


Defect detection in battery electrode production using supervised and unsupervised learning with laser speckle photometry data

Jaromír Klarák^{1*} , Lili Chen², Ulana Cikalova², Peter Malík¹,
Robert Andok¹, Beatrice Bendjus²

¹ Institute of Informatics, Slovak Academy of Sciences, 845 07 Bratislava, Slovakia

² Fraunhofer Institute for Ceramic Technologies and Systems IKTS, Maria-Reiche-Strasse 2, 01109 Dresden, Germany

* Corresponding author's e-mail: jaromir.klarak@savba.sk

ABSTRACT

The push for zero-emission transport is largely driven by the adoption of battery-powered vehicles. A critical aspect of a successful battery storage system is the production of high-quality electrodes, which necessitates rigorous inspection processes and defect detection systems. In this paper, we present data obtained using laser speckle photometry (LSP) technology and perform defect detection using two approaches: the YOLOv4 model and the newly developed U2S-CNNv2 model. The U2S-CNNv2 model combines unsupervised and supervised learning to identify defects beyond the training dataset. Our goal is to develop an efficient detection of defects for battery electrode production to meet stringent quality control standards. Our findings show that YOLOv4 is highly effective for deployment in inspection processes, capturing very small defects and operating at 50 frames per second (fps). YOLOv4 achieved an impressive 93.82% accuracy in correctly detecting and 91.10% in correctly labeling defects. Conversely, the U2S-CNNv2 model excels in precisely localizing defect areas and identifying unknown defects or patterns not included in the training dataset. However, it operates at a slower pace of around 3 fps and has a detection accuracy of 83.83% and correct labeling rate of 54.84%.

Keywords: defect detection, anomaly detection, battery, electrode, supervised learning, unsupervised learning, deep learning.

INTRODUCTION

The constant growth of the market for electrotechnical devices in various areas such as electromobility, medical technology, and renewable energies requires powerful and reliable electrical energy storage systems. The material from which these energy storage devices are made plays a major role in their performance [1]. Ceramics are used in a wide variety of applications due to their chemical, thermal, and mechanical resistance, as well as their high electrical resistance. These properties offer great potential for lithium-ion batteries (LIB) as energy storage devices. In LIB, ceramics are used as components of the electrodes, which significantly influence the battery's performance.

Battery production is complex and consists of many steps. If a defective electrode is not identified and removed during the production process, it can result in high costs, as the defects may only become apparent later in the production cycle. If a defective battery passes quality tests, it can cause substantial material damage in its environment [2, 3]. Therefore, it is essential to enhance production processes to be more efficient and sustainable. The coating process is particularly critical, with a high scrap rate of 2–5% and accounting for 20–40% of the total manufacturing cost. Surface defects occurring during the coating process are especially critical for the performance and safety of the battery cell and must be detected. To address this, it is necessary to implement an inspection system,

particularly for defect detection, in the electrode production line to ensure the quality of electrodes used in batteries. Research in the field of defect detection is continuously evolving. Early research designed CNN models to detect objects in visual data. This detection method was based on a two-step process: the first step suggested region proposals, which were then classified into labels in the second step. This method, known as Regions with CNN features (R-CNN), was introduced by Girshick et al. [4]. Improved versions of this approach include Fast R-CNN [5] and Faster R-CNN [6], which offer better optimization and real-time object detection capabilities.

The evolution in object detection continued with the development of the one-step detector, you look only once (YOLO) [7]. YOLO is a deep learning algorithm developed for real-time object detection. It employs a single convolutional neural network (CNN) model for end-to-end object detection by regressing bounding boxes as Regions of Interest (RoI). Unlike traditional CNN-based object detection algorithms such as the R-CNN family, YOLO treats object detection as a regression problem. The input image undergoes a single inference to determine the locations and the classes of all objects within the image, along with their respective confidence probabilities. YOLO offers several advantages, including fast detection speed, low background false detection rate, and strong generalization ability, making it an appropriate model for industrial applications. The success of this model has led to the creation of an entire family of similar and improved models. For example, an improved YOLO model applied to detect defects on steel strip surfaces achieved a 99% detection rate at a speed of 83 fps [8]. Similar solutions based on YOLOv3 achieved a precision of 79% [9]. The application of an LF-YOLOv4-based model demonstrated elaborate surface defect detection in LIB electrode [10]. The comparison of YOLO-based models has become a focus in defect detection research. One study describes a YOLOv5 model deployed to capture defects on coated electrode surfaces, identifying the exact areas with defects using Canny Edge Detection [11]. Another application of the newer YOLOv8 model trained to detect eight types of defects on battery separators achieved a precision of 92.20% [12]. While YOLO-based models are single-stage detectors, two-stage detectors are primarily based on R-CNN variations. An

improved Faster R-CNN model trained to detect defects in LIB electrodes coatings achieved a precision of 97.2%, compared to YOLOv3's 84.2% and SSD300's 77.00% [13].

In addition to these models, anomaly detection is another approach to capturing defects. Applications on the MVTech database (carpet), acne patches, or PCB fiberglass substrates illustrate this approach [14]. One study proposed the inspection of aircraft components using an autoencoder, demonstrating good reconstruction ability with defect-free samples, though not as effective with defected samples [15]. The adversarial autoencoder applied within industrial applications identifies defective and defect-free samples, overlaying detection results over the input to pinpoint defects [16]. Another application describes defect detection on metallic surface parts using an autoencoder, classifying anomalies based on type and severity [17]. The main deficiency of anomaly detection is the lack of labeled information about the anomalies.

To address these limitations, a new approach called U2S-CNN combines the benefits of supervised and unsupervised methods for defect detection [18]. This method uses an autoencoder trained on unlabeled data, followed by clustering and classification performed by a classifier trained on labeled data [19]. A very similar solution, SIGMA (Spectral Interpretation Using Gaussian Mixtures and Autoencoder), was developed by Dr. Tung et al. from the University of Cambridge. This analysis tool comprises three steps: an autoencoder (Unsupervised learning), a clustering process using Gaussian mixture modeling (GMM) (Unsupervised learning), and non-negative matrix factorization. The tool aims to provide material analysis by isolating background-subtracted EDS spectra of individual phases [20, 21].

The current state of the art in defect detection primarily relies on YOLO models. While effective in detection, defect detection differs from standard object detection in its need to capture unknown shapes and patterns not included in the training dataset. To address this, a second branch of research focuses on anomaly detection, which does not generate annotated outputs for defects but instead identifies anomalies. Our proposed method, based on U2S-CNN, aims to combine the advantages of both approaches to achieve better results and supplement the deficiencies of existing methods, ensuring successful industrial deployment of inspection systems.

SOURCE OF DATA – LASER SPECKLE PHOTOMETRY

The production of LIB is a highly specific and technologically demanding process. A common approach to addressing the challenges in electrode inspection is the implementation of inline optical sensor systems, which enable automatic and continuous surface inspection within the production line. This approach allows for the early identification of scrap, thereby preventing material- and energy-intensive processing through the targeted removal of defective electrodes. Detecting surface defects across the entire coating width at high web speeds, in real time, and with high resolution is a significant challenge. In particular, the identification of low-contrast defects in the micrometer range is often inadequate with conventional optical inspection systems using an LED lighting unit due to the rough coating surface. An innovative solution to this problem is the use of Laser Speckle Photometry (LSP), developed at Fraunhofer IKTS-MD. In LSP technology, a laser source is used to illuminate inspected objects instead of the LED illuminations in the conventional optical inspection systems. Due to the high spatial and temporal coherence of laser source, speckle patterns regarding as the “fingerprint” of are generated providing the texture characterizations of surfaces being inspected. LSP captures and analyzes the laser speckle patterns as the detection signal and is applicable to all classes of LIB electrodes [22, 23].

SAMPLES AND DATA PREPARATION

LIB consist of two electrodes, each with a current collector, a separator that separates the electrodes, and an ion-conducting electrolyte liquid in the spaces between the electrodes. LIB production can be divided into three main stages: electrode production, cell assembly, and formation. The electrode production process is particularly complex. Both the anode and cathode undergo similar steps but use different materials and carrier foils. The production process of electrode includes four key steps: mixing all components, coating the carrier film, drying the coated substrate, and calendaring. Coating is performed in a roll-to-roll (R2R) process, either intermittently or continuously, depending on the system. After coating, the coated carrier film is passed through

a dryer. According to a roadmap published by the German Engineering Federation (VDMA), the coating width of carrier films is expected to extend up to 2000 mm by 2030, with coating speeds reaching up to 100 m/min if possible [3].

Each step in the production process has the potential to introduce defects that can significantly impact the quality of the electrode surface, and consequently, the performance of the final product. During the slurry preparation, material clumps (agglomerates), air inclusions, or impurities from previous slurries can occur. These issues can lead to clogging of the coating tool during the coating process, resulting in coating-free stripes or holes due to burst air bubbles on the carrier film [24]. Cracks are a major problem during the drying phase, often arising from improperly set machine parameters such as film speed and drying oven temperature, or from an incorrect slurry recipe.

The samples used in this study are uncalendared lithium titanate oxide cathodes and graphite anodes applied to 20 μm thick aluminum foil and copper foil respectively, containing agglomerates and other defects such as cracks, pinholes, line defects, and foreign particles. A 10 cm segment was cut lengthwise from each sample, to be used later as test material for model evaluation, while the remaining part was used for training. The test material was strictly not used in training to ensure the validity of the evaluation results.

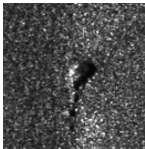
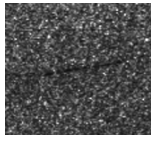
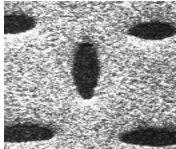

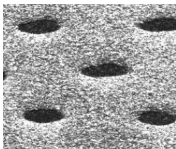
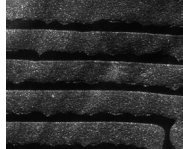
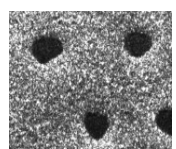

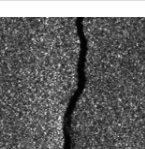
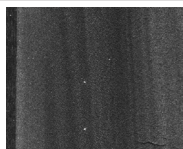
Initially, electrode samples with predetermined defects were measured using LSP. The acquired speckle images were then meticulously labeled through manual annotation using Label Studio software. During the labeling phase, both the position and size of defects (bounding box) were systematically recorded image by image. Following the labeling process, images containing the associated defect information were integrated into a model for subsequent training. The defects data used obtained by LSP are listed in Table 1. The annotated data includes 17,545 occurrences of defects across 4,684 image samples.

DETECTION IN ELECTRODE SURFACE FOR BATTERY INDUSTRY

U2S-CNNv2 method

For defect detection in the battery industry, we applied the U2S-CNN method [19], which has been improved for this specific application. This

Table 1. Summary of defects in training dataset and corresponding descriptions

Defect type	Description	Example image	Class	Defect type	Description	Example image	Class
			Num. of defects				Num. of defects
Agglomerate	A ball-like gathering of particles		0	Crack-trans	Cracks along the transversal direction of electrode foil		5
			2,087				1,976
Pinhole-long	Oval pinhole with its long semi-axis along the running direction of electrode foil		1	Line-long	Line defects along the running direction of electrode foil		6
			1,919				1,791
Pinhole-trans	Oval pinhole with its long semi-axis along the transversal direction of electrode foil		2	Line-trans	Line defects along the transversal direction of electrode foil		7
			1,756				1,742
Pinhole-round	Round pinhole		3	Line-Diag	Line defects along the diagonal direction of electrode foil		8
			2,590				862
Crack-long	Cracks along the running direction of electrode foil		4	Foreign particle	Contamination of foreign particles		9
			974				1,848

method comprises three parts: an autoencoder, clustering using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [25], and a classification neural network (classifier).

Autoencoder

The autoencoder was designed with an input layer of shape $640 \times 640 \times 3$. The first four layers (shown in blue in Figure 1) are convolution 2D layers with 8, 32, 64, and 128 filters, respectively. The decoding part consists of four conv2DTranspose layers with 128, 64, 32, and 3 filters. The output layer is in the shape of $640 \times 640 \times 3$. The primary purpose of the autoencoder is to train on correct samples of electrodes. During the reconstruction process, defects are replaced by the correct patterns of electrodes. The main improvements include higher resolution, a shallower architecture, and the use of transpose convolution neural networks. The training process involved 821 samples, separated into 80% training and 20% validation samples. The accuracy reached

only 32%. The output from the autoencoder is used in the Loss function (Figure 1), which identifies the most different pixels between generated and inspected pixels. These differing pixels are then input for the clustering process.

Clustering

In the comparison process, anomalies (pixels with significantly different values) are separated into logical groups, or clusters. These clusters are the result of the clustering process performed by DBSCAN. This process groups pixels into clusters based on two conditions: the Eps-neighborhood of a point (denoted as Eps) and the condition of having at least a minimum number of points (denoted as MinPts). Based on these two parameters, pixels (or points) are categorized into three types: core points, border points, and noise. A point is classified as a core point if its Eps-neighborhood contains more than MinPts points. If it does not meet this condition but is still within the Eps-neighborhood of a core point, it is categorized as a border point. Points

that do not satisfy either condition are categorized as noise [25]. From these clusters, Regions of Interest (RoIs) are obtained. These RoIs are resized to $224 \times 224 \times 3$ for the classification process.

Classifier

The previous classifier, based on the Xception architecture [26], was replaced by a custom architecture. The previous version used an input layer of shape $71 \times 71 \times 3$. The new design architecture, with an input layer of shape $224 \times 224 \times 3$, is better suited for these purposes and achieves better results with higher resolution. The new classifier contains six conv2D layers and four fully connected layers. The output is in the form of 11 classes, corresponding to 10 types of defects and one class for good samples detected in this experiment. The last class was added to identify incorrect reconstructions detected as anomalies in the second step. To avoid labeling correct samples as defects, it is necessary to train the classifier on good samples as well. A detailed and schematic illustration of the entire U2S-CNNv2 is in Figure 1.

The entire process of applying U2S-CNNv2 is illustrated in Figure 2. Figure 2a shows an image with captured defects in the coating of an electrode, represented as black entities on the surface. Due to the characteristics of electrode production, the defects are mainly in this form or as scratches. The primary purpose of the autoencoder is to assimilate these defects into the standard pattern of a well-coated surface of electrodes. The reconstruction results are illustrated in Figure 2b. Differences between Figures 2a and 2b are shown

in Figure 2c. For the classification process, it is suitable to focus on the most different pixels to perform a faster clustering process. The pixels representing differences are defined in Figure 2d.

These pixels are clustered by DBSCAN into clusters. This method is suitable for this task because points or pixels are separated according to their density occurrence in the image. Unlike other clustering methods, DBSCAN does not require a pre-defined number of clusters [27]. This method is resistant to very different shapes of pixel or data densities, which is crucial given the defect shapes in Figure 2a. The result from clustering is illustrated in Figure 2e, where the clusters are colored, with each cluster illustrated with a random color.

In the next step, RoIs are created from these clusters, which are crucial as input to the classifier. These clusters have to be in the form of rectangles, produced by determining the minimal and maximal values of cluster pixel coordinates in the X and Y axes for each cluster. After clustering, these regions are resized to a unified size suitable for the classifier in the form of a standard CNN network. The classifier then labels RoIs according to their high probability of being specific types of defects.

YOLOv4 method

For this experiment, the original YOLOv4 architecture was used and implemented in the PyTorch framework. The training process was defined with the following parameters: learning rate of 0.05, weight decay of 0.005, 300 epochs, and a batch size of 4. The input image size was 640×640 pixels with 3 color channels: red, green,

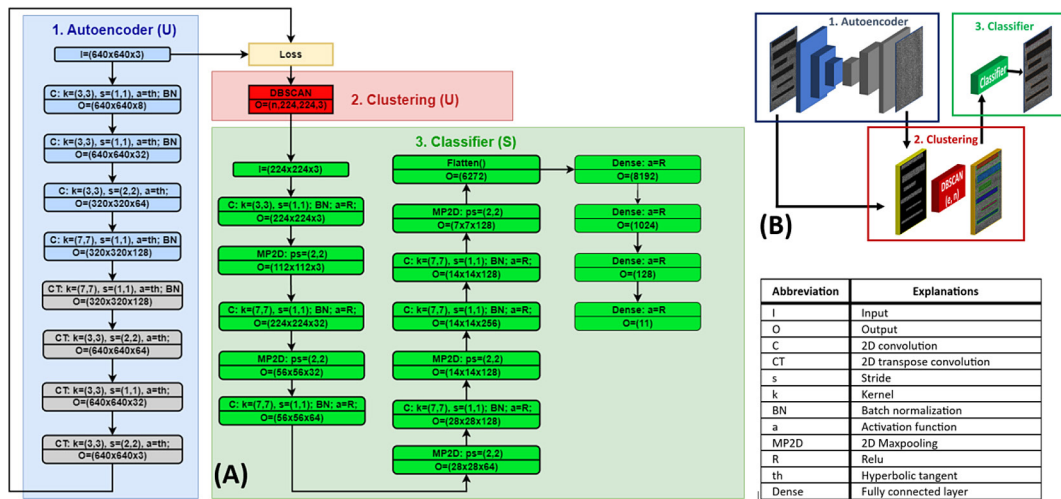


Figure 1. Detailed (A) and schematic (B) architecture of U2S-CNNv2 method applied for battery industry

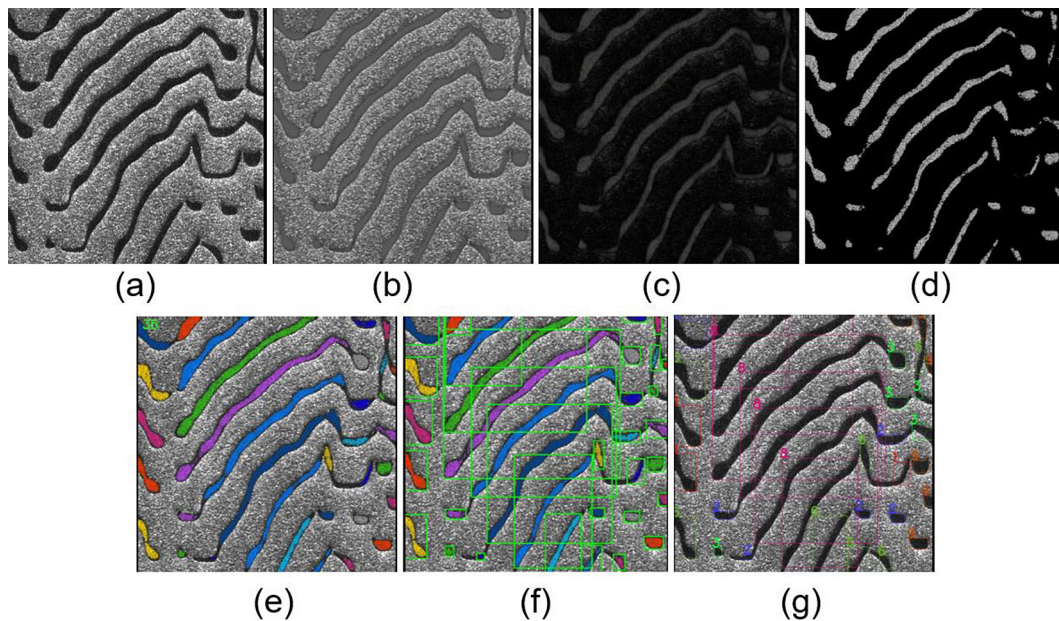


Figure 2. U2S-CNN in battery industry, (a) – tested image, (b) – reconstructed image, (c) – difference between input (a) and output – reconstructed image (b), (d) – threshold differences from (c) (highlighted differences), (e) – clustered differences to clusters by DBSCAN method and make regions of interest (ROIs), (f) – classified ROIs from (f) to labels

and blue (RGB). The types of defects used for the training process are listed in Table 1. Additionally, to enhance the speed of inference and meet the requirements of inline real-time evaluation, the model was converted to the Open Neural Network Exchange (ONNX) format and further implemented on the TensorRT inference platform.

RESULTS

The entire verification of U2S-CNNv2 and YOLOv4 was performed on 505 image samples sized $640 \times 640 \times 3$ in RGB, containing 6357 defects. The complete results from this verification are presented in Table 2. YOLOv4 detected 5964 out of 6357 defects, achieving a detection rate of 93.82%. The correctly detected and labeled defects were 5791, or 91.10%. Regarding the type of defects, the best results were achieved for defect types 1, 2, 3, 4, 5, and 8. The detection rate for defect types 6 and 7 were less than 85%, because these defects typically have a very long and thin shape, which is presumably harder to be detected. Poor results were observed in detecting contamination by foreign particles (type 9), with a detection rate of only 26.67%. This type of defect often appears as very bright pixels, with a very small number of pixels, mainly up to 10 pixels corresponding to a defect. In this

case, the speckle patterns of the foreign particles are sometimes difficult to be distinguished from strong reflections caused by local surface structures. The U2S-CNNv2 model consists of three parts (unsupervised, unsupervised, and supervised learning), where only the last step involves learning from labeled data (classifier). There are 11 classes correspond to 10 classes to defects (Table 1) and one class (class 10) for correct sample patterns. The most incorrectly classified defects were labeled “3” and “4”. The correctly labeled samples in the validation dataset were 81.45%. The number of correctly detected defects by U2S-CNNv2 was 5329, or 83.83% (Table 2). The correctly labeled defects were 3486, or 54.84%. The good detection results were achieved for defect types 1, 2, 3, 6, 7, and best result for 8. According to these results, there are incorrect labeling between types 1,2,3 and mainly type 1 categorized as many times as type 6. Very bad results with 0% accuracy were observed for defect types 0 and 9, which are very small defects. Defect types 4 and 5 also had poor results, which are 4.88% (type 4) and 12.03% (type 5) reached by U2S-CNNv2. The best labeling accuracy was achieved for defect type 8. In comparison, the YOLOv4 model performed better or much better in capturing and labeling defects. For defect types 6 and 7, U2S-CNNv2 achieved better detection results than YOLOv4

Table 2. Results of capturing defects by YOLOv4 and U2S-CNNv2 for test dataset (n = 605 samples)

Type	Number of defects	YOLOv4				U2S-CNNv2			
		True detected		True labeled		True detected		True labeled	
		n	[%]	n	[%]	n	[%]	n	[%]
0	118	103	87.29	102	86.44	0	0.00	0	0.00
1	944	944	100.00	920	97.46	872	92.37	414	43.86
2	1102	1102	100.00	1073	97.37	989	89.75	681	61.80
3	1797	1786	99.39	1733	96.44	1559	86.76	994	55.31
4	123	118	95.93	114	92.68	27	21.95	6	4.88
5	133	123	92.48	122	91.73	46	34.59	16	12.03
6	823	574	69.74	535	65.00	637	77.40	494	60.02
7	261	211	80.84	194	74.33	209	80.08	104	39.85
8	996	986	99.00	982	98.60	984	98.80	777	78.01
9	60	17	28.33	16	26.67	6	10.00	0	0
Total	6357	5964	93.82	5791	91.10	5329	83.83	3486	54.84

for defect type 6. In all other cases, YOLOv4 reached better results in capturing and labeling results than U2S-CNNv2. Further analysis of the performance of these models involved comparing the number of detections to the true defect occurrences in the test data, as shown in Table 3. YOLOv4 suggested 11,230 region proposals (N in Table 3). Compared to the number of defects (Nd = 6357), the ratio Nd/N was 56.61%. This means that YOLOv4 suggested many regions as detected defects. The true region proposals according to defects were 10,173, or 90.59%. Correctly labeled proposals were 87.13%. This indicates that in this configuration and setup,

YOLOv4 suggested many region proposals, leading to a higher number of regions proposed for captured defects. Thus, YOLOv4 generated more false positive values (0.0941) compared to U2S-CNNv2 (0.0906) (Table 4). However, YOLOv4 performed much better in terms of false negatives (0.0618) compared to U2S-CNNv2 (0.1617). From view of unsuccessful capturing of defects, the value of Undetected of defects (Ud) were 6.18% (YOLOv4) and 16.16% (U2S-CNNv2). It is opposite value to true detected of defects mentioned in Table 2. Incorrectly labeled of defects (Md) means true detected defects by

Table 3. Result of detections by YOLOv4 and U2S-CNNv2 in compared to true number of defects in test dataset

Test dataset (n = 605 samples)	YOLOv4		U2S-CNNv2	
	n	Ratio to N [%]	n	Ratio to N [%]
Number of defects (Nd)	6357	56.61	6357	90.31
Detected regions (N)	11230	–	7039	–
Correct detection of defects (D)	10173	90.59	6401	90.94
Correct labeled of defects (L)	9785	87.13	4957	70.42
---		Ratio to Nd [%]		Ratio to Nd [%]
Undetected of defects (Ud)	393	6.18	1028	16.16
Incorrectly labeled of defects (Md)	566	8.90	2871	45.16

Table 4. Result of defect detections by YOLOv4 and U2S-CNNv2 in way of true positive, false positive and false negative

Test dataset (n = 605 samples)	YOLOv4		U2S-CNNv2	
	True positive	False positive	True positive	False positive
	0.9059	0.0941	0.9094	0.0906
False negative	0.0618	---	0.1617	---

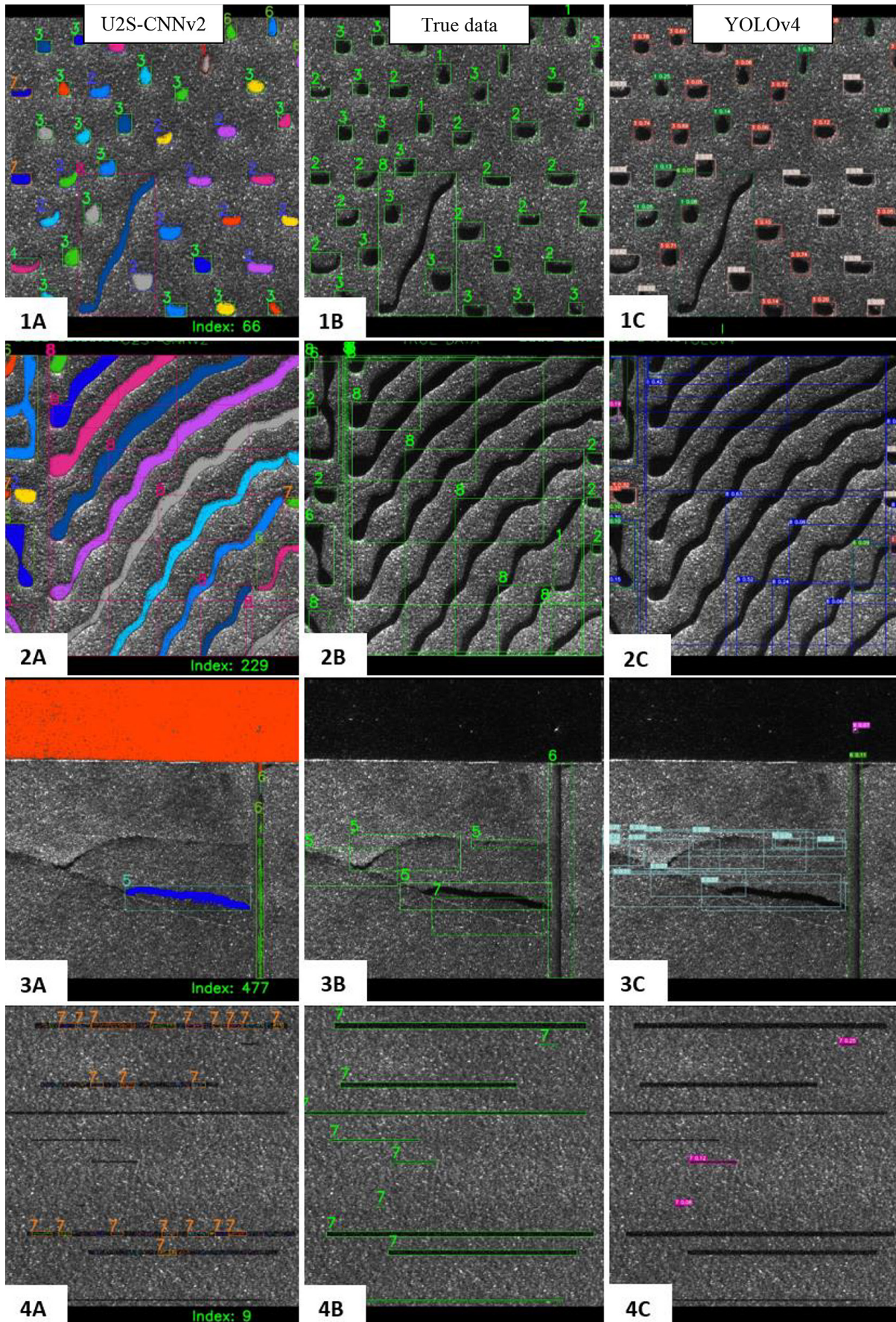


Figure 3. The results (1–4) from U2S-CNN (A - type) and YOLOv4 (C - type) with True data (B - type)

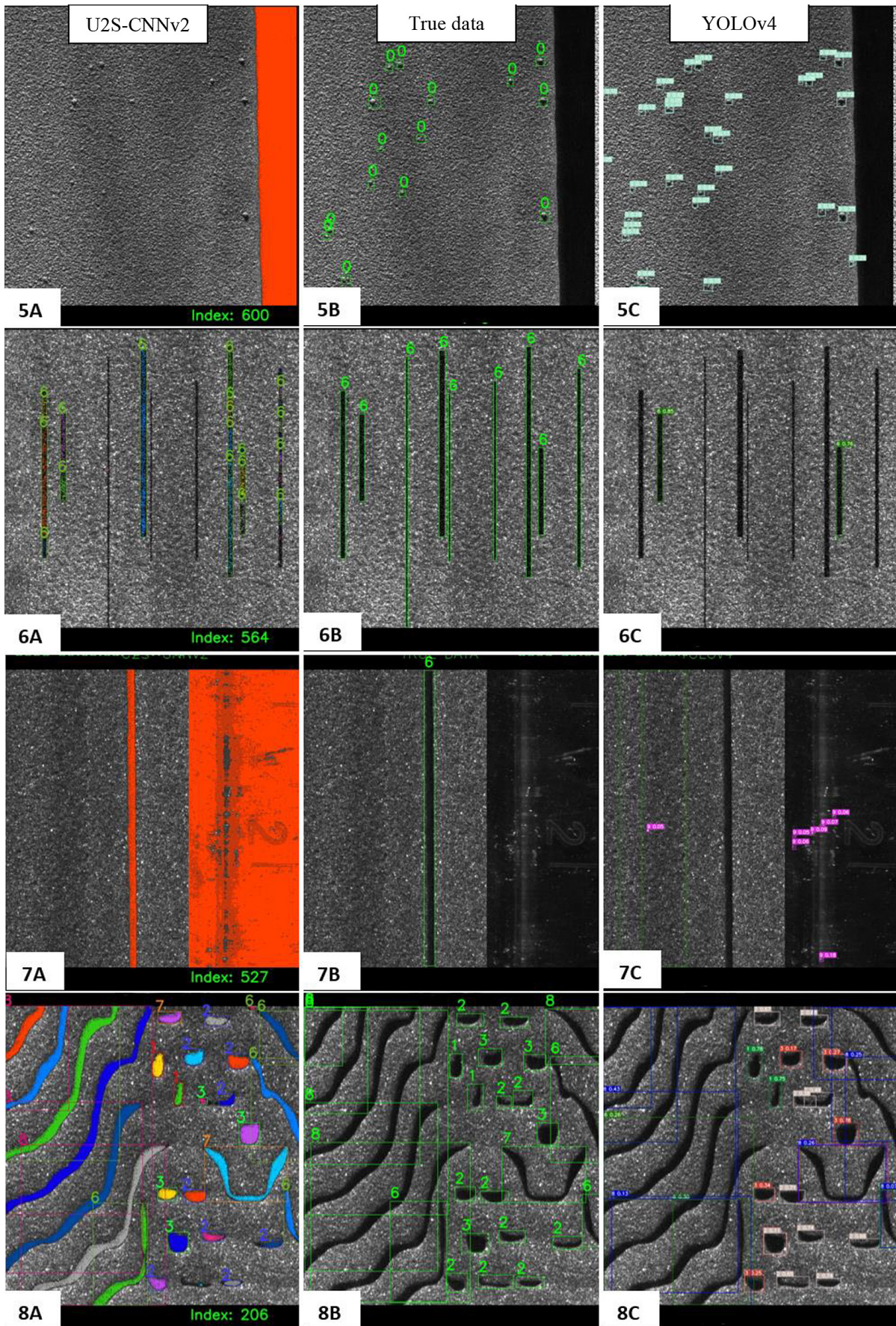


Figure 4. The results (5–8) from U2S-CNN (A - type) and YOLOv4 (C - type) with True data (B - type)

detectors but not correctly labeled were results 8.90% (YOLOv4) and 45.16% (U2S-CNNv2).

From the empirical qualitative evaluation, several key results emerged. Each type of defect has its own specifications. In Figure 3.1B, the true data of defects are shown, mainly of defect types 1, 2, and 3. In this case, both models captured defects very well (Figure 3.1A for U2S-CNNv2 and 1C for YOLOv4). For U2S-CNNv2, there were more incorrectly labeled defects compared to YOLOv4, but both models provided reliable results. In Figure 3.2B, the results were good for both models as well. U2S-CNNv2 showed a significant advantage in highlighting exact areas of defects compared to the RoIs of YOLOv4, with which defects can be easily recognized by people from the images especially for images which have high defect densities. In Figure 3.3C, mainly defect types 5, 6, and 7 occurred. YOLOv4 detected the defects very well, except for type 7. It also captured defect type 9 (Figure 3.3C), which corresponds to the pixel characteristics in this space. U2S-CNNv2 detected only one RoI of type 5, missing others due to poor reconstruction quality and low loss values in this area. However, it detected a large orange rectangle (Figure 3.3A), demonstrating proof of anomaly detection outside the training dataset. Similar observations are found in Figure 4, specifically in 5A and 7A. In Figure 4.4B, defect type 7 occurred, supporting the results noted in Table 2 where U2S-CNNv2 performed better than YOLOv4. The region proposals of U2S-CNNv2 were mainly in smaller pieces than the true RoIs, due to slightly worse reconstruction and insufficiently different pixels in

terms of loss values. For defect type 0, YOLOv4 achieved good results, while U2S-CNNv2 performed very poorly (Figure 4.5A, 5B, 5C). Figure 4.8A, 8B, 8C further supports this analysis.

Merging detection with supervised and unsupervised methods

The YOLOv4 and U2S-CNNv2 models are based on different approaches and methods. By combining these methods, we achieved significant improvements, particularly in the detection and labeling of defect types “6” and “7.” For other defects, the improvements were minor or negligible due to the already high accuracy primarily achieved by YOLOv4.

In the case of defect type “9,” the improvement in detection was only 10.00%, with no improvement in labeling. This is likely because foreign particles typically appear as just a few very bright pixels, making them particularly difficult to accurately detect. Overall, the success rate for correctly detecting defects ranged from 84.57% to 100.00%, and for correctly labeling defects, the rate ranged from 81.17% to 99.20% (excluding type “9”), as defined in Table 5.

CONCLUSIONS

In this paper, we present the issue of defect detection on electrode surfaces. The source of data obtained by LSP technology are very feasible for inspection task in industrial application. The evaluation of data we performed by comparing

Table 5. Results in detection of defects by merging YOLOv4 and U2S-CNNv2

Type	Number of defects	YOLOv4 U U2S-CNNv2				Improvement			
		True detected		True labeled		True detected		True labeled	
		n	[%]	n	[%]	n	[%]	n	[%]
0	118	103	87.29	102	86.44	0	0.00	0	0.00
1	944	944	100.00	924	97.88	0	0.00	4	0.42
2	1102	1102	100.00	1077	97.73	0	0.00	4	0.36
3	1797	1789	99.55	1741	96.88	3	0.17	8	0.45
4	123	120	97.56	114	92.68	2	1.63	0	0.00
5	133	125	93.98	122	91.73	2	1.50	0	0.00
6	823	696	84.57	668	81.17	59	7.17	133	16.16
7	261	233	89.27	216	82.76	22	8.43	22	8.43
8	996	995	99.90	988	99.20	9	0.90	6	0.60
9	60	23	38.33	16	26.67	6	10.00	0	0.00
Total	6357	6130	96.43	5968	93.88	103	1.62	177	2.78

the new developed version of U2S-CNNv2 method for this task and YOLOv4. It can be observed that almost all defects can be detected by both methods, demonstrating their reliability for defect detection on battery electrodes. Defects can be recognized with both high precision and accuracy. However, challenges remain in detecting small defects, such as thin line defects and small pinholes and foreign particles.

YOLOv4 achieved very good results in defect detection, confirming results published in scientific papers about the deployment of this model. In this study, YOLOv4 had a better performance in detecting small and standard defects. The model's speed and efficiency are also notable, with the ability to analyze 50 fps. The overall detection accuracy was excellent, with 93.82% of defects correctly detected and 91.10% correctly labeled. The primary area for improvement is in detecting long and thin defects.

We described the principle, architecture, and purpose of the new version of U2S-CNNv2. This method combines the advantages of unsupervised and supervised learning for defect detection in industrial applications. Compared to the original method, significant improvements include higher resolution (640×640, compared to the previous 224×224) and a more effective, less complex classifier. The main advantages of this method are its ability to detect defects outside of the training dataset or unknown objects. This confirms results from a similar solutions described in introduction of this paper. Another confirmed advantage is in highlighting the exact areas of defects, not just regions of interest (ROIs). This method achieved a true positive detection rate of 0.9094 and a false positive detection rate of 0.0906, which are slightly better than those of the YOLOv4 model.

However, U2S-CNNv2 has some disadvantages, including sensitivity during the reconstruction process (autoencoder) and slow performance. This slowness is primarily due to the logical principles of the DBSCAN clustering algorithm, which is very slow. The average time tested on 100 samples was 20 ms per sample for the clustering process and 1017 ms per image when using the DBSCAN function from the scikit-learn library. A parallel solution of DBSCAN required 211 ms per sample. The classification of ROIs took 115 ms per sample, resulting in an overall speed rate of 3 fps. This performance was tested on a PC with a GPU 3090 and an AMD Ryzen 9 5900X processor.

By combining both methods, we achieved a defect capture rate of 77.40% to 100% in LSP data from scanned surfaces of LIB electrodes. This scanning technology, in conjunction with the applied detectors, shows great promise as a non-destructive method for ensuring manufacturing quality, making it suitable for industrial applications. This approach guarantees better qualitative parameters for electrodes, thereby guarantee the overall quality of producing LIBs.

Acknowledgements

The authors would like to thank all the colleagues in the institutions for the support in the investigations. The research content of defect detection and classification is funded by Federal Ministry of Education and Research (BMBF) within the competence cluster “Analytics/Quality Assurance (AQua)” in the project “Inline-Klassifizierung von Beschichtungsfehlern zur Ermittlung der Kritikalität in der Elektrodenherstellung (KritBatt)”. The research content of porosity characterization of electrode foil is funded by M-Era.Net with the program of State Ministry of Science and Cultural Affairs of Saxony (SMWK) within the Framework of “RL EuProNet” with characters of 10-06022 in the project “Inline evaluation of Li-ion battery electrode porosity using machine learning algorithms (BattPor)”.

REFERENCES

1. Etienneble A, Besnard N, Adrien J, Tran-Van P, Gautier L, Lestriez B, et al. Quality control tool of electrode coating for lithium-ion batteries based on X-ray radiography. *J Power Sources*. 2015;298:285–91, <https://doi.org/10.1016/j.jpowsour.2015.08.030>.
2. Roadmap Battery Production Equipment 2030 - Update 2023. [cited 2024 Jan 18]. Available from: https://www.researchgate.net/publication/370472366_Roadmap_Battery_Production_Equipment_2030_-_Update_2023
3. Roadmap Batterie-Produktionsmittel 2030 - Update 2023. [cited 2024 Jan 18]. Available from: https://www.researchgate.net/publication/370472223_Roadmap_Batterie-Produktionsmittel_2030_-_Update_2023
4. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit*. 2014;580–7. <https://doi.org/10.1109/CVPR.2014.81>

5. Girshick R. Fast R-CNN. 2015 [cited 2022 Dec 7]. p. 1440–8. Available from: <https://github.com/rbgirshick/>, <https://doi.org/10.1109/ICCV.2015.169>
6. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans Pattern Anal Mach Intell.* 2017;39:1137–49. <https://doi.org/10.1109/TPAMI.2016.2577031>
7. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit.* 2016;2016-December:779–88, <https://doi.org/10.1109/CVPR.2016.91>
8. Li J, Su Z, Geng J, Yin Y. Real-time Detection of Steel Strip Surface Defects Based on Improved YOLO Detection Network. *IFAC-PapersOnLine.* 2018;51:76–81, <https://doi.org/10.1016/j.ifacol.2018.09.412>
9. Yu L, Wang Z, Duan Z. Detecting Gear Surface Defects Using Background-Weakening Method and Convolutional Neural Network. *J Sensors.* 2019;2019, <https://doi.org/10.1155/2019/3140980>
10. Zhang W, Zhang N, Yang W, - al, Song L, Wang Y, et al. LF-YOLOv4: a lightweight detection model for enhancing the fusion of image features of surface defects in lithium batteries. *Meas Sci Technol.* 2023 [cited 2024 Jan 17];35:025005. Available from: <https://iopscience.iop.org/article/10.1088/1361-6501/ad0690>
11. Choudhary N, Clever H, Ludwigs R, Rath M, Gannouni A, Schmetz A, et al. Autonomous Visual Detection of Defects from Battery Electrode Manufacturing. *Adv Intell Syst.* 2022 [cited 2024 Jan 17];4:2200142. Available from: <https://onlinelibrary.wiley.com/doi/full/10.1002/aisy.202200142>
12. Tzelepakis A, Leontaris L, Dimitriou N, Koukidou E, Bollas D, Karamanidis A, et al. Automated defect detection in battery line assembly via deep learning analysis. 2023 [cited 2024 Jan 17];1196–208. Available from: www.eccomasproceedia.org,
13. Ma X, Tang Y, Lu Y, Liu S. Localization Algorithm of R Angle of the tab on Lithium Battery Coating Based on Faster RCNN. *Proc - 2021 Int Conf Comput Inf Sci Artif Intell CISAI 2021.* 2021;391–5, <https://doi.org/10.1109/CISAI54367.2021.00081>
14. Tsai DM, Jen PH. Autoencoder-based anomaly detection for surface defect inspection. *Adv Eng Informatics.* 2021;48:101272, <https://doi.org/10.1016/j.aei.2021.101272>
15. Kähler F, Schmedemann O, Schüppstuhl T. Anomaly detection for industrial surface inspection: application in maintenance of aircraft components. *Procedia CIRP.* 2022;107:246–51, <https://doi.org/10.1016/j.procir.2022.05.197>
16. Niu T, Li B, Li W, Qiu Y, Niu S. Positive-Sample-Based Surface Defect Detection Using Memory-Augmented Adversarial Autoencoders. *IEEE/ASME Trans Mechatronics.* 2022;27:46–57, <https://doi.org/10.1109/TMECH.2021.3058147>
17. Tao X, Zhang D, Ma W, Liu X, Xu D. Automatic metallic surface defect detection and recognition with convolutional neural networks. *Appl Sci.* 2018 [cited 2021 Mar 16];8:1575. Available from: <http://www.mdpi.com/2076-3417/8/9/1575>, <https://doi.org/10.3390/app8091575>
18. Kuric I, Klarák J, Sága M, Císar M, Hajdučík A, Wiecek D. Analysis of the possibilities of tire-defect inspection based on unsupervised learning and deep learning. *Sensors* 2021, 21, 7073. [cited 2022 Jan 11];21:7073. Available from: <https://www.mdpi.com/1424-8220/21/21/7073/htm>, <https://doi.org/10.3390/s21217073>
19. Klarák J, Andok R, Malík P, Kuric I, Ritomský M, Klačková I, et al. From Anomaly Detection to Defect Classification. *Sensors* 2024, 24, 429. [cited 2024 Feb 6];24:429. Available from: <https://www.mdpi.com/1424-8220/24/2/429/htm>, <https://doi.org/10.3390/s24020429>
20. Tung PY, Sheikh HA, Ball M, Nabiei F, Harrison RJ. SIGMA: Spectral Interpretation Using Gaussian Mixtures and Autoencoder. *Geochemistry, Geophys Geosystems.* 2023;24, <https://doi.org/10.1029/2022GC010530>
21. GitHub - poyentung/sigma: Python code for phase identification and spectrum analysis of energy dispersive x-ray spectroscopy (EDS). [cited 2024 Jul 19]. Available from: <https://github.com/poyentung/sigma?tab=readme-ov-file>
22. Chen L, Cikalova U, Münch S, Stüwe T, Dam V, Bendjus B. Artificial intelligence – A solution for inline characterization of li-ion batteries. [cited 2024 Jan 18]; Available from: <https://www.ndt.net/?id=27662>
23. F. Kux, U. Cikalova, L. Chen, J. Henschel BB. An Innovative Approach Using Laser Speckle Photometry for Efficient and Accurate Surface Defect Detection. 14th Ed Int Conf Adv Lithium Batter Automob Appl (ABAA-14),. 2023.
24. Mohanty D, Hockaday E, Li J, Hensley DK, Daniel C, Wood DL. Effect of electrode manufacturing defects on electrochemical performance of lithium-ion batteries: Cognizance of the battery failure sources. *J Power Sources.* 2016;312:70–9, <https://doi.org/10.1016/j.jpowsour.2016.02.007>
25. Ester M, Kriegel H-P, Sander J, Xu X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. 1996. Available from: www.aaii.org,
26. Chollet F. Xception: Deep Learning With Depth-wise Separable Convolutions. 2017. p. 1251–8, <https://doi.org/10.1109/CVPR.2017.195>
27. Clustering – scikit-learn 0.24.0 documentation. [cited 2021 Jan 14]. Available from: <https://scikit-learn.org/stable/modules/clustering.html#dbscan>