDiM-EfficientNet (accessed June 4, 2022).
[66] P.C. Mahalanobis, On the generalized distance in Statistics, (1936). http://localhost:8080/xmlui/handle/10263/6765 (accessed June 21, 2022).
[67] P. Bergmann, M. Fauser, D. Sattlegger, C. Steger, Uninformed Students: Student-Teacher Anomaly Detection with Discriminative Latent Embeddings, in: 2020 IEEECVF Conf. Comput. Vis. Pattern Recognit. CVPR, 2020: pp. 4182–4191. https://doi.org/10.1109/CVPR42600.2020.00424.
[68] Y. Shi, J. Yang, Z. Qi, Unsupervised anomaly segmentation via deep feature reconstruction, Neurocomputing. 424 (2021) 9–22. https://doi.org/10.1016/j.neucom.2020.11.018.
[69] Z.C. Lipton, C. Elkan, B. Narayanaswamy, Thresholding Classifiers to Maximize F1 Score, arXiv, 2014. https://doi.org/10.48550/arXiv.1402.1892.
[70] X. Wang, T.E. Huang, T. Darrell, J.E. Gonzalez, F. Yu, Frustratingly Simple Few-Shot Object Detection, (2020). https://doi.org/10.48550/arXiv.2003.06957.

[71] N.-D. Nguyen, T. Do, T.D. Ngo, D.-D. Le, An Evaluation of Deep Learning Methods for Small Object Detection, J. Electr. Comput. Eng. 2020 (2020) e3189691. https://doi.org/10.1155/2020/3189691.
[72] J. Cartucho, R. Ventura, M. Veloso, Robust Object Recognition Through Symbiotic Deep Learning In Mobile Robots, in: 2018 IEEERSJ Int. Conf. Intell. Robots Syst. IROS, 2018: pp. 2336–2341. https://doi.org/10.1109/IROS.2018.8594067.
[73] J. Yu, Y. Zheng, X. Wang, W. Li, Y. Wu, R. Zhao, L. Wu, FastFlow: Unsupervised Anomaly Detection and Localization via 2D Normalizing Flows, (2021). https://doi.org/10.48550/arXiv.2111.07677.
[74] D. Gudovskiy, S. Ishizaka, K. Kozuka, CFLOW-AD: Real-Time Unsupervised Anomaly Detection with Localization via Conditional Normalizing Flows, arXiv, 2021. https://doi.org/10.48550/arXiv.2107.12571.
[75] C.-Y. Wang, A. Bochkovskiy, H.-Y.M. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, arXiv, 2022. https://doi.org/10.48550/arXiv.2207.02696.