Autonomous visual detection of defects from battery electrode manufacturing

Nirmal Choudhary[1,2], Henning Clever[3], Robert Ludwigs[3], Michael Rath[4], Aymen Gannouni[4], Arno Schmetz[6], Tom Hülsmann[6], Julia Sawodny[5], Leon Fischer[1,2], Achim Kampker[3], Juergen Fleischer[5], Helge S. Stein*[,1,2]

***correspondence should be addressed to: helge.stein@kit.edu**

1: Helmholtz Institute Ulm (HIU), Helmholtzstr. 11, 89081 Ulm, Germany

2: Karlsruhe Institute of Technology (KIT), Institute of Physical Chemistry (IPC), Fritz-HaberWeg 2, 76131 Karlsruhe, Germany

3: Chair of Production Engineering of E-Mobility Components (PEM), RWTH Aachen University, Bohr 12, 52072 Aachen, Germany

4: Information Management in Mechanical Engineering (IMA), RWTH Aachen University, Dennewartstr. 27, 52068 Aachen, Germany

5: wbk Institute of Production Science, Karlsruhe Institute of Technology, Kaiserstrasse 12, 76131 Karlsruhe, Germany

7: Fraunhofer Research Institution for Battery Cell Production FFB, Mendelstraße 11, 48149 Münster, Germany

Abstract

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, 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 here 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.

Introduction

The demand for electrical energy is subject to continuous 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[10]. In this term reliable production facilities that have appropriate quality assurance systems are particularly desirable. Within this work, the process for manufacturing electrode sheets for lithium-ion battery 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. 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 first production steps have far-reaching effects on the battery performance and its lifetime[4] and are thus quality-determining. In order to avoid cascading effects in the further production steps, early-stage defect identification is extremely important to improve the electrode and manufacturing quality[2,3]. Since the coating and calendering process account for approx. 22% of the total cost of electrode production, there is great potential for cost reduction in optimizing these processes[5,6,8]. In particular, the coating and calendering 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 impurities[4,11]. Electrode handling between process steps can also cause scratches on the electrode sheets. Last, the compaction of the electrode sheets by the calendering process leads to wrinkles and electrode deformations[7,8,12]. Manual identification of these defects on the surface of electrodes is a very time-consuming, error-prone, inefficient, expensive and lengthy process[9]. 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. Convolutional neural networks[18] (CNN) 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 a long time 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 establishes the superiority on many complex computer vision tasks such as autonomous driving[19], and on the fly object detection[20] in many areas. Over the past 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.[7] 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 paper we studied the application of automatic defect detection techniques in the electrode manufacturing production process.

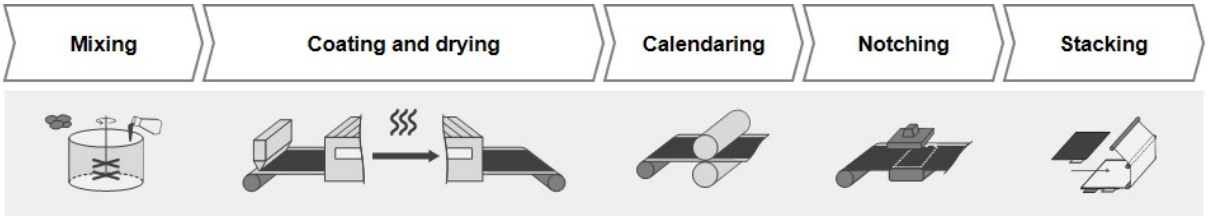


Figure 1: Schematic view of the steps in electrode production[30]. The first step in electrode manufacturing is slurry mixing where the raw active material is mixed with binder, solvent, and other additives. Coating and drying is the process of dispersing the slurry onto the aluminium (cathode) and copper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

In Figure 1 a 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[8]. Agglomerates/blisters are primarily caused by inhomogeneities in the slurry. *These inhomogeneities are caused, for example, by insufficient mixing times. In addition, agglomerates may form due to insufficient mixing of the powdered active material. Bubbles can be caused by air inclusions in the mixing process in combination with a missing degassing step. Furthermore, deviations caused e.g. due to excessive coating rates[8] or insufficient slurry feed rates during coating lead to bubbles and missing spots. Cracks may appear after the drying process due to the handling of the electrodes and the winding and unwinding between the process steps of the electrode production.*

In order to gather the data for object detection algorithm, a fast and precise Cognex camera is placed on the calendering machine to take the pictures of the produced electrode at a certain interval.

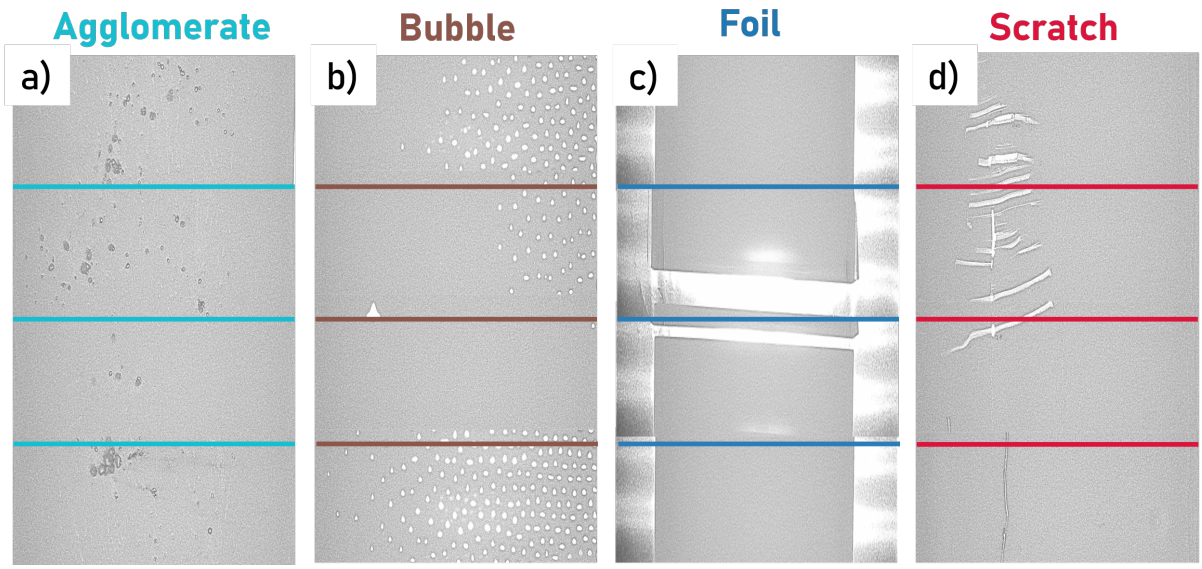


Figure 2: Exemplary images of defective electrodes captured by Cognex camera placed on the coating machine: a) a coarse accumulation of active material on the electrode surface is referred to as "agglomerate"; b) a hole on the electrode surface is called "bubble" or "pinhole", c) the active material 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 labelled 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 4 images.

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 3, 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 labelled images with 80 different objects), it obtained 0.007 seconds per image (~140 frames per 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 could also allow for the implementation of a near real time feedback loop to find optimal coating settings for defect free electrodes.

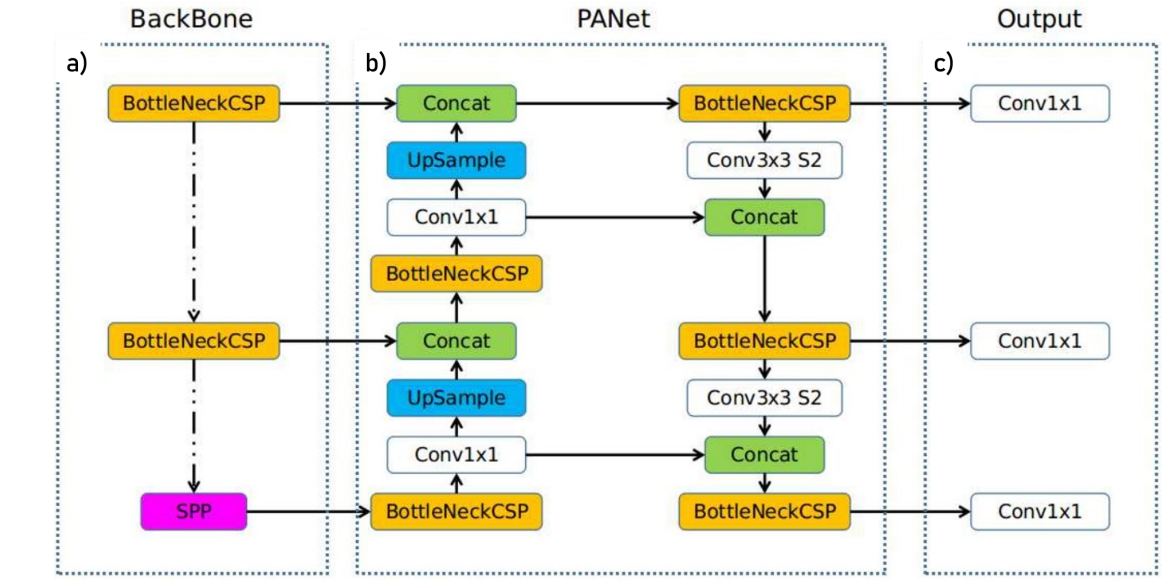


Figure 3: The architecture of YOLO[32] is based on a) model backbone, b) model neck and c) model head. a) Model backbone extracts the important features from the given input image. YOLOv5[31] uses CSPNet (Cross Stage Partial Networks) as a backbone. b) Model neck generates the feature pyramids which 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.

Methods

The process of defect detection is divided into three steps: (i) data collection i.e. (collecting the electrode images that include agglomerates, bubbles, foil, and scratches, (ii) image annotation i.e. (manually drawing bounding boxes around selected defects and assigning of the class labels, (iii) model training. on custom dataset (train the YOLOv5 model on the labelled 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 at a convection drying temperature of 100°C (chamber 1) and 80°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 favoured 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 an commenced series production.

A total of 882 optical photographs of electrodes were taken from the Cognex camera placed on the coating machine. Figure 2 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, Lebelbox, 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 YOLOv5 GitHub repository[36] was cloned to the local computer equipped with Windows 10 Pro operating system, CPU AMD Ryzen Threadripper PRO 3975WX 32-Cores @3.50 GHz, GPU NVIDIA RTX A6000 and 128 GB RAM, CUDA 11.2 and CUDNN 8.1.1 to accelerate the GPU computing. All the dependencies were installed from the requirements.txt file. For the training purpose, a train_data folder was created where all the images with their respective labels were stored. The data is splitted 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 the following parameters: image_size=1280, batch_size=64 (number of images which will be processed in one batch), subdivision=16 (number of mini batches in one batch), optimizer=SGD (stochastic gradient descent), lr0=0.01 (initial learning rate), lrf=0.1 (final OneCycleLR learning rate), scale=0.5 (scales the image contents while keeping the image size constant), epochs=500 (number of iterations). The best weights observed in epoch 174, the training process stopped after the 274th epoch as there was no improvement in the last 100 epochs. The best weights were autosaved in the runs folder inside the yolo working directory for further validation. These weights were 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.

Results & Discussion

Figure 4 shows three different types of loss plots, precision, recall, and mean average precision (mAP) plots for the training 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 obj-loss plot is significantly a measure of the likelihood that a defect is present in the predicted bounding box. If the obj-loss is low, it means that the predicted bounding box contains a defect with higher probability. The recorded value of obj-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. 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 precision and recall of the model is 88% and 84% respectively which indicates the good performance. Mean average precision (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.

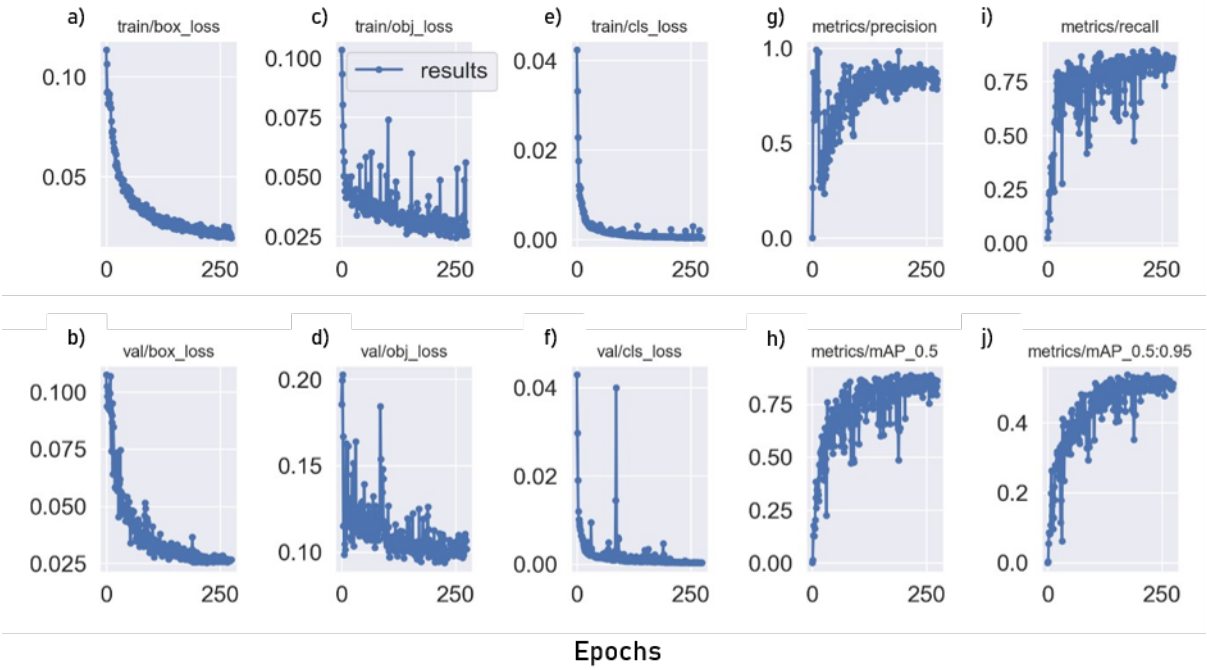


Figure 4: The performance of the YOLOv5 model trained on the custom data: a), b) the box-loss plot indicates the performance of the model in terms of locating the bounding box around the defect; c), d) the obj-loss plot is significantly a measure of the likelihood that a defect is present in the predicted 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 while; i) recall is the ratio of accurately predicted labels to all the correct labels; h), j) the mean average precision (mAP) is a measure to evaluate the performance of the object detection models in terms of actual and predicted bounding box differences.

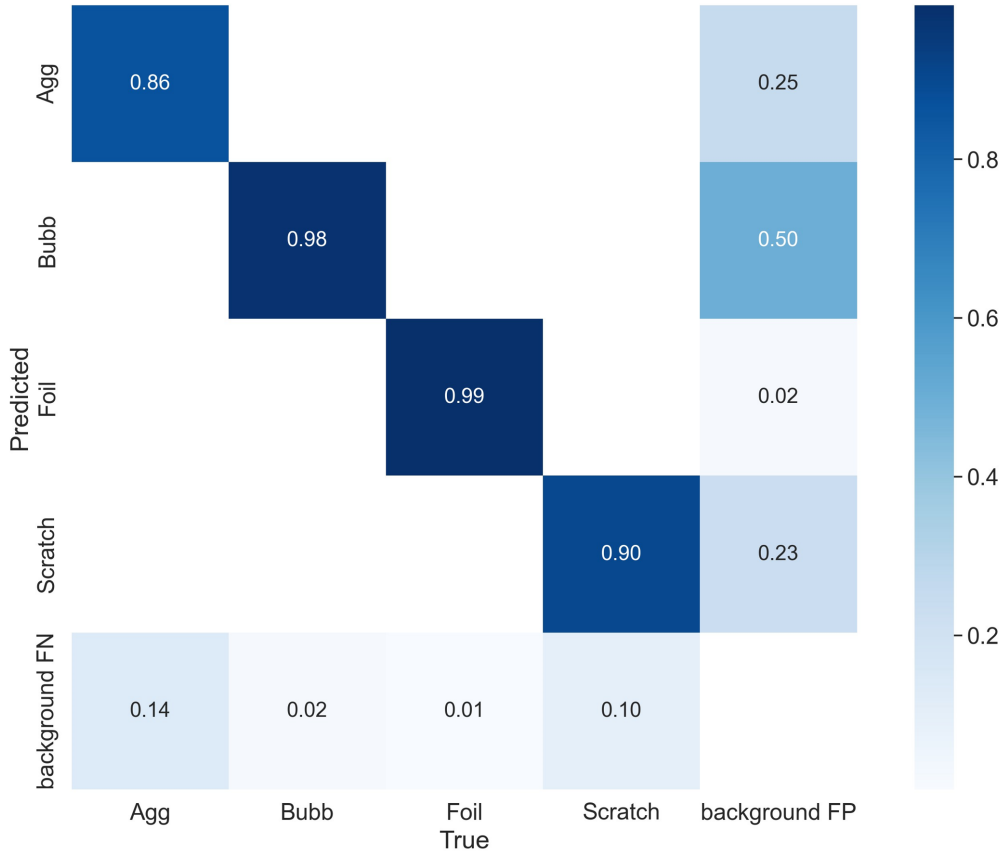


Figure 5: The confusion matrix showing the actual and predicted class labels for defects. Foil class achieved the height precision and recall. Bubble class obtained higher recall as compared to precision. The model performed almost consistently while predicting the agglomerates and scratches. The model achieved 75% total accuracy.

A confusion matrix[37] is a NXN 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.

$$IoU = \frac{area\,(BBg \cap BBp)}{area\,(BBg \cup BBp)}$$

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 true positives (TP), false positives (FP), and false negatives (FN) 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). True positive is an outcome of correct detection made by the model where the actual annotation and predicted bounding box locate the same defect (detection

with IoU ≥ threshold). False positive is a result of wrong detection where the ground truth annotation and predicted bounding box does not match. False negative is a false alarm i.e. there is a defect but the model can not 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, 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 labelled 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 labelled instances of these classes as compared to the foil class, the FP can be further reduced by obtaining and labelling 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 9.5ms 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.

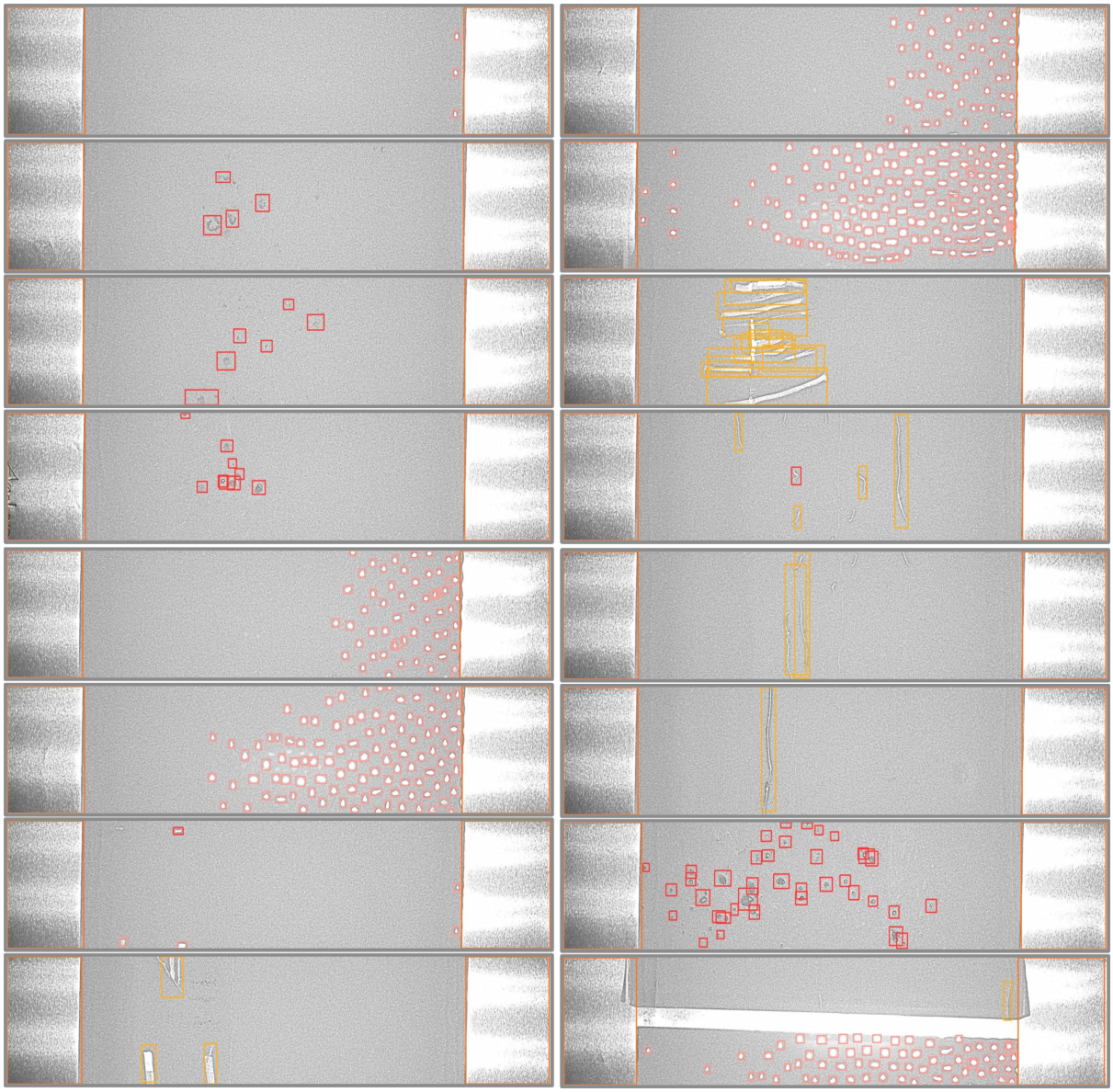


Figure 6: A batch of defected electrode images from the test dataset. The model predicts the bounding boxes around different types of defects that indicate the spatial location of the defects on the electrode surface along with the confidence score. The score demonstrates the possibility that the predicted 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).

Spatial distribution

This paper 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 the relation between input parameters and defects. The whole data (882 images) is 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 aluminium 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 Figures 7, it can be concluded that there are more defects in the right part of the electrode surface.

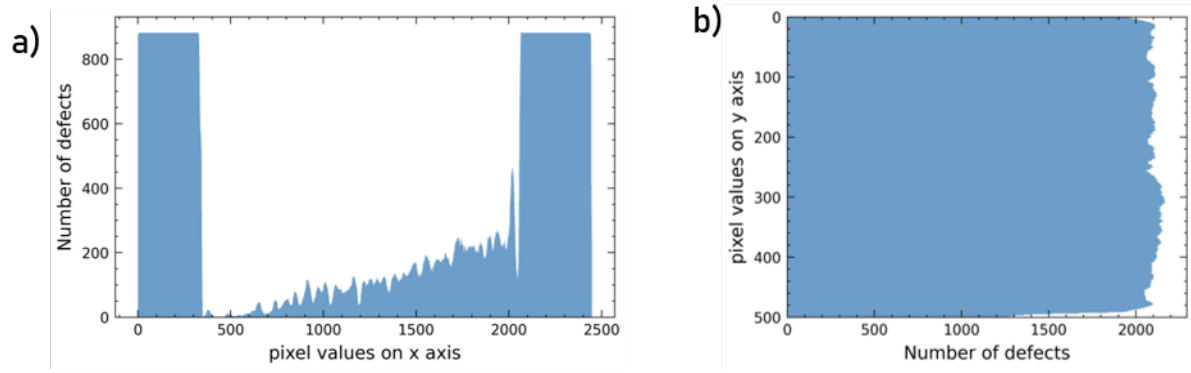


Figure 7: The distribution of the defects present on the electrode surface. a) The count of defects along the horizontal axis where x axis represents the pixel values and y axis represents the number of defects. In the horizontal direction, the left and right parts of the image consist of foil. Since the data set 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. Since 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.

Since 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 centre 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 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.

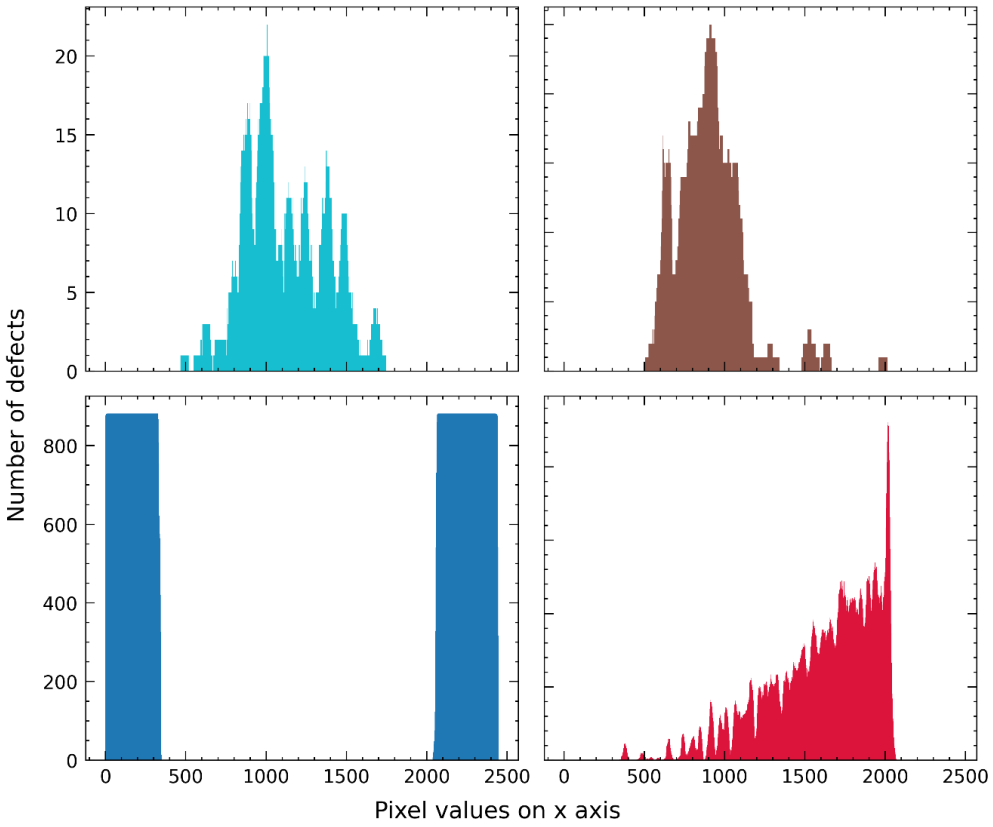


Figure 8: The per category defects on the horizontal axis. a)Scratch; b) Agglomerate; c) Foil; d) Bubble. Horizontal axis represents the pixel value along the axis and the vertical axis represents the number of defects per category. The agglomerates and scratches are distributed in a nearly Gaussian shape and occur in the central part of the electrode. These types of defects rarely occur in the entire data set. 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.

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

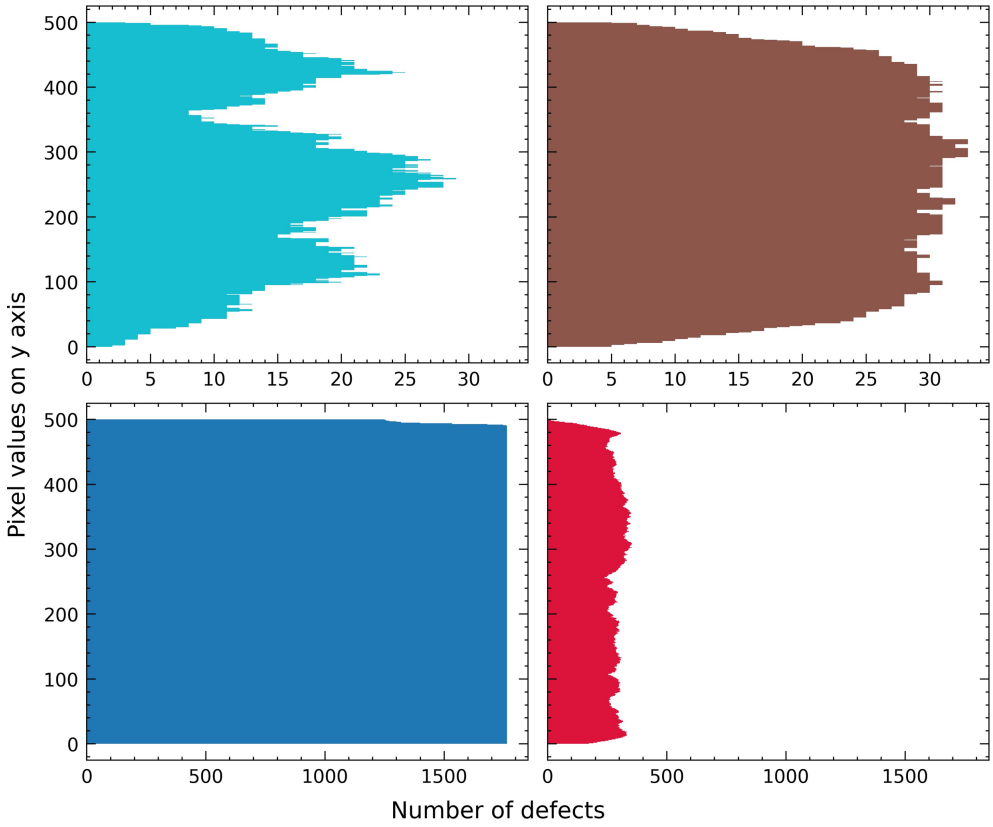
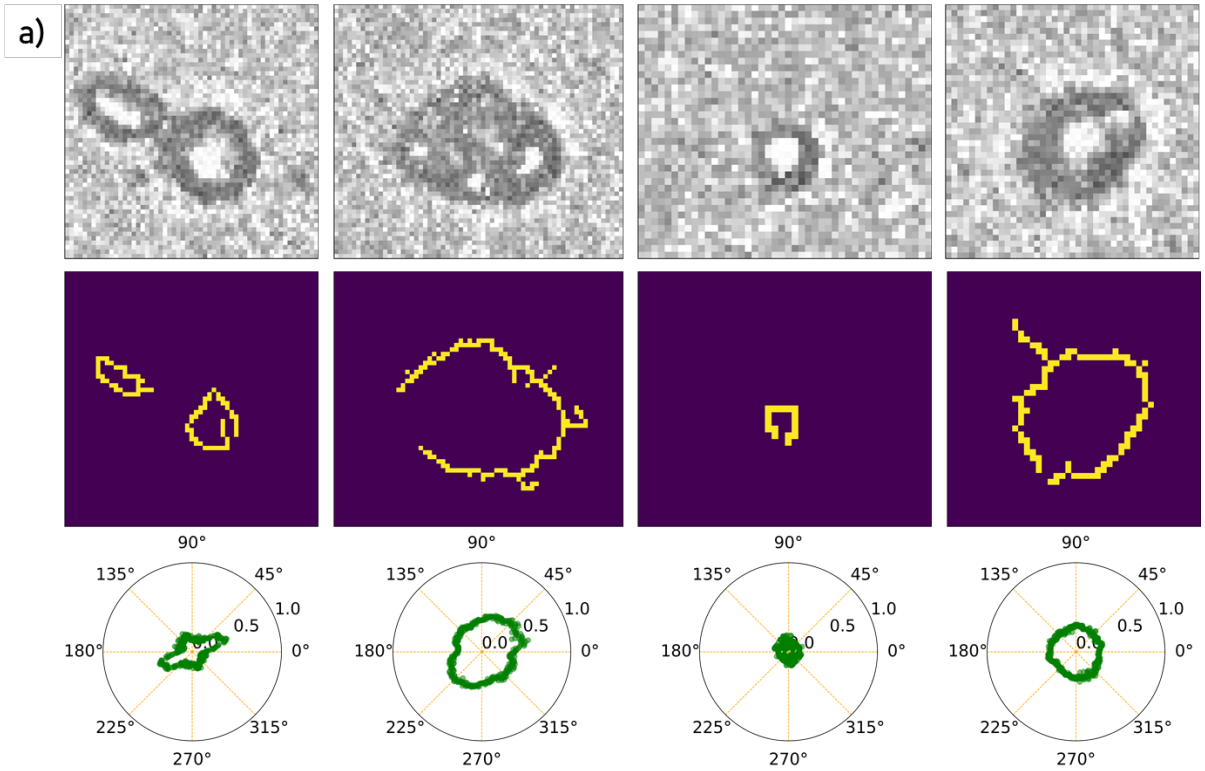
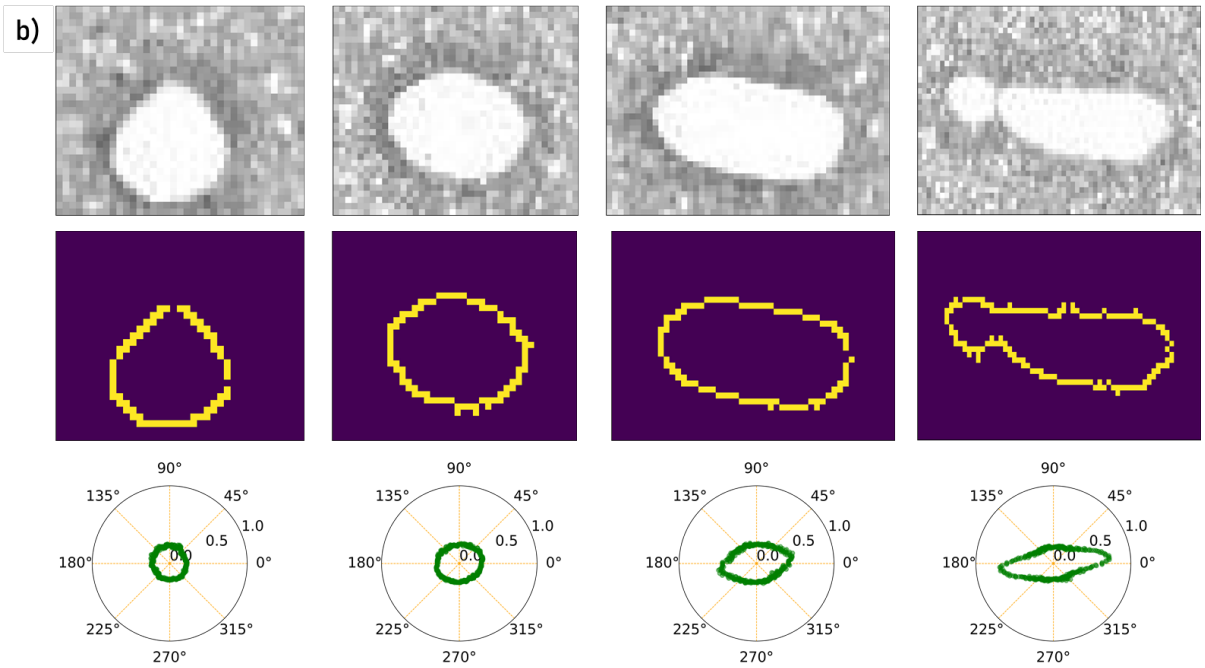


Figure 9: The per category defects on the vertical axis. a)Scratch; b) Agglomerate; c) Foil; d) Bubble. Horizontal axis represents the number of defects per category and the vertical axis represents the pixel value along the axis. There are fewer scratches and agglomerates compared to bubbles. Since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

The information about the size and shape of the defect can be helpful for physics-based simulations. 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 5x5 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 non-maximum 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 non-edges and will be rejected. All edges whose intensity gradient is in between these two thresholds are categorized either as edges or non-edges 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. Since 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 degrees 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.


a)
90°
135°
45°
1.0
0.5
180°
0°
225°
315°
270°


b)
90°
135°
45°
1.0
0.5
180°
0°
225°
315°
270°

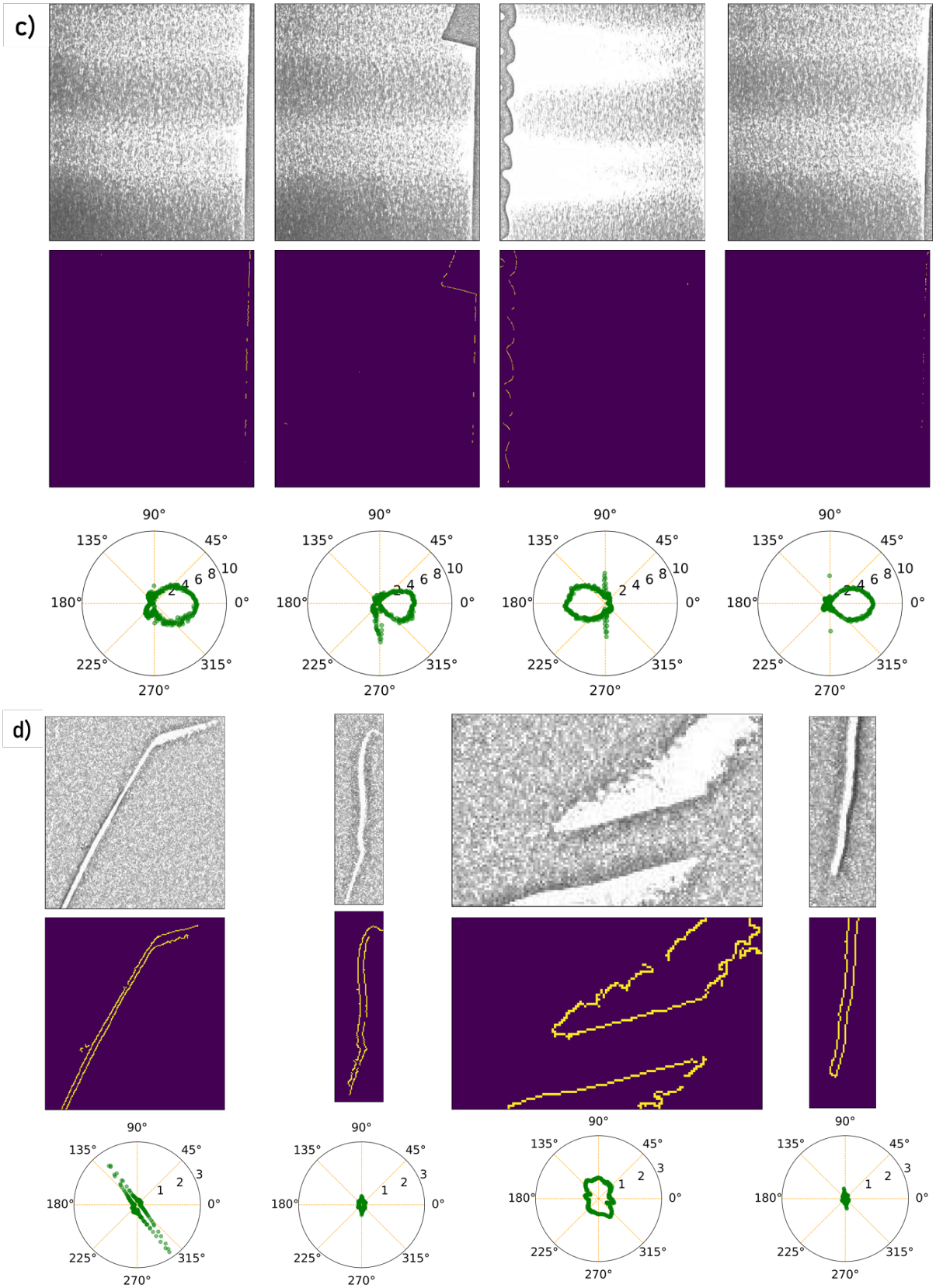


Figure 10: The defects of each class together with their shape and size: (a) Agglomerate; (b) Bubble; (c) Foil; (d) Scratch. The images in the first row are the actual defects on the electrode, cropped by the model from the original electrode images based on the bounding box coordinates. The second row 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 degrees.

Conclusion

In this paper, we propose YOLOv5 , a Deep Learning-based framework, for early detection of defects in electrode production lines. This study can be used as a preventive measure to improve electrode and manufacturing quality. This model has been trained on a few examples and yet shows promising performance that can make any human in the loop inefficient. The model can precisely predict each defect class in static (image) and dynamic (video) datasets and achieved 88% mAP for all the classes combined. Since there was less training data, we see great potential to improve the performance of the model by training with an additional number of training data instances.

The model is capable of detecting defects in near at a web speed of 9 m/s, 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.

The original YOLOv5 model was trained on the COCO dataset, a popular computer vision dataset that contains labeled images of 80 different objects but does not contain defect related data. Therefore, the model is trained on a custom dataset where the images are annotated using the LabelMe tool. This is a manual and time-consuming task, as the performance of the model is highly affected by the number of annotated instances of the defects and the exact defect localization. Active learning is a potential candidate to annotate a larger number of images to improve the performance of the model.

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 programme 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.

Author Contributions

tbd

References

(1) Michaelis, D.; Rahimzei, E.; Kampker, P.; Heimes, H.; Offermanns, C.; Locke, M.; Löbberding, H.; Wennemar, S.; Thielmann, A.; Hettesheimer, D.; Neef, C.; Kwade, A.; Haselrieder, W.; Blömeke, S.; Doose, S.; von Drachenfels, N.; Drees, R.; Fröhlich, A.; Gottschalk, L.; Huang, Z. *Roadmap Batterie-Produktionsmittel 2030 - Update 2020*; 2021.

(2) Gutoff, E. B.; Cohen, E. D. *Coating and Drying Defects: Troubleshooting Operating Problems*; John Wiley & Sons, 2006.

(3) Gitis, A.; Kwade, A.; Sauer, D. U. Flaw Detection in the Coating Process of Lithium-Ion Battery Electrodes with Acoustic Guided Waves, Institut für Stromrichtertechnik und Elektrische Antriebe ISEA, 2017.

(4) Mohanty, D.; Hockaday, E.; Li, J.; Hensley, D. K.; Daniel, C.; Wood, D. L. Effect of Electrode Manufacturing Defects on Electrochemical Performance of Lithium-Ion Batteries: Cognizance of the Battery Failure Sources. *J. Power Sources* **2016**, *312*, 70–79. https://doi.org/10.1016/j.jpowsour.2016.02.007.

(5) Duffner, F.; Mauler, L.; Wentker, M.; Leker, J.; Winter, M. Large-Scale Automotive Battery Cell Manufacturing: Analyzing Strategic and Operational Effects on Manufacturing Costs. *Int. J. Prod. Econ.* **2021**, *232*, 107982. https://doi.org/10.1016/j.ijpe.2020.107982.

(6) Sakti, A.; Michalek, J. J.; Fuchs, E. R. H.; Whitacre, J. F. A Techno-Economic Analysis and Optimization of Li-Ion Batteries for Light-Duty Passenger Vehicle Electrification. *J. Power Sources* **2015**, *273*, 966–980. https://doi.org/10.1016/j.jpowsour.2014.09.078.

(7) Wen, H.; Huang, C.; Guo, S. The Application of Convolutional Neural Networks (CNNs) to Recognize Defects in 3D-Printed Parts. *Materials* **2021**, *14* (10), 2575. https://doi.org/10.3390/ma14102575.

(8) Reynolds, C. D.; Slater, P. R.; Hare, S. D.; Simmons, M. J. H.; Kendrick, E. A Review of Metrology in Lithium-Ion Electrode Coating Processes. *Mater. Des.* **2021**, *209*, 109971. https://doi.org/10.1016/j.matdes.2021.109971.

(9) Han, L.; Liu, Y. M.; Lu, W. Z.; Xiao, F. Design of Automatic Production Line for Electrode Defects Inspection of Li-Ion Power Battery. *Appl. Mech. Mater.* **2014**, *470*, 400–403. https://doi.org/10.4028/www.scientific.net/AMM.470.400.

(10) Masias, A.; Marcicki, J.; Paxton, W. A. Opportunities and Challenges of Lithium Ion Batteries in Automotive Applications. *ACS Energy Lett.* **2021**, 6 (2), 621–630. https://doi.org/10.1021/acsenergylett.0c02584.

(11) Badmos, O.; Kopp, A.; Bernthaler, T.; Schneider, G. Image-Based Defect Detection in Lithium-Ion Battery Electrode Using Convolutional Neural Networks. *J. Intell. Manuf.* **2020**, *31* (4), 885–897. https://doi.org/10.1007/s10845-019-01484-x.

(12) Mayer, D.; Wurba, A.-K.; Bold, B.; Bernecker, J.; Smith, A.; Fleischer, J. Investigation of the Mechanical Behavior of Electrodes after Calendering and Its Influence on Singulation and Cell Performance. *Processes* **2021**, 9 (11), 2009. https://doi.org/10.3390/pr9112009.

(13) Wood, D. A.; Kafiabadi, S.; Al Busaidi, A.; Guilhem, E. L.; Lynch, J.; Townend, M. K.; Montvila, A.; Kiik, M.; Siddiqui, J.; Gadapa, N.; Benger, M. D.; Mazumder, A.; Barker, G.; Ourselin, S.; Cole, J. H.; Booth, T. C. Deep Learning to Automate the Labelling of Head MRI Datasets for Computer Vision Applications. *Eur. Radiol.* **2022**, *32* (1), 725–736. https://doi.org/10.1007/s00330-021-08132-0.

(14) Vishnu, R.; Krishna Prakash, N. Mobile Application-Based Virtual Assistant Using Deep Learning.

In *Soft Computing and Signal Processing*; Reddy, V. S., Prasad, V. K., Wang, J., Reddy, K. T. V., Eds.; Springer: Singapore, 2022; pp 609–617. https://doi.org/10.1007/978-981-16-1249-7_57.

(15) Pan, X.; Lin, X.; Cao, D.; Zeng, X.; Yu, P. S.; He, L.; Nussinov, R.; Cheng, F. Deep Learning for Drug Repurposing: Methods, Databases, and Applications. *WIREs Comput. Mol. Sci. n/a* (n/a), e1597. https://doi.org/10.1002/wcms.1597.

(16) Dutta, S. An Overview on the Evolution and Adoption of Deep Learning Applications Used in the Industry. *WIREs Data Min. Knowl. Discov.* **2018**, 8 (4), e1257. https://doi.org/10.1002/widm.1257.

(17) LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521* (7553), 436–444. https://doi.org/10.1038/nature14539.

(18) Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–21. https://doi.org/10.1109/TNNLS.2021.3084827.

(19) Li, Y.; Wang, H.; Dang, L. M.; Nguyen, T. N.; Han, D.; Lee, A.; Jang, I.; Moon, H. A Deep Learning-Based Hybrid Framework for Object Detection and Recognition in Autonomous Driving. *IEEE Access* **2020**, 8, 194228–194239. https://doi.org/10.1109/ACCESS.2020.3033289.

(20) Zaidi, S. S. A.; Ansari, M. S.; Aslam, A.; Kanwal, N.; Asghar, M.; Lee, B. A Survey of Modern Deep Learning Based Object Detection Models. *Digit. Signal Process.* **2022**, *126*, 103514. https://doi.org/10.1016/j.dsp.2022.103514.

(21) Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*; O'Reilly Media, Inc., 2019.

(22) Vasudevan, S. K.; Pulari, S. R.; Vasudevan, S. *Deep Learning: A Comprehensive Guide*; Chapman and Hall/CRC: New York, 2021. https://doi.org/10.1201/9781003185635.

(23) Baloni, D.; Verma, S. Detection of Hydrocephalus Using Deep Convolutional Neural Network in Medical Science. *Multimed. Tools Appl.* **2022**, 1–23. https://doi.org/10.1007/s11042-022-11953-w.

(24) Kandpal, L. M.; Park, E.; Tewari, J.; Cho, B.-K. Spectroscopic Techniques for Nondestructive Quality Inspection of Pharmaceutical Products: A Review. *J. Biosyst. Eng.* **2015**, *40* (4), 394–408. https://doi.org/10.5307/JBE.2015.40.4.394.

(25) Amar, M.; Gondal, I.; Wilson, C. Vibration Spectrum Imaging: A Novel Bearing Fault Classification Approach. *IEEE Trans. Ind. Electron.* **2015**, *62* (1), 494–502. https://doi.org/10.1109/TIE.2014.2327555.

(26) Li, P.; Dolado, I.; Alfaro-Mozaz, F. J.; Casanova, F.; Hueso, L. E.; Liu, S.; Edgar, J. H.; Nikitin, A. Y.; Vélez, S.; Hillenbrand, R. Infrared Hyperbolic Metasurface Based on Nanostructured van Der Waals Materials. *Science* **2018**, *359* (6378), 892–896. https://doi.org/10.1126/science.aaq1704.

(27) Oliveira, H.; Correia, P. L. Automatic Road Crack Segmentation Using Entropy and Image Dynamic Thresholding. In *2009 17th European Signal Processing Conference*; 2009; pp 622–626.

(28) Ngan, H. Y. T.; Pang, G. K. H.; Yung, N. H. C. Automated Fabric Defect Detection—A Review. *Image Vis. Comput.* **2011**, *29* (7), 442–458. https://doi.org/10.1016/j.imavis.2011.02.002.

(29) Mahajan, P.; Kolhe, S.; Patil, P.; Mahajan, J. A Review of Automatic Fabric Defect Detection Techniques. *undefined* **2009**.

(30) Hawley, W. B.; Li, J. Electrode Manufacturing for Lithium-Ion Batteries—Analysis of Current and next Generation Processing. *J. Energy Storage* **2019**, *25*, 100862. https://doi.org/10.1016/j.est.2019.100862.

(31) Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; NanoCode012; Kwon, Y.; TaoXie; Fang, J.;

imyhxy; Michael, K.; Lorna; V, A.; Montes, D.; Nadar, J.; Laughing; tkianai; yxNONG; Skalski, P.; Wang, Z.; Hogan, A.; Fati, C.; Mammana, L.; AlexWang1900; Patel, D.; Yiwei, D.; You, F.; Hajek, J.; Diaconu, L.; Minh, M. T. *Ultralytics/Yolov5: V6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference*; Zenodo, 2022. https://doi.org/10.5281/zenodo.6222936.

(32) Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *ArXiv150602640 Cs* **2016**.

(33) Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C. L.; Dollár, P. Microsoft COCO: Common Objects in Context. *ArXiv14050312 Cs* **2015**.

(34) Chen, Y.; Zeng, X.; Chen, X.; Guo, W. A Survey on Automatic Image Annotation. *Appl. Intell.* **2020**, *50* (10), 3412–3428. https://doi.org/10.1007/s10489-020-01696-2.

(35) Russell, B. C.; Torralba, A.; Murphy, K. P.; Freeman, W. T. LabelMe: A Database and Web-Based Tool for Image Annotation. *Int. J. Comput. Vis.* **2008**, *77* (1–3), 157–173. https://doi.org/10.1007/s11263-007-0090-8.

(36) *Ultralytics/Yolov5*; Ultralytics, 2022.

(37) Padilla, R.; Netto, S. L.; da Silva, E. A. B. A Survey on Performance Metrics for Object-Detection Algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*; 2020; pp 237–242. https://doi.org/10.1109/IWSSIP48289.2020.9145130.