

Autonomous Visual Detection of Defects from Battery Electrode Manufacturing

Nirmal Choudhary, Henning Clever, Robert Ludwigs, Michael Rath, Aymen Gannouni, Arno Schmetz, Tom Hülsmann, Julia Sawodny, Leon Fischer, Achim Kampker, Juergen Fleischer, and Helge S. Stein*

The increasing global demand for high-quality and low-cost battery electrodes poses major challenges for battery cell production. As mechanical defects on the electrode sheets have an impact on the cell performance and their lifetime, inline quality control during electrode production is of high importance. Correlation of detected defects with process parameters provides the basis for optimization of the production process and thus enables long-term reduction of reject rates, shortening of the production ramp-up phase, and maximization of equipment availability. To enable automatic detection of visually detectable defects on electrode sheets passing through the process steps at a speed of 9 m s^{-1} , a You-Only-Look-Once architecture (YOLO architecture) for the identification of visual detectable defects on coated electrode sheets is demonstrated within this work. The ability of the quality assurance (QA) system developed herein to detect mechanical defects in real time is validated by an exemplary integration of the architecture into the electrode manufacturing process chain at the Battery Lab Factory Braunschweig.

appropriate quality assurance systems are particularly desirable. Within this work, the process for manufacturing electrode sheets for lithium-ion battery (LiB) cells, a widely used and established energy storage system, is considered. The transferability of insights generated for this chemistry should, however, also extend to any other cell chemistry like Na- or Mg-ion batteries as we believe the herein investigated defects to be universal to the method and just different in occurrence for different chemistries. The electrode manufacturing process considered herein includes four production steps, which are shown in Figure 1. As the initial steps of the battery cell production, these have far-reaching effects on the battery performance and its lifetime^[3,4] and are thus dominant in quality determination. In order to avoid cascading effects in the further production steps,


1. Introduction

The demand for electrical energy storage is subject to an explosive growth.^[1] In this context, energy storage systems such as battery cells are becoming increasingly relevant. High-quality cells at the lowest possible cost represent one of the central challenges of battery cell production. One way to reduce manufacturing costs is to improve the production processes using digital methods.^[2,3] In this term, reliable production facilities that have

early-stage defect identification is extremely important to improve the electrode and manufacturing quality.^[5,6] As the coating and calendaring process account for approximately 22% of the total cost of electrode production, there is great potential for cost reduction in optimizing these processes.^[7–9] In particular, the coating and calendaring process may cause mechanical electrode defects that are visually detectable. Typical effects due to the coating effect^[4] (including drying) include agglomerations, bubbles/pinholes, cracks, or

N. Choudhary, L. Fischer, H. S. Stein
Helmholtz Institute Ulm (HIU)
Helmholtzstr. 11, 89081 Ulm, Germany
E-mail: dc6967@kit.edu

N. Choudhary, L. Fischer, H. S. Stein
Institute of Physical Chemistry (IPC)
Karlsruhe Institute of Technology (KIT)
Fritz-HaberWeg 2, 76131 Karlsruhe, Germany

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202200142>.

© 2022 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202200142

H. Clever, R. Ludwigs, A. Kampker
Chair of Production Engineering of E-Mobility Components (PEM)
RWTH Aachen University
Bohr 12, 52072 Aachen, Germany

M. Rath, A. Gannouni
Information Management in Mechanical Engineering (IMA)
RWTH Aachen University
Dennewartstr. 27, 52068 Aachen, Germany

A. Schmetz, T. Hülsmann
Fraunhofer Research Institution for Battery Cell Production FFB
Mendelstraße 11, 48149 Münster, Germany

J. Sawodny, J. Fleischer
WBK Institute of Production Science
Karlsruhe Institute of Technology
Kaiserstrasse 12, 76131 Karlsruhe, Germany

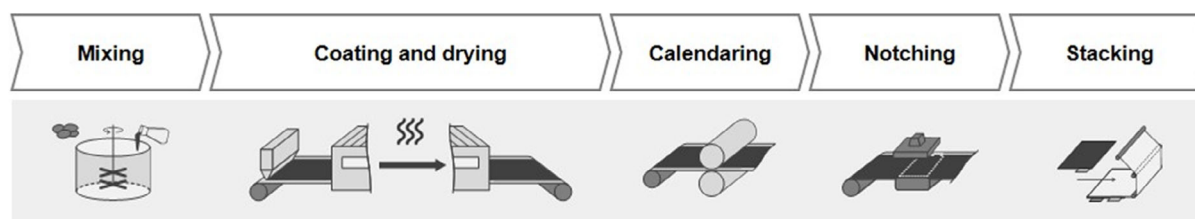


Figure 1. Schematic view of the steps in electrode production.^[30] The first step in electrode manufacturing is slurry mixing where the raw active material is mixed with binder, solvent, and other additives. Coating and drying is the process of dispersing the slurry onto the aluminum (cathode) and copper (anode) metal foils and drying the coated material to evaporate the solvents. In the calendaring step, the coated material is pressed onto the metal foil to provide the consistent thickness, porosity, and adhesion. Notching cuts the calendared electrodes into the right shape. Notching is the step in which individual electrode sheets are cut out of the electrode coil.

impurities.^[4,10] Electrode handling between process steps can also cause scratches on the electrode sheets. Lastly, the compaction of the electrode sheets by the calendaring process leads to wrinkles and electrode deformations.^[9,11,12] Manual identification of these defects on the surface of electrodes is a very time-consuming, error-prone, inefficient, expensive, and lengthy process. Deep learning approaches offer a promising methodology for automated detection of mechanical defects and deformations of the electrode.

Deep learning has been widely used in different areas such as computer vision,^[13] virtual assistants,^[14] medicine and pharmaceuticals,^[15] and retail.^[16] It^[17] is a member of machine learning methods that use multiple latent layers to discover potential underlying patterns and extract higher level features from data and recently gained adoption in the battery field.^[3] Convolutional neural networks (CNNs)^[18] are a type of artificial neural network (ANN) which can accept images as input and reduce the complex preprocessing. Due to the large, complex, and parallel computation carried out by CNN, the central processing unit (CPU) takes too long to complete the training process. However, the availability of graphical processing units (GPU) and specialized computing hardware accelerates the processing required for training the deep CNNs and established superiority on many complex computer vision tasks such as autonomous driving,^[19] and on the fly object detection^[20] in many areas. Over the last few years, many architectures of CNN have been developed, such as LaNet (1998), AlexNet (2012), GoogleNet (2014), VGG (2014), and ResNet (2015), which have significantly reduced the error in the benchmark ImageNet Large Scale Visual Recognition Challenge.^[21,22]

The challenge in defect detection in battery electrode manufacturing is that there are relatively few training examples with that one needs to teach the model a specific shape and the high speed of the electrodes rendering any human in the loop inefficient.

Deep learning-based automatic object detection algorithms have already proved their significance in many areas. Yanfen et al.^[19] proposed a vision-based system to detect various objects and to predict the intention of pedestrians for autonomous driving. Baloni et al.^[23] introduced a deep convolutional neural network (DCNN) detection method to identify hydrocephalus, a disease found in the central nervous system and requires an early-stage treatment. For pharmaceutical products, Kandpal et al.^[24] summarized the application of hyper spectroscopy,

vibrational spectroscopy (Raman, FTIR, IR),^[25] infrared,^[26] and other spectroscopy techniques. Wen et al.^[11] performed deep learning-based image analysis to identify defects (cracks and pores) in 3D metallic additive manufacturing parts. Oliveira et al.^[27] developed an entropy and image dynamic thresholding method for automatic identification and classification of the cracks (horizontal, vertical, miscellaneous, or no cracks) on the road. The authors^[28,29] reviewed the application of defect detection methods used in the fabric production industry. Therefore, in this article we studied the application of automatic defect detection techniques in the electrode manufacturing production process.

In Figure 1a schematic diagram of the electrode production workflow is shown.^[30] It is necessary to reduce or eliminate the defects such as agglomerates, bubbles, and scratches to produce good quality electrodes.^[9] Agglomerates/blisters are primarily caused by inhomogeneities in the slurry. These inhomogeneities are caused, for example, by insufficient mixing times. In addition, deviations in the slurry feeding rate during coating lead to agglomerates/blisters.^[4] Bubbles can occur, for example, due to excessive coating rates.^[9] Cracks may appear after the drying process scratches due to electrode handling between the process steps of electrode production. In the electrode production process, agglomerates are possibly formed by an insufficient mixing process. Bubbles can occur due to air inclusion within the mixing step in combination with a missing degassing step. In order to gather the data for object detection algorithm, a fast and precise Cognex camera is placed on the calendaring machine to take the pictures of the produced electrode at a certain interval.

The selected deep learning model to identify defects and draw bounding boxes around the defects along with the prediction probability and labels is YOLOv5,^[31] which is the fifth version (latest) of the popular YOLO^[32] (You Only Look Once) CNN family. The original YOLO model was developed in 2015 and was the first object detection model to draw bounding boxes around the objects and identify the class labels. As illustrated in **Figure 2**, YOLOv5 is a single-stage object detector that requires only a single pass through a neural network during training to predict almost all “similar” objects present in an image or video. YOLOv5 can detect objects with very high precision. While training the model on COCO (Common Objects in Context) dataset^[33] (a popular computer vision dataset of labeled images with 80 different objects), it obtained 0.007 s per image (≈ 140 frames per

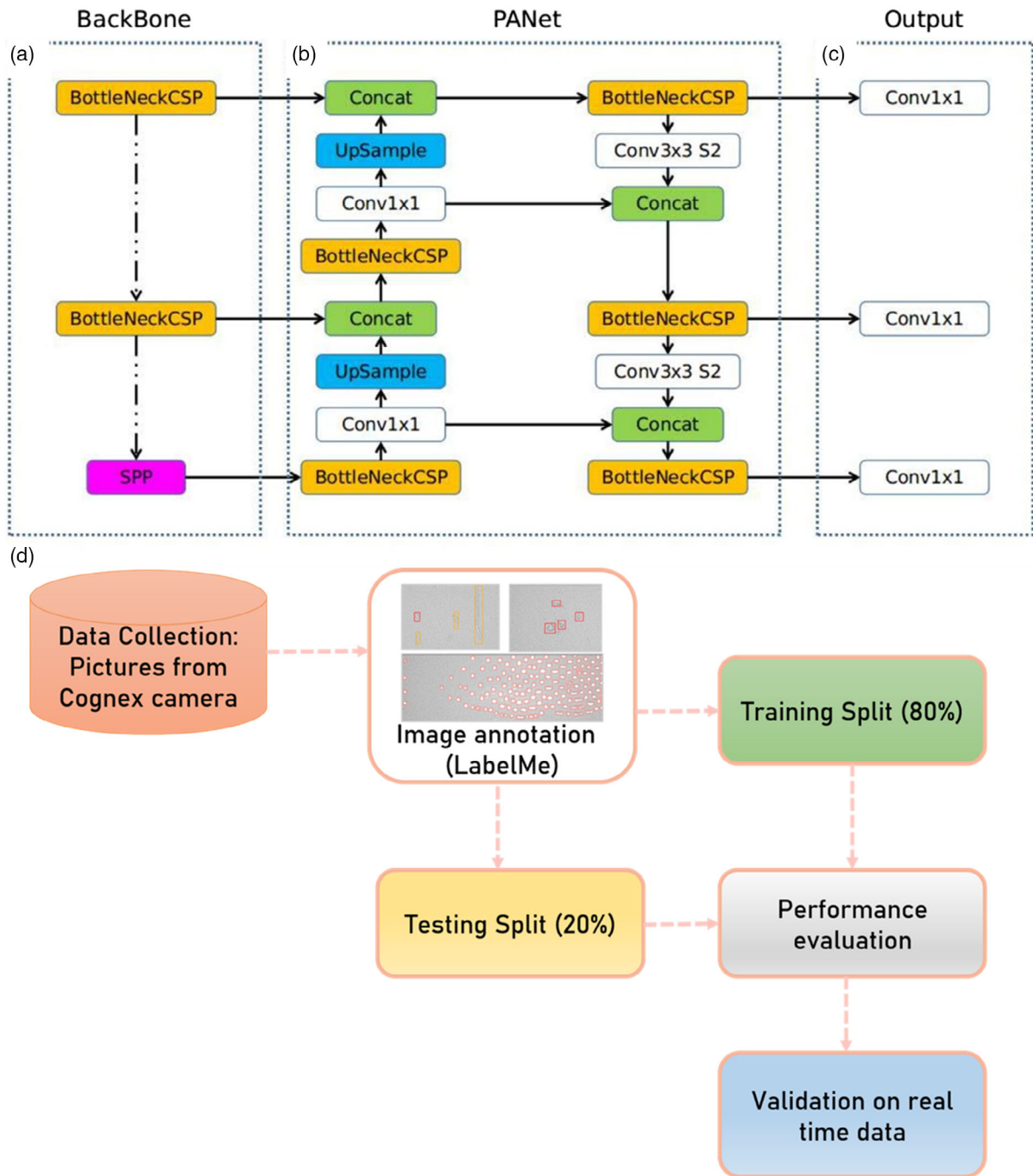


Figure 2. The architecture of YOLO^[32] is based on a) model backbone, b) model neck, and c) model head. a) Model backbone extracts the important features from the given input image. YOLOv5^[31] uses CSPNet (Cross Stage Partial Networks) as a backbone. b) Model neck generates the feature pyramids which helps the model to identify the unseen objects effectively. The common feature pyramid techniques are FPN (Feature Pyramid Network), BiFPN (Bi-directional Feature Pyramid Network), PANet (Path Aggregation Network), etc. YOLOv5 uses PANet as a neck to get a feature pyramid. c) Model head performs the final object detection part. The final output vector consists of bounding boxes, class probabilities, and object labels. d) General overview of the proposed model/workflow in operation highlighting image acquisition and the test train split as well as validation in a potential production run.

second) inference time. This makes us confident that the application of YOLOv5 for defect detection could be directly applied in real production systems. A successful implementation of this

technique, as shown in Figure 2d, could also allow for the implementation of a near real-time feedback loop to find optimal coating settings for defect-free electrodes.

2. Experimental Section

The process of defect detection is divided into three steps: 1) data collection, i.e., collecting the electrode images that include agglomerates, bubbles, foil, and scratches, 2) image annotation, i.e., manually drawing bounding boxes around selected defects and assigning of the class labels, and 3) model training on custom dataset, i.e., training the YOLOv5 model on the labeled images.

The infrastructure of eLab at RWTH Aachen University was used to generate experimental data. The machines used for anode production at PEM at RWTH University were an Eirich EL5 intensive Mixer and a coating machine from Buerkle Process Technologies which is equipped with a precision slot die unit. The coating speed during the coating trials was set to 1.0 m min^{-1} at a convection drying temperature of $100 \text{ }^\circ\text{C}$ (chamber 1) and $80 \text{ }^\circ\text{C}$ (chamber 2). The web tension is set via the winding unit with servo drive (counter roll with asynchronous drive). A Cognex Camera integrated in the Buerkle coating machine was used to take images of the dried electrode. These images were later used for training and validation.

Typical defects that occur in electrode production such as bubbles that lead to pinholes or agglomerates were introduced intentionally. Agglomerates, for example, are favored by a shortened dry mixing phase. If agglomerates were detected on the slot die during the coating trials, they have been partially removed manually. Therefore, the coating images represent an isolated example and are not representative of a commenced series production. Therefore, the coating images represent an isolated example and are not representative of a commenced series production.

A total of 882 optical photographs of electrodes were taken from the Cognex camera placed in the coating machine. **Figure 3** shows four types of defects in some images, which were identified by expert analysis: agglomerate, bubble, foil, and scratch. Some of the images contain a large number of defects, while most are defect free; others have only metal foil on both sides of the

electrode. There are a variety of paid and free image annotation softwares^[34] such as V7, Labelbox, ScaleAI, LabelMe, LabelImg, etc. We used the LabelMe^[35] tool to draw the bounding boxes around the defects and to give the class labels which are then used to train the YOLOv5 model. The labels are stored in a .yaml file with the same name and in the same directory where the images are stored. Each .yaml file contains the class label, coordinates of the bounding box, height, and width for the corresponding image.

The underlying algorithm was YOLOv5 and cloned from their GitHub repository.^[36] Data were split into two parts, 80% of the data was assigned to the training set and the remaining 20% was assigned to the validation set. As YOLOv5 is trained on the COCO dataset, a custom configuration file is generated to store the path of training data and validation data along with the number of classes and class labels. The model was trained on images of size 1280×1280 pixels, a batch size of 64 (number of images which will be processed in one batch), a subdivision of 16 (number of mini batches in one batch), and the stochastic gradient descent optimizer at a learning rate of 0.01 and a final learning rate to learn the defects of 0.1. We utilize an image scaling of 0.5 which scales the image contents while keeping the image size constant and train for a maximum of 500 epochs with early stopping after 100 epochs of no improvement. The best results were observed in epoch 174. The weights of this epoch were autosaved and used to test the performance of the model on the test data. The model was trained on 80% of the data and tested on the remaining 20% of the data. The best weights after the training were stored into the YOLO working directory to predict the defects on the new unseen data.

3. Results and Discussion

Figure 4 shows three different types of loss plots, precision, recall, and mean average precision (mAP) plots for the training

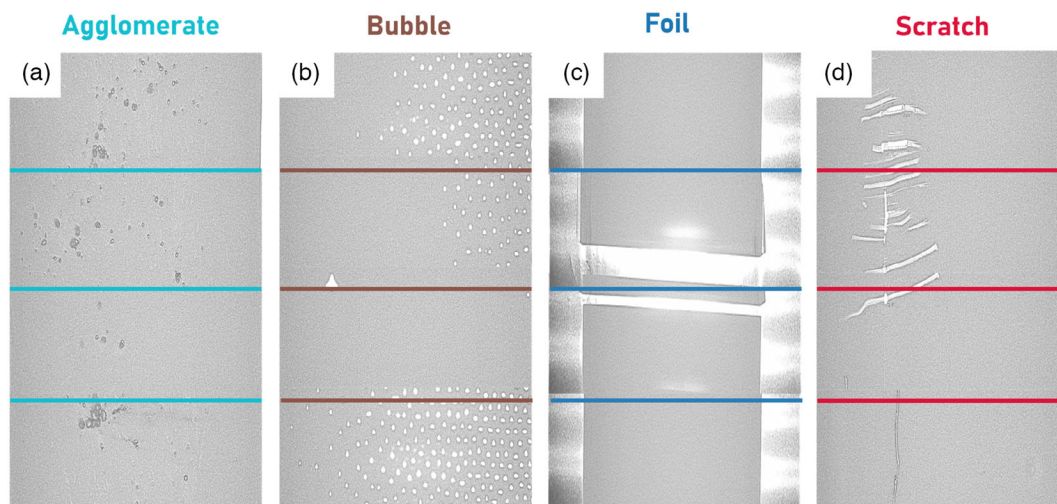


Figure 3. Exemplary images of defective electrodes captured by Cognex camera placed on the coating machine: a) a coarse accumulation of active material on the electrode surface is referred to as “agglomerate”; b) a hole on the electrode surface is called “bubble” or “pinhole”; c) the active material is coated on the metal foil and the foil is not a part of the electrode; therefore, the left and right side of the electrode is also considered as a defect and labeled as “foil.” The start and end part of the electrode also contains foil; d) a line on the electrode surface is annotated as “scratch.” The lines are created to separate the images and the colors are added to distinguish each defect class, i.e., each category has four images.

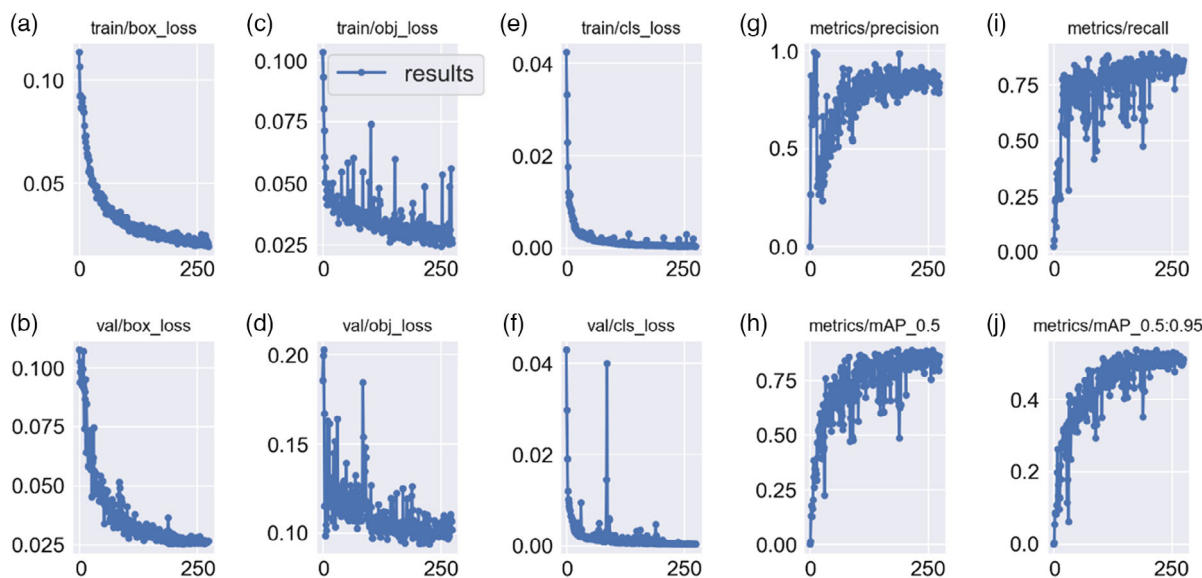


Figure 4. The performance of the YOLOv5 model trained on the custom data: a), b) the box-loss plot indicates the performance of the model in terms of locating the bounding box around the defect; c), d) the object-loss plot is significantly a measure of the likelihood that a defect is present in the predicted bounding box; e), f) the cls-loss plot represents how efficiently the model can predict the class label of a defect; g) precision is the ratio of accurately predicted labels to the all predicted labels; i) recall is the ratio of accurately predicted labels to all the correct labels; h), j) the mAP is a measure to evaluate the performance of the object detection models in terms of actual and predicted bounding box differences and represents aggregate averaging.

and validation set generated during the training of the YOLOv5 model. The box-loss plot indicates the performance of the model in terms of locating the bounding box around the defect. This measure examines the potential competency of the model to predict the center of the bounding boxes and determines how well the bounding box covers the defect. In the beginning when the training started, the box-loss was 0.11 and it was reduced to 0.01 after training (after around 280 epochs) for the training set. The loss for validations set was also reduced by around 80% (from 0.10 to 0.02) after 280 steps. The object-loss plot is significantly a measure of the likelihood that a defect is present in the predicted bounding box. If the object-loss is low, it means that the predicted bounding box contains a defect with higher probability. The recorded value of object-loss after 280 epochs was 0.02 and 0.09 with a decrease by around 76% and 49% for the training and validation sets. The cls-loss plot represents how efficiently the model can predict the class label of a defect. The model performs pretty well in terms of predicting the correct class labels. The model achieved 0.00036 and 0.00033 cls-loss for training and validation sets, respectively, which is considerably less.

The precision and recall are relevance-based performance measures for a supervised learning model. We believe the accuracy of the model not being primarily dependent on the material used, as any changes thereof would simply need sufficient correctly labeled objects (defects) upon changes as most defect originate from manufacturing process imperfections, which we believe to be universal to all materials. Precision is the ratio of accurately predicted labels to the all predicted labels, while recall is the ratio of accurately predicted labels to all the correct labels. The class averaged precision and recall of the model are 88% and 84%, respectively, which indicates good performance

(see Figure S1–S3, Supporting Information, for F1 scores for the different classes and the precision and recall values for all classes). mAP is a measure to evaluate the accuracy of object detection models. It compares the actual bounding box to the predicted one and generates a score. The model achieved 88% mAP for all the classes combined; therefore, it is significantly precise in its detections. The mAP for individual classes is: 65% for agglomerates, 98% for bubbles, 99% for foil, and 90% for scratches. As there were nearly 300 agglomerates in the data, we see a great potential to improve the performance of the model by training with an additional number of instances of agglomerate class. The mAP of the model with different intersection over union (IoU) thresholds from 0.5 to 0.95 is denoted as mAP_{0.5:0.95}.

A confusion matrix^[37] is a $N \times N$ performance table for the classification models that compares the predicted responses with the actual responses. For object detection and instant segmentation applications, the confusion matrix is less intuitive. Intersection over Union (IoU) plays an important role to calculate confusion matrix for object detection tasks. IoU is also known as Jaccard index and is a measure to evaluate the extent of overlap between two bounding boxes or masks. The ground truth bounding box coordinates (BB_g) and predicted bounding box coordinates (BB_p) are required to compute the IoU.

$$IoU = \frac{\text{area}(BB_g \cap BB_p)}{\text{area}(BB_g \cup BB_p)} \quad (1)$$

IoU is computed by the ratio of overlapping area to the union area between the ground truth bounding box and predicted bounding box. The number of TP, FP, and false negatives (FNs) are determined by the IoU score. The confusion matrix

can be generated at different IoU thresholds. The results in **Figure 5** are output at 0.45 IoU threshold (default threshold value to get the maximum accuracy). TP is an outcome of correct detection made by the model where the actual annotation and predicted bounding box locate the same defect (detection with $\text{IoU} \geq \text{threshold}$). FP is a result of wrong detection where the ground truth annotation and predicted bounding box do not match. FN is a false alarm, i.e., there is a defect but the model cannot detect it. For the defect detection task, we compute the confusion matrix for each image. The diagonal entries in the confusion matrix represent a TP, and the last row and last column represent the FN and FP, respectively. Once the confusion matrix for each image is computed, we can obtain the number of TP, FP, and FN for each class to get the accurate insights about the model. The images can be sorted in the descending order on the basis of FP or FN values to detect the images where the performance of the model is not adequate. For example, if the images are not correctly labeled, then active learning can be a potential approach to label the images. The values of TP for agglomerate, bubble, foil, and scratch are 0.86, 0.98, 0.99, and 0.90. Additionally, the values of FN for each class are very low which indicate that the model is able to accurately identify the defects. The model has slightly high values of FP for agglomerate, bubble, and scratch class. As our dataset contains lesser labeled instances of these classes as compared to the foil class,

the FP can be further reduced by obtaining and labeling more data for these classes.

After training the model on 80% of the data, it is tested on the remaining 20% of the data to check if the model can predict the defects correctly. The model was then also trained on 70% of the data and tested on the remaining 30% of the data, with no significant degradation in performance. **Figure 6** shows the potential competency of the model to predict the different types of defects. The model achieved 9.5 ms inference time during the prediction. The model is able to predict bubbles (pink bounding boxes), agglomerates (red bounding boxes), foil (orange bounding boxes), and scratches (yellow bounding boxes). As the figure shows, the model is able to predict almost all bubbles and foil. Some instances of agglomerates and scratches are not predicted by the model considering the fact that we do not have the sufficient instances of these defects to train the model.

3.1. Spatial Distribution

This article demonstrates the application of defect detection algorithms as a preventive measure in the electrode production line. The defects are correlated with the input parameters such as type of active material, mixing time, coating speed, slit width, etc. The proposed model can identify the spatial location of the defects on the electrode surface which can be useful to establish

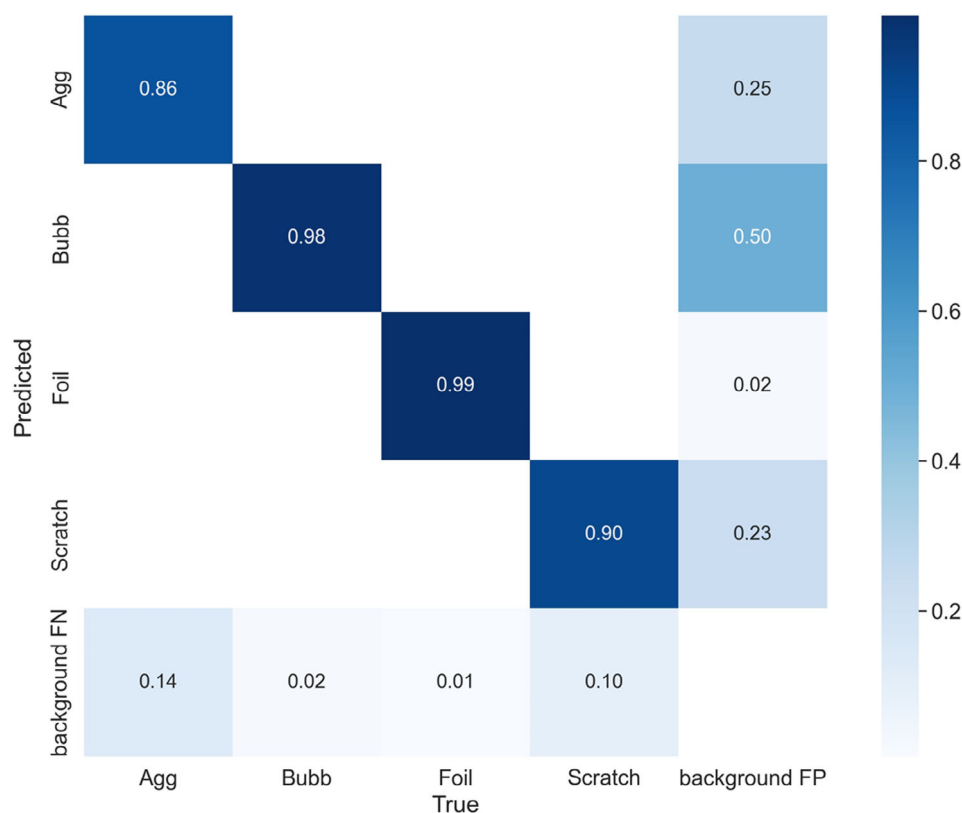


Figure 5. The confusion matrix showing the actual and predicted class labels for defects with the true positive (TP) rate on the diagonal and the last column showing the false positive (FP) rate. Foil class achieved the height precision and recall. Bubble class obtained higher recall as compared to precision. The model performed almost consistently while predicting the agglomerates and scratches. The model achieved a precision of 88%, a recall of 84%, and a 75% total accuracy.

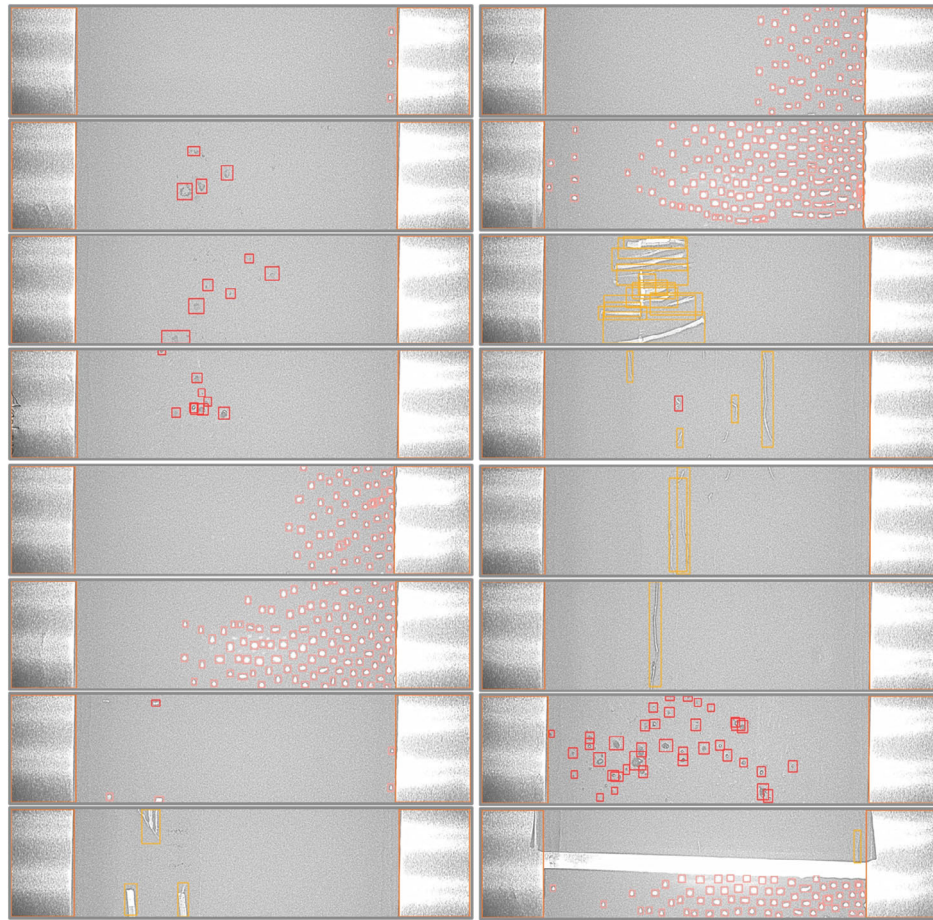


Figure 6. A batch of defected electrode images from the test dataset. The model predicts the bounding boxes around different types of defects that indicate the spatial location of the defects on the electrode surface along with the confidence score. The score demonstrates the possibility that the predicted bounding box contains a defect and what is the accuracy of the bounding box. The colors indicate different classes of defects. The red color stands for agglomerates, the pink color for bubbles, the yellow color for scratches, and the orange color for foils. All the predicted bounding boxes have a confidence score greater than 0.25 (default confidence threshold for YOLOv5).

the relation between input parameters and defects. The whole data (882 images) are fed into the model to identify the total number of defects present on the electrode surface on the horizontal and vertical axis. Each input image has an original height of 500 pixels and a width of 2448 pixels. The total number of defects present on the horizontal axis is shown in **Figure 7**. The active material is coated on the aluminum or copper foil to produce the positive and negative electrodes. The point at which the coating contacts the foil can be easily determined by **Figure 7**. About the first and last 350 pixels on the horizontal axis of the image are identified as foil and therefore the remaining area is covered with active material. The number of defects of foil class are equal to the number of images in the dataset as each image consists of an underneath foil and it can be validated by **Figure 7**. The defects are not equally distributed on the electrode surface and more defects appear on the right side of the images which may be due to the fact that the slit width is not perfectly aligned. This correlation between the defects and input parameters can be utilized as a preventive measure to reduce the defects during the production of the next electrode. **Figure 7** exhibits the

distribution of defects on the vertical axis, which specifies that the defects are uniformly distributed along the axis. From **Figure 7**, it can be concluded that there are more defects in the right part of the electrode surface.

As each plausible defect category can be generated by different input parameters chosen during the electrode coating process, we determined the number of defects per category along the horizontal and vertical axes to study the effects of these parameters. **Figure 8** represents these defects along the horizontal axis. The agglomerates and scratches appear in the central part of the electrode and have a Gaussian distribution. Agglomerates can occur if the raw material is not adequately mixed and the produced slurry is not homogeneous. There are only a few instances of agglomerates and scratches and these can be reduced by adjusting the input parameters. The instances of the foil class correspond to the number of images in the dataset. It can be seen from the figure that the model can detect almost all the foils on both sides of the electrode, while it does not detect any foils in the middle part of the image. The electrode has a considerable number of bubbles, which increase gradually as

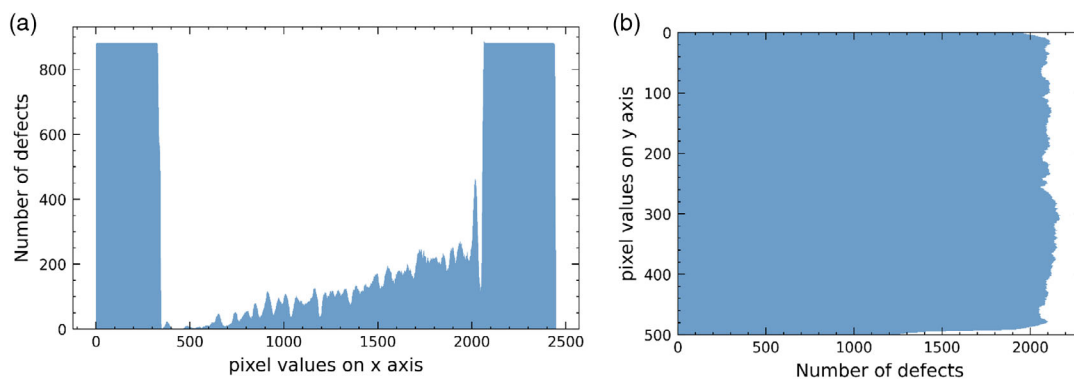


Figure 7. The distribution of the defects present on the electrode surface. a) The count of defects along the horizontal axis where x axis represents the pixel values and y axis represents the number of defects. In the horizontal direction, the left and right parts of the image consist of foil. As the dataset consists of 882 images, the number of defects on the left and right sides is almost 882. There are more defects appearing in the right side of the electrode; b) the count of defects along the vertical axis where x axis represents the number of defects and y axis represents the pixel values. In the vertical direction, the top and bottom parts of the image consist of foil. As there are two foils along the vertical axis and the dataset consists of 882 images, there are almost 1764 of the total defects due to foil. The defects are almost perfectly aligned along the axis.

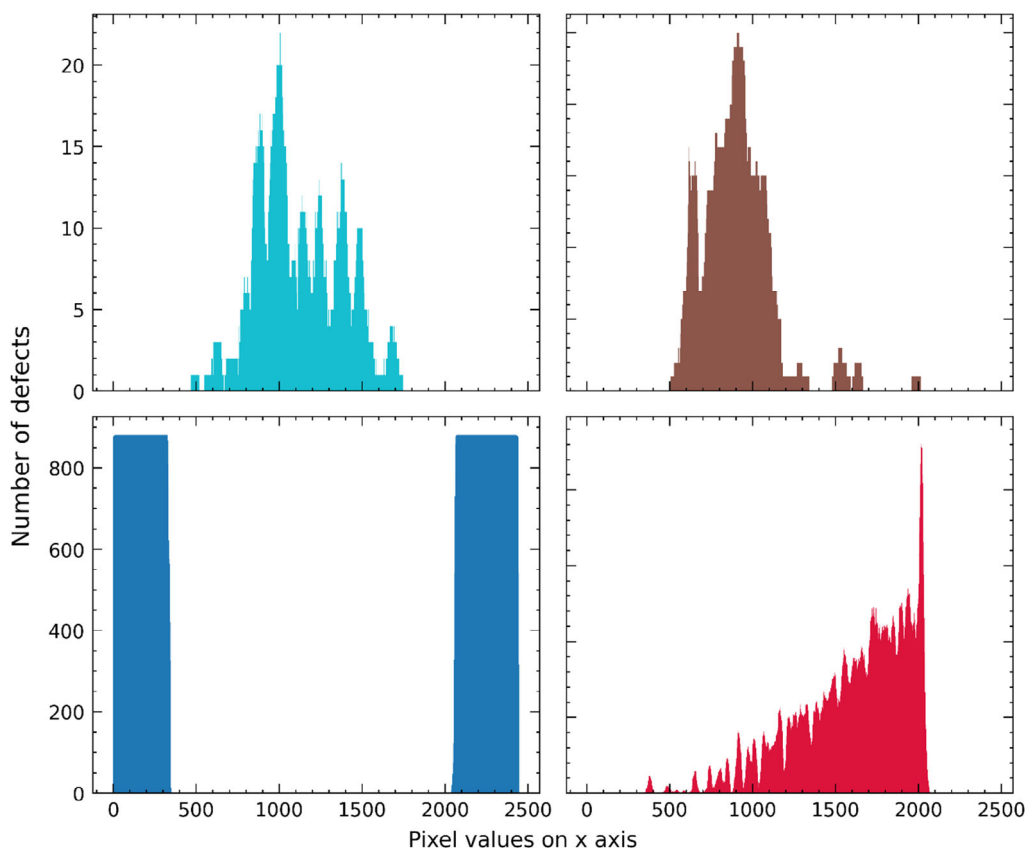


Figure 8. The per category defects on the horizontal axis: a) scratch; b) agglomerate; c) foil; and d) bubble. Horizontal axis represents the pixel value along the axis and the vertical axis represents the number of defects per category. The agglomerates and scratches are distributed in a nearly Gaussian shape and occur in the central part of the electrode. These types of defects rarely occur in the entire dataset. Since the dataset contains 882 images, the number of instances of the foil class is almost 882 on the left and right sides of the images. The number of bubbles increases from left to right, and there are a large number of instances of this particular class on the electrode.

we go from left to right. Bubbles form where the coating of the electrode is missing and can be caused by gas inclusion during the mixing process. There are a lot of bubbles near the foil on the right side.

Figure 9 illustrates the defects for each class on the vertical axis. Similar to the horizontal axis, there are few scratches and agglomerates on the vertical axis. The number of instances of the foil class is almost twice the number of images in the

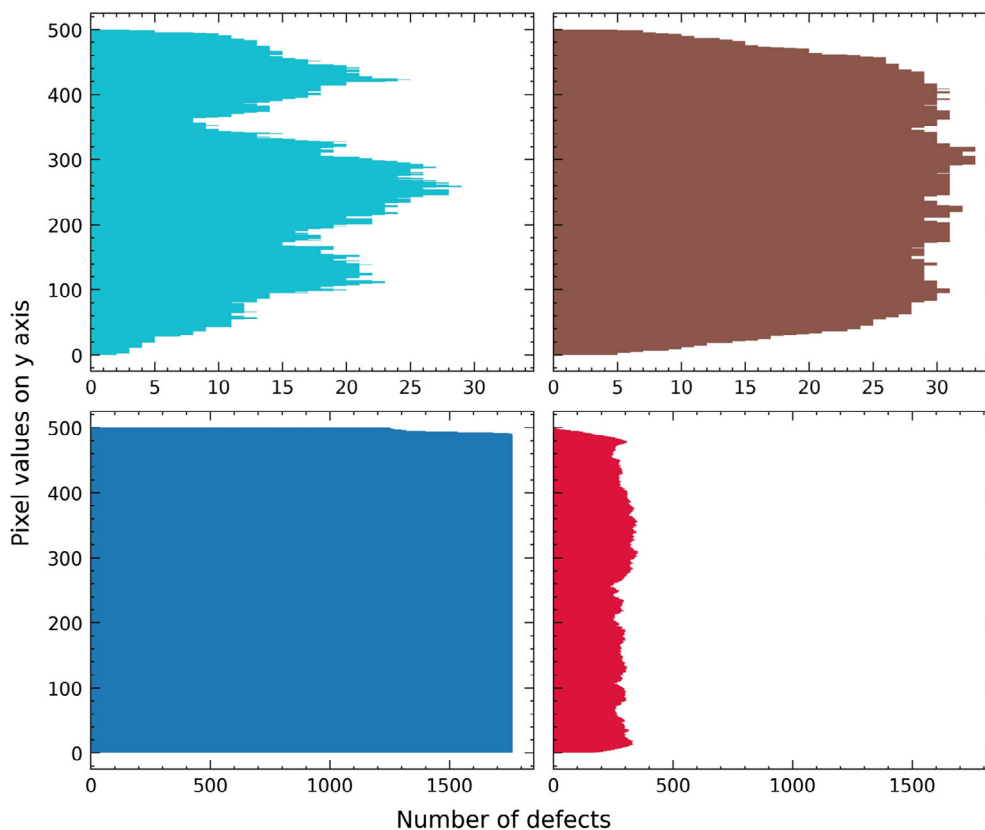


Figure 9. The per category defects on the vertical axis: a) scratch; b) agglomerate; c) foil; and d) bubble. Horizontal axis represents the number of defects per category and the vertical axis represents the pixel value along the axis. There are fewer scratches and agglomerates compared to bubbles. As the dataset consists of 882 images, there are ≈ 1764 instances of the foil category. The bubbles are almost uniformly distributed along the axis.

dataset. The electrode has a large number of bubbles, but they are evenly distributed over the entire electrode.

The information about the size and shape of the defect can be helpful for physics-based simulations as the electrochemical performance of LiB is highly correlated to the quality of the used electrodes. If the electrodes are not defective, the battery is expected to have a longer life as well as more capacity. However, in the case of bigger and porous defects, the foil located under the active material comes into direct contact with the electrolyte, and the battery may suffer a significant loss of electrical properties. The defects on the basis of bounding boxes are cropped out from the original electrode images and then stored in a separate directory. The model detected 283 agglomerates, 5235 bubbles, 1765 foil, and 305 scratches. The shape of the defects is determined with one of the most popular and widely used canny edge detection algorithms. Edge detection is noise sensitive, so the initial step is to filter out the background noise in the image by applying a 5×5 Gaussian filter. The filtered image is then processed with a Sobel kernel in the horizontal and vertical directions to determine the intensity gradient of the image. After determining the magnitude and direction of the gradient, all unwanted pixels in the image that potentially do not belong to the edge are removed by nonmaximum suppression. The hysteresis threshold is used to find the edges. Two threshold values are set for this purpose, `min_threshold` and `max_threshold`. The edges with intensity gradient

greater than `max_threshold` are classified as edges, while those with intensity gradient less than `min_threshold` are classified as nonedges and will be rejected. All edges whose intensity gradient is in between these two thresholds are categorized either as edges or nonedges depending on their connectivity. They are considered as part of the edges if they are linked to any pixels with “edges.” Otherwise, they are discarded as well. As there is a significant amount of background noise, we manually adjust the threshold for each image to identify the edges that best resemble the shape of the defect. **Figure 10** shows the cropped defects from the original image, their shape obtained with edges, and their respective sizes plotted on a polar chart. The detected edge image is represented in an array, and for each row the distance between the edges is then calculated from subtracting the first pixel value from the last pixel value if there is a one in the array. The average value of the difference is plotted on the polar diagram. The image is rotated 360° clockwise with step size 1 and the average distance is calculated and plotted for each rotation. The defect size is indicated by the radial axis labels.

4. Conclusion

We propose the utilization of YOLOv5, a Deep Learning-based object detection framework, for detection of defects early in

electrode production process. This study can be used as a preventive measure to improve electrode and manufacturing quality. This model has been trained on only a few examples and yet shows promising performance that outperforms any possible human in the loop in terms of speed and accuracy. The model

can precisely predict each defect class in static (image) and dynamic (video) datasets and achieved 88% mAP for all the classes combined. As there was less training data, we see expect to improve the performance of the model even more by training with additional training data.

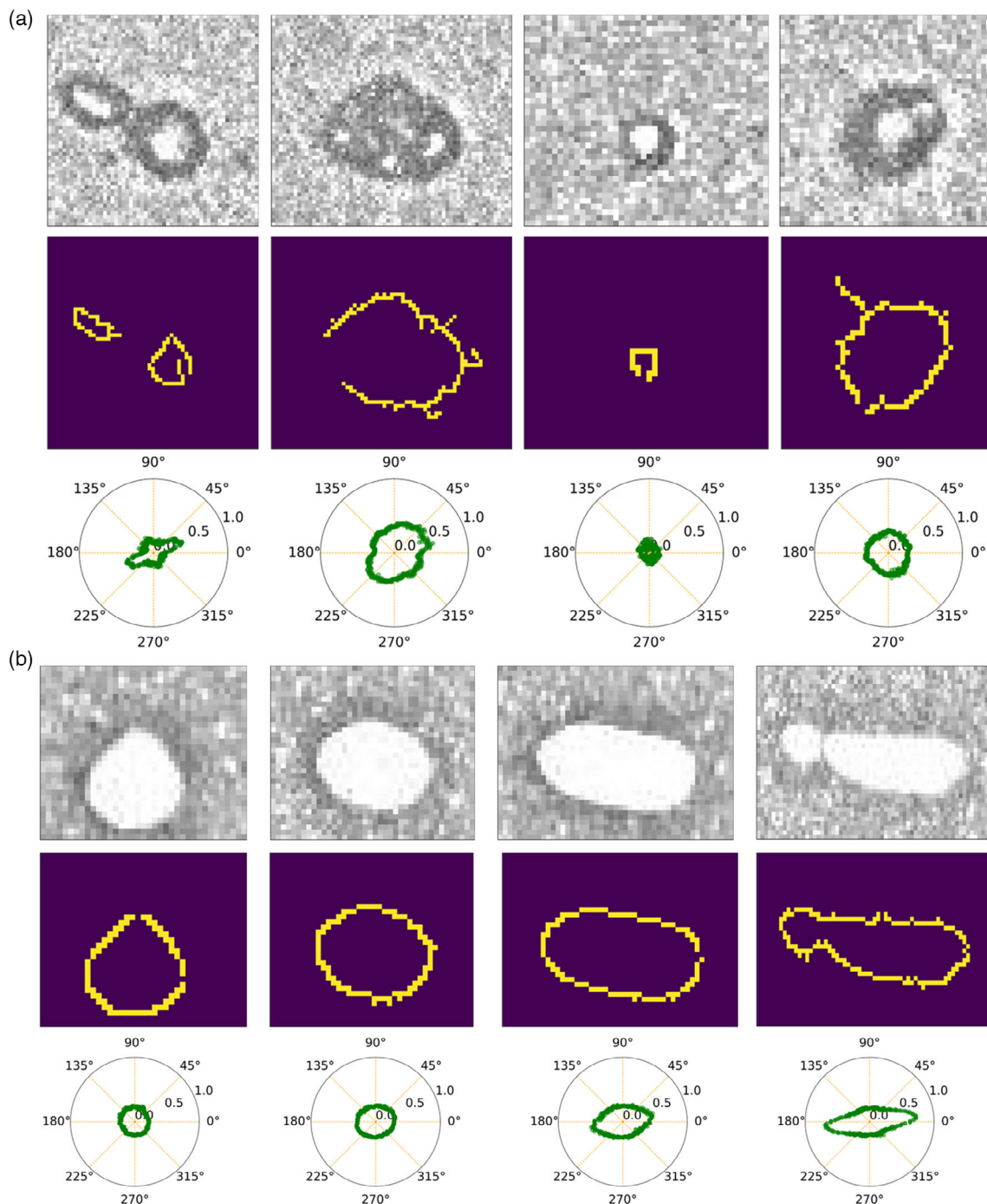


Figure 10. The defects of each class together with their shape and size: a) agglomerate; b) bubble; c) foil; d) scratch. The images in the first row are the actual defects on the electrode, cropped by the model from the original electrode images based on the bounding box coordinates. The second row shows the edges detected by the Canny Edge Detection algorithm to represent the shape of the defect. The last row shows the size of the defect in a polar diagram while rotating the defect by 360°.

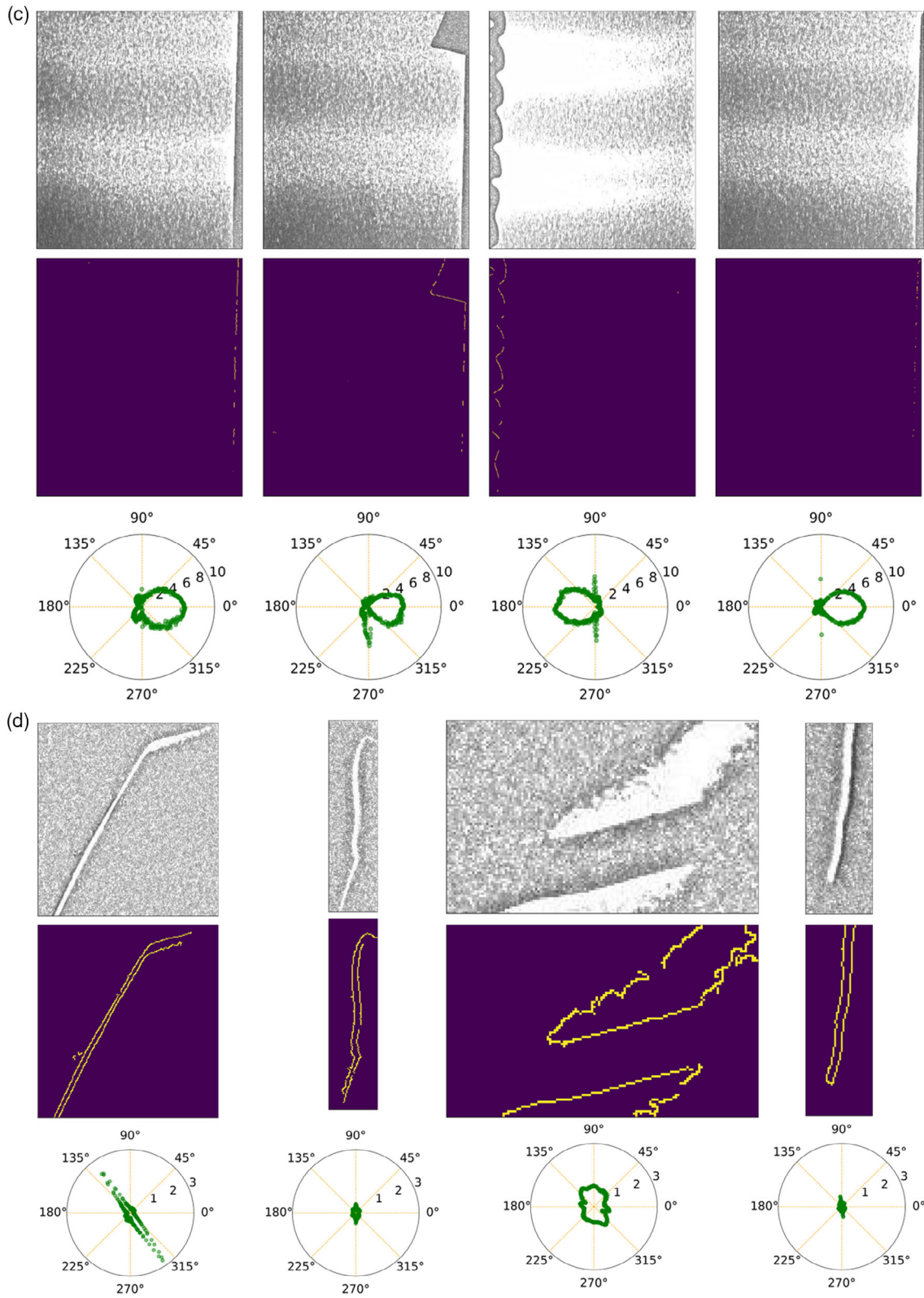


Figure 10. Continued.

The model is capable of detecting defects at a web speed of 9 m s^{-1} , thus reducing the ramp-up times and allowing closed-loop optimization with exact defect localization. The model achieved an inference time of 9.5 ms in prediction while being trained on 80% of the data and tested on the remaining 20% of the data. Therefore, it is possible to adjust the parameters of the coating process during the current production to reduce the defects generated in the previous batch. Active learning would be a potential candidate to annotate a larger number of images in real production systems to improve the performance of the model even more.

Concluding, the utilization of fast and accurate, surrogate trained, models for defect detection in a battery production line is demonstrated and is expected to reduce the ramp-up time significantly and enable faster mitigation of defect inducing system states.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

This work contributes to the research performed at CELEST (Center for Electrochemical Energy Storage Ulm-Karlsruhe) and was funded by the German Research Foundation (DFG) under Project ID 390874152 (POLiS Cluster of Excellence). This project received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 957189. This project received funding from BMBF in the framework of the BMBF-Kompetenzcluster InZePro for the projects DataBatt No. 03XP0323D and InForm No. 03XP0363A.

Open Access funding enabled and organized by Projekt DEAL.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The trained model and code for training is supplied with this manuscript available at <https://github.com/Helge-Stein-Group/defect-detection> but the actual training data is not shared.

Keywords

battery, coating, machine-learning, YOLO

Received: May 30, 2022

Revised: August 20, 2022

Published online:

- [1] D. Michaelis, E. Rahimzei, P. Kampker, H. Heimes, C. Offermanns, M. Locke, H. Löbbberding, S. Wennemar, A. Thielmann, D. Hettesheimer, C. Neef, A. Kwade, W. Haselrieder, S. Blömeke, S. Doose, N. von Drachenfels, R. Drees, A. Fröhlich, L. Gottschalk, Z. Huang, Roadmap Batterie-Produktionsmittel 2030 - Update 2020, Frankfurt am Main, Germany **2021**.
[2] A. Masias, J. Marcicki, W. A. Paxton, *ACS Energy Lett.* **2021**, 6, 621.

- [3] X. Chen, L. Ye, Y. Wang, X. Li, *Adv. Intell. Syst.* **2019**, 1, 1900102.
[4] D. Mohanty, E. Hockaday, J. Li, D. K. Hensley, C. Daniel, D. L. Wood, *J. Power Sources* **2016**, 312, 70.
[5] E. B. Guttoff, E. D. Cohen, in *Coating and Drying Defects: Troubleshooting Operating Problems*, John Wiley & Sons, Hoboken, NJ **2006**.
[6] A. Gitis, A. Kwade, D. U. Sauer, *Flaw Detection in the Coating Process of Lithium-Ion Battery Electrodes With Acoustic Guided Waves*, Institut für Stromrichtertechnik und Elektrische Antriebe ISEA, Aachen, Germany **2017**.
[7] F. Duffner, L. Mauler, M. Wentker, J. Leker, M. Winter, *Int. J. Prod. Econ.* **2021**, 232, 107982.
[8] A. Sakti, J. J. Michalek, E. R. H. Fuchs, J. F. Whitacre, *J. Power Sources* **2015**, 273, 966.
[9] C. D. Reynolds, P. R. Slater, S. D. Hare, M. J. H. Simmons, E. Kendrick, *Mater. Des.* **2021**, 209, 109971.
[10] O. Badmos, A. Kopp, T. Bernthaler, G. Schneider, *J. Intell. Manuf.* **2020**, 31, 885.
[11] H. Wen, C. Huang, S. Guo, *Materials* **2021**, 14, 2575.
[12] D. Mayer, A.-K. Wurba, B. Bold, J. Bernecker, A. Smith, J. Fleischer, *Processes* **2021**, 9, 2009.
[13] D. A. Wood, S. Kafabadi, A. Al Busaidi, E. L. Guilhem, J. Lynch, M. K. Townsend, A. Montvila, M. Kiik, J. Siddiqui, N. Gadapa, M. D. Bengier, A. Mazumder, G. Barker, S. Ourselin, J. H. Cole, T. C. Booth, *Eur. Radiol.* **2022**, 32, 725.
[14] R. Vishnu, N. K. Prakash, in *Soft Computing and Signal Processing* (Eds: V. S. Reddy, V. K. Prasad, J. Wang, K. T. V. Reddy), Springer, Singapore **2022**, pp. 609–617.
[15] X. Pan, X. Lin, D. Cao, X. Zeng, P. S. Yu, L. He, R. Nussinov, F. Cheng, *WIREs Comput. Mol. Sci.* **2022**, 12, 1597.
[16] S. Dutta, *WIREs Data Min. Knowl. Discov.* **2018**, 8, 1257.
[17] Y. LeCun, Y. Bengio, G. Hinton, *Nature* **2015**, 521, 436.
[18] Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou, in *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, IEEE, Piscataway, NJ pp.1–21.
[19] Y. Li, H. Wang, L. M. Dang, T. N. Nguyen, D. Han, A. Lee, I. Jang, H. Moon, *IEEE Access* **2020**, 8, 194228.
[20] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, B. Lee, *Digit. Signal Process.* **2022**, 126, 103514.
[21] A. Géron, in *Hands-on Machine Learning With Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media, Inc., Sebastopol, CA **2019**.
[22] S. K. Vasudevan, S. R. Pulari, S. Vasudevan, in *Deep Learning*, Chapman and Hall/CRC: New York **2021**.
[23] D. Baloni, S. Verma, *Multimed. Tools Appl.* **2022**, 81, 16171.
[24] L. M. Kandpal, E. Park, J. Tewari, B.-K. Cho, *J. Biosyst. Eng.* **2015**, 40, 394.
[25] M. Amar, I. Gondal, C. Wilson, *IEEE Trans. Ind. Electron.* **2015**, 62, 494.
[26] P. Li, I. Dolado, F. J. Alfaro-Mozaz, F. Casanova, L. E. Hueso, S. Liu, J. H. Edgar, A. Y. Nikitin, S. Vélez, R. Hillenbrand, *Science* **2018**, 359, 892.
[27] H. Oliveira, P. L. Correia, in *2009 17th European Signal Processing Conf.*, Glasgow, Scotland **2009**, pp. 622–626.
[28] H. Y. T. Ngan, G. K. H. Pang, N. H. C. Yung, *Image Vis. Comput.* **2011**, 29, 442.
[29] P. Mahajan, S. Kolhe, P. Patil, J. Mahajan, *Adv. Comput. Res.* **2009**, 1, 18.
[30] W. B. Hawley, J. Li, *J. Energy Storage* **2019**, 25, 100862.
[31] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, T. Xie, J. Fang, imyhxy, K. Michael, Lorna, A. V, D. Montes, J. Nadar, Laughing, tkianai, yxNONG, P. Skalski, Z. Wang, A. Hogan, C. Fati, L. Mamma, AlexWang1900, D. Patel, D. Yiwei, F. You, J. Hajek, L. Diaconu, M. T. Minh, Ultralytics/Yolov5: V6.1 – TensorRT, TensorFlow Edge TPU and

- OpenVINO Export and Inference, Zenodo, **2022**. <https://doi.org/10.5281/zenodo.6222936>.
- [32] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, ArXiv150602640 Cs, **2016**.
- [33] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, P. Dollár, Microsoft COCO: Common Objects in Context, ArXiv14050312 Cs **2015**.
- [34] Y. Chen, X. Zeng, X. Chen, W. Guo, *Appl. Intell.* **2020**, *50*, 3412.
- [35] B. C. Russell, A. Torralba, K. P. Murphy, W. T. Freeman, *Int. J. Comput. Vis.* **2008**, *77*, 157.
- [36] Ultralytics/Yolov5, Ultralytics **2022**.
- [37] R. Padilla, S. L. Netto, E. A. B. da Silva, in *Int. Conf. on Systems, Signals and Image Processing (IWSSIP)*, Rio de Janeiro, Brazil **2020**, pp. 237–242.