processes

MDPI

Article

Developing an Anomaly Detection System for Automatic Defective Products' Inspection

Yu-Hsin Hung

Department of Industrial Engineering and Management, National Yunlin University of Science and Technology, Yunlin 64002, Taiwan; hungyh@yuntech.edu.tw

Abstract: Since unqualified products cause enterprise revenue losses, product inspection is essential for maintaining manufacturing quality. An automated optical inspection (AOI) system is an efficient tool for product inspection, providing a convenient interface for users to view their products of interest. Specifically, in the screw manufacturing industry, the conventional methods are the human visual inspection of the product and for the inspector to view the product image displayed on the dashboard of the AOI system. However, despite the inspector and the approach used, inspection results strongly depend on the inspector's experience. Moreover, machine learning algorithms could improve the efficiency of human visual inspection, thus addressing the above problem. Based on these facts, we improved anomaly detection efficiency during product inspection, using product image data from the AOI system to obtain valuable information. This study notably used the visual geometry group network, Inception V3, and Xception algorithms to detect qualified and unqualified products during product image analytics. Therefore, we considered that the analyzed results could be integrated into a proposed cloud system for human–machine interaction. Thus, administrators can receive reminders concerning the anomaly-inspected notification through the proposed cloud system, comprising a message queuing telemetry transport protocol, an application programming interface, and a cloud dashboard. From the experimental results, the above-mentioned algorithms had more than 93% accuracy, especially Xception, which had a better performance during the defective type classification. From our study, the proposed system can successfully apply the obtained data in data communication, anomaly dashboards, and anomaly notifications.

Keywords: deep learning; anomaly detection; image processing

Citation: Hung, Y.-H. Developing an Anomaly Detection System for Automatic Defective Products' Inspection. *Processes* **2022**, *10*, 1476. https://doi.org/10.3390/pr10081476

Academic Editors: Piotr Gierlak, Satish Kumar and Arunkumar Bongale

Received: 27 May 2022
Accepted: 26 July 2022
Published: 27 July 2022

1. Introduction

A screw product comprises a cap head and a thread. Since screws are critical equipment components, a faulty one can cause major problems, such as the equipment failing to operate normally. Hence, screw inspection is critical for its manufacturer. In this study, a screw manufacturer served as a case subject. Then, the screw inspection process was divided into four stages. The first stage was to inspect the surface of the cap head; the second was to inspect the screw's roundness; the third was to inspect the screw thread; and the fourth was to inspect the screw eccentricity (Figure 1). An automated optical inspection (AOI) system is a visual tool that uses a camera to automatically photograph a product during its inspection. In this case, the inspector also watched the photographed image on an AOI dashboard to further detect any defective surface on the cap head. Notwithstanding, the inspector's subjective assessment still determines whether or not the product is defective, and sometimes the inspector may incorrectly conduct a defective inspection. Therefore, this study focused on improving the cap head surface's inspection efficiency.

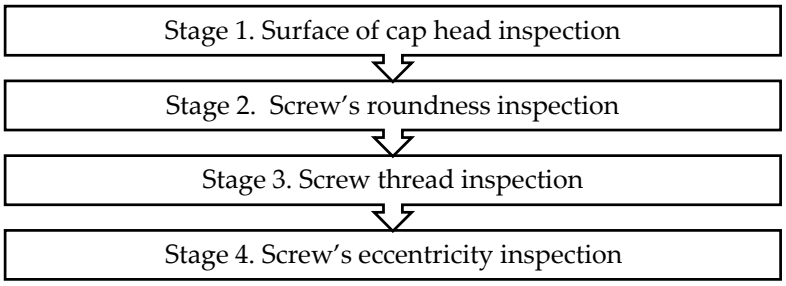


Figure 1. Full inspection workflow.

Industry 4.0 aims to enable digitalization, intelligentization, interconnectivity, and automation using cloud technologies, cyber-physical systems (CPS), artificial intelligence (AI), and machine learning (ML). In CPS, internet of things (IoT)-related tools (e.g., sensors, internet-connected devices, and software applications) are used to smart wire objects for industrial applications. Consequently, massive industrial data can be obtained within the CPS environment during internet-connected manufacturing. Previously, Kamat and Sugandhi [1] investigated the correlation between anomaly detection and predictive maintenance in Industry 4.0. They observed that since a human visual inspection hardly identifies anomaly events and rare observations for anomaly detection, data collected from CPS could be analyzed to automatically find defected patterns. Alternatively, according to IBM, automation includes information technology, business processes, and other types of automation [2]. On this basis, another study reported that data-driven anomaly detection could improve inspection efficiency [3], and cloud technologies have recently been widely applied to automation applications. Moreover, automating all or part of the manual tasks using specialized software and methodologies can enable auto-notification while detecting an anomaly. Hence, cloud technologies, such as the message queuing telemetry transport (MQTT) protocol, application programming interface (API), and web programming, have recently been applied widely to automatically function in human and machine communication.

Although improving product quality in the screw manufacturing process was previously demonstrated using the above-mentioned technologies, this study used the visual geometry group network (VGG-16), Inception V3, and Xception algorithms for real industrial image datasets. Then, the MQTT protocol, API, and web programming methods were employed in the proposed system to develop cloud dashboards, auto-notifications, and data communication systems. As a result, information could be quickly communicated and visualized on a cloud platform. Based on the above background, this study also detected anomaly product quality using deep learning approaches, surveyed the difference in classification performance between the different convolutional networks and training data sizes, enabled real-time communication and visualization by developing a communication protocol to activate real-time supervisory control and data acquisition (SCADA), and automatically transferred the inspection result from the product to their users via a Bot and IoT protocol. Subsequently, this study detected the defective surface of screw products and applied an integrated trained module for further use. Section 2 reviews the anomaly detection-related literature and discusses the relationship between anomaly detection, image processing, and predictive maintenance, whereas Section 3 describes the data analytics procedures, some primary deep learning approaches, dataset and evaluation criteria, and the proposed system. Next, Section 4 presents experiment-related information, such as those of the use case, dataset description, and evaluation criteria and Section 5 presents the findings of this investigation. Section 6 discusses the impact of cloud data analytics and related matters. Finally, Section 7 concludes this study, describing its significance and limitations and providing suggestions for future work.

2. Related Work

Related literature demonstrates that machine learning, data mining, mathematical methods, and deep learning approaches provide users with a convenient way to ana-

lyze data from manufacturing sites for anomaly detection (Table 1). Table 1 lists the current approaches for anomaly detection. Some anomaly detection-related research used machine-learning approaches. For example, Winters et al. [4] investigated control charts and autoregressive models to detect anomalies for predictive maintenance; Erdmann [5] used an unsupervised approach to recognize anomalous sensor data for predictive maintenance; Minarini [6] proposed a framework for detecting anomalies in log-based predictive maintenance; Alaoui-Belghiti et al. [7] proposed unsupervised online methods using optimal transport to recognize the anomalies for predictive maintenance; Farbiz et al. [8] proposed cognitive analytics with unsupervised learning to predict the machine status for equipment health maintenance; Carrasco et al. [9] proposed a framework for temporal unsupervised algorithm evaluation, which detects time-series analytics early. In addition, data mining approaches such as correlation analysis [10] have been applied to equipment maintenance. For instance, Perini [11] used the Markov Chain and autoencoder for off-road vehicle maintenance.

Table 1. The main anomaly detection-related approaches in predictive maintenance.

Method	Studies
CNN	[12,13]
NN-related (e.g., ANN, autoencoder)	[11,14,15]
Machine learning	[4–9,16]
Data mining (correlation analysis, Markov Chain)	[10,11]
Mathematics (e.g., fuzzy)	[17,18]

Recently, anomaly detection using images has attracted many researchers [19]. Therefore, this study applied image data in data analytics for anomaly detection. First, we used the Publish or Perish software to survey emerging studies' time trends and disciplinary distribution and related literature were retrieved. Notably, this study included keywords from top-cited articles to present the current technological trends: "deep learning," "unsupervised learning", "semi-unsupervised learning", "supervised learning", and "restructure". These keywords are high-frequency for anomaly detection-related topics concerned with image issues. Then, the retrieved papers were sorted based on their average citations per year and total citations. Table 2 lists the main methods identified.

Table 2. The main anomaly detection approaches from 2017 to 2022 that used images.

Method	Studies
Generative adversarial networks (GAN)	[20,21]
Convolutional neural network (CNN)	[22–25]
Neural networks (NN)-related	[26,27]
Image process technologies	[28–32]
Mathematics	[33,34]

Most findings have demonstrated that convolutional neural network (CNN) and image process approaches are the core approaches that drive research associated with anomaly detection, especially in the medical, manufacturing, and transportation fields. The result of our literature retrieval also showed that deep learning approaches, such as the ImageNet dataset, cancer diagnosis, and defective product detection, exhibited high accuracies in some cases. As a result, deep learning for anomaly detection has gained massive popularity in anomaly detection-related research. Of these deep learning approaches, CNN can learn directly from data. Previously, Haselmann et al. [22] used a deep CNN to recognize an anomalous surface. In their study, the CNN framework extracted

the features of the cell image through a self-learning capability. Similarly, while Xu et al. [23] used a hierarchical CNN to detect anomalous chest X-ray images, Nguyen et al. [25] used a deep CNN to restructure images and detect the anomalous region of magnetic resonance imaging scans. Khan et al. [24] also used a deep CNN to recognize anomalous spectrograms. Accordingly, generative adversarial networks (GANs) are another deep-learning method that uses an ensemble neural network model (e.g., CNN) for automatically discovering and learning the regularities or patterns in input data. GANs are also usually used in data augmentation. In previous studies, while Deecke [20] used the GAN in visual inspection for anomaly detection, Berg et al. [35] used it to identify anomalous contaminated image data. Zhou et al. [21] also used the GAN to identify anomalous retinal optical coherence tomography images. The above studies show that deep learning approaches achieve state-of-the-art performance in image anomaly detection using images. However, for the expected reason, the original image data have noise. For example, Chithirala et al. [36] indicated that image data with noise may influence the deep learning analysis result. Therefore, the original image dataset must be preprocessed. A study reported that while the filters used in image processing can enhance the image or edge features [37], they also effectively remove noise and blurred areas in images. Hence, reducing the noise in an image is the main preprocessing stage before being imported into the deep learning model for training [38]. It has also been reported that while performing image denoising, sufficiently retaining the features in the image is necessary, which is a critical aspect during preprocessing.

Some studies have proposed image processing technologies: mathematical methods or neuron networks (NNs) to analyze image data for identifying an anomaly. For example, Zhang et al. [28] proposed a three-stage tensor decomposition and divided it into three steps for detecting anomalies from hyperspectral images. Similarly, while Ayhan et al. [29] proposed a two-step alignment approach for multispectral image anomaly detection, Cohen and Hoshen [39] used correspondences based on a multiresolution feature pyramid in the sub-image detection. In addition, Mishra et al. [40] applied the transformer method in image localization and anomaly detection. However, Mishra et al. [41] proposed a deep reconstruction-based pyramidal approach to exact image features for anomaly detection. Likewise, although Zhuang et al. [30] proposed a robust hyperspectral image denoiser to process its anomaly detection, Müller et al. [27] used NN for feature extraction. Alternatively, another study employed the support vector machine and Gaussian mixture models to recognize anomalous images Müller et al. [27]. Accordingly, while Cozzolino and Verdoliva proposed an autoencoder method for recognizing the anomalous-spliced image region, Li et al. [33] proposed a Gaussian distribution model estimation method to identify the anomalous image descriptors in traffic videos. Furthermore, although Verdoj and Grangetto [31] used the graph Fourier transform to enhance the efficacy of the Reed–Xiaoli detector for medical image anomaly detection, Wang et al. [34] used the discrete probability model and deep autoregressive module for image anomaly detection. Lastly, Vojir et al. [32] also proposed a reconstruction module to identify unknown objects for autonomous driving anomaly detection.

3. Methodology

Based on this study's aims, we designed an automatic inspection process for anomaly detection from data analytics and system design aspects (Figure 2). The first part applied data analytics in anomaly detection to stepwisely process an existing image according to the standard data analytics procedure. This study preprocessed the inspection data following the standard data analytics procedure. The standard data analytics procedure involves preprocessing, feature extraction, training, evaluation of models, and extensive application, as shown below. However, the second part is concerned with system implementation. As a result, this study proposes a real-time anomaly detection system.

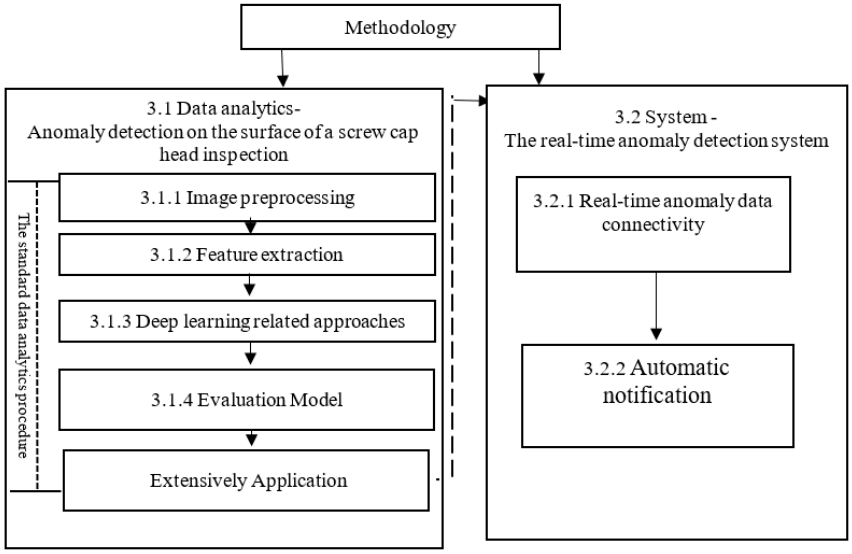


Figure 2. The methodology framework.

3.1. Data Analytics for Anomaly Detection

3.1.1. Image Preprocessing

Noise is usually generated in an image based on sudden disturbances from a photographic sensor or dust and iron filings on a product's surface during manufacturing. As a result, the noise is presented sparsely with black and white pixels, also known as salt-and-pepper noise. Image filter techniques can effectively remove noise from images, including image edges and features, resulting in an enhancement that improves image quality. For example, bilateral filters can adjust each pixel's intensity with a weighted average of intensity values from neighboring pixels. Notably, this study used a bilateral filter to reduce the noise of the screw surface's inspection image, and weight estimation was based on the Gaussian distribution, as shown below [Equation (1)].

$$g(x) = \left(f \times G^8\right)(x) = \int_R f(y) G^8(x - y) dy \quad (1)$$

where the weight for f(y) refers to the spatial distance $\| x - y \|$, which equals G(x − y). Notably, since the bilateral filter adds a weighting term according to the f(y) − f(x) distance, the weights explicitly depend on the image values. Thus, it requires explicit normalization, such that the aggregate of all weights equals 1.

3.1.2. Feature Extraction

After reducing the noise in the image, the preprocessed image data can be extracted as a feature using feature extraction technologies. Papageorgiou et al. [42] previously proposed an alternate framework using Haar wavelets instead of the conventional image intensities. Subsequently, Viola and Jones realized the idea of using Haar wavelets and developed the so-called Haar-like features [43]. Since the Haar-like feature is the main feature extraction technology applied in face and object detection [43], it was first extracted by taking rectangular regions from the image and dividing them into multiple parts. Next, the pixel intensities were aggregated in each region, after which the difference between these aggregations was calculated. The representation of features is often visualized as white and black adjacent rectangles. Additionally, as shown below (Figure 3a), the Haar-like features have three edge characteristics: the line, center, and four-rectangle features. Since the input image is notably divided into subsections, the Haar-like feature of each subsection can be extracted by moving the window of the target size over the input image. This study used the Haar-like feature to obtain contour features for each screw image (Figure 3b).

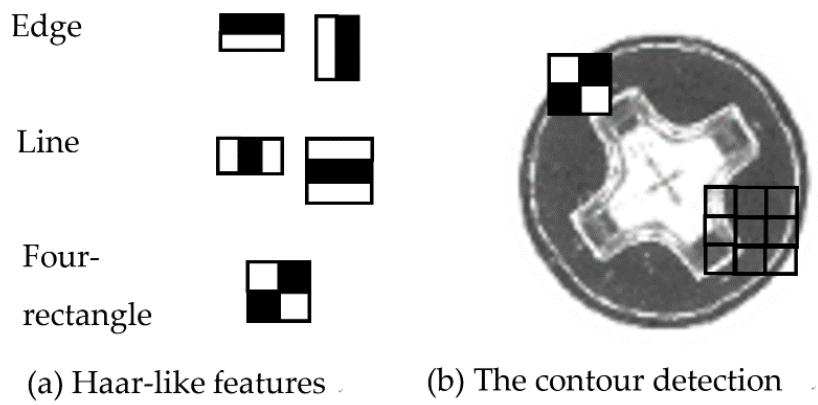


Figure 3. Haar-like features applied on the contour detection of the screw's cap head.

3.1.3. Preliminaries Model-Deep Learning Neural Network

CNN is a deep learning approach that outperforms other deep learning approaches in image processing. It comprises three layers: the convolutional, pooling, and full-connection layers. The convolutional layer is first used to extract the features of this network. It then learns to find spatial features in an input image. Subsequently, pooling layers combine the outputs of neuron clusters in one layer into a single neuron in the next layer to reduce data dimensions. Finally, fully connected layers connect neurons in one layer to every neuron in another layer. Since the convolution network's framework consists of three main structures: a local receptive field, weight sharing, and pooling, CNN algorithms can be applied directly to the original image process and the convolution network, greatly reducing the training parameters of neural networks. As a result, the different CNN architectures produce varying analysis results, depending on the diverse type of training image data. Interestingly, this study used the three CNN-based networks: VGG-16, Inception V3, and Xception, to detect the defective surface of the screw cap head. The three models are described below.

(1) VGG-16

Simonyan and Zisserman [44] initially proposed the VGG network architecture. VGG-16 refers to VGG models with 16 layers. Its architecture consists of five blocks of convolutional layers that combine two previous blocks with two fully connected layers with other blocks consisting of three fully connected layers each. Convolutional layers use 3×3 kernels with a stride of 1 and padding of 1 to ensure that each activation map retains the same spatial dimensions as the previous layer. Subsequently, a rectified linear unit (ReLU) activation is performed immediately after each convolution, after which a max-pooling operation is used at the end of each block to reduce the spatial dimension. Two fully connected layers with 4096 ReLU-activated units are notably used before the final 1000 fully connected softmax layer (Figure 4).

(2) Inception V3

Compared with the VGG network, the Inception network comprises symmetric/asymmetric blocks of convolutional layers, such as convolutions, max pooling, average pooling, filter concatenations, dropouts, and fully connected layers. As a result, the inception network's architecture explicitly factors in several operations [45]. First, each convolution kernel is tasked with simultaneously mapping cross-channel and spatial correlations. Then, the softmax function is used to estimate the loss (Figure 5).

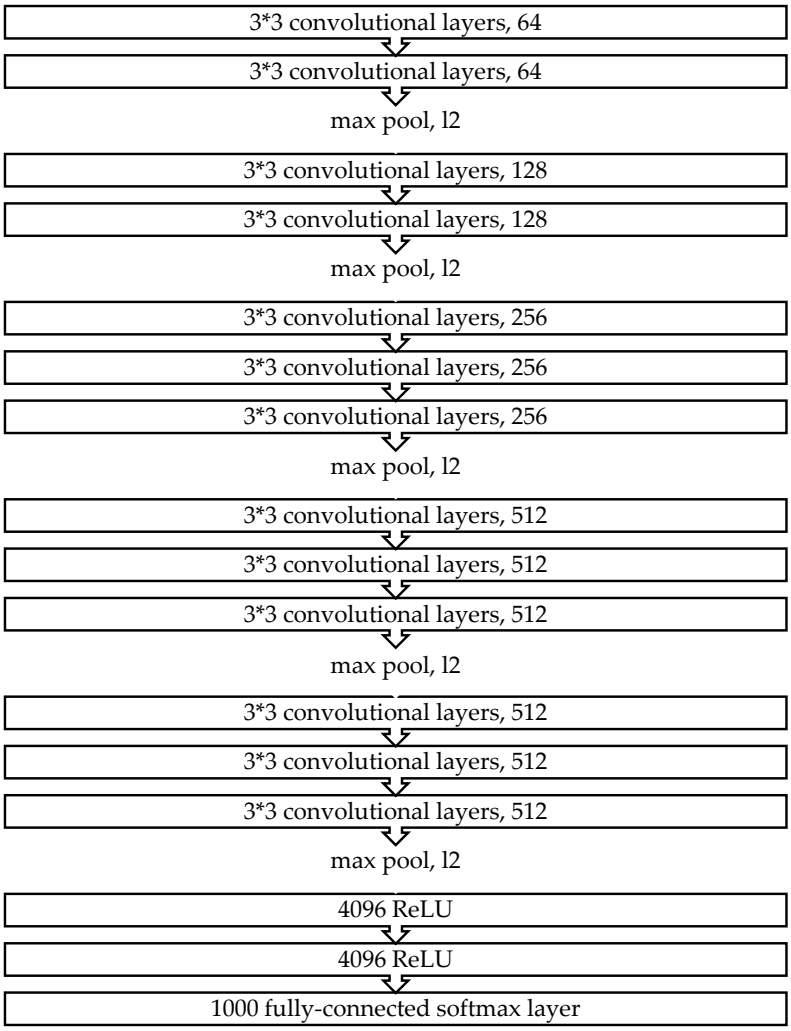


Figure 4. The VGG-16 architecture.

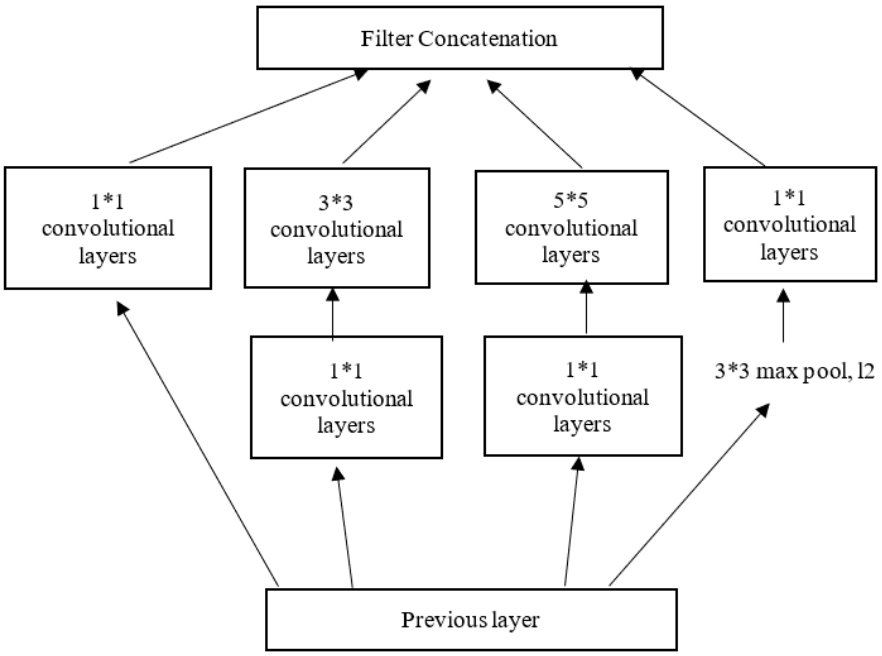


Figure 5. The Inception V3 architecture.

(3) Xception

Chollet proposed the Xception architecture in 2017, based on the Inception architecture. Figure 6 demonstrates the framework of Xception [46]. Compared with Inception, the Xception model uses depth-wise separable convolutions, which include a spatial convolution performed independently for each channel and a 1×1 convolution across channels. Subsequently, Xception separately maps the spatial correlations for each output channel from the cross-channel correlation across a two- and one-dimensional space.

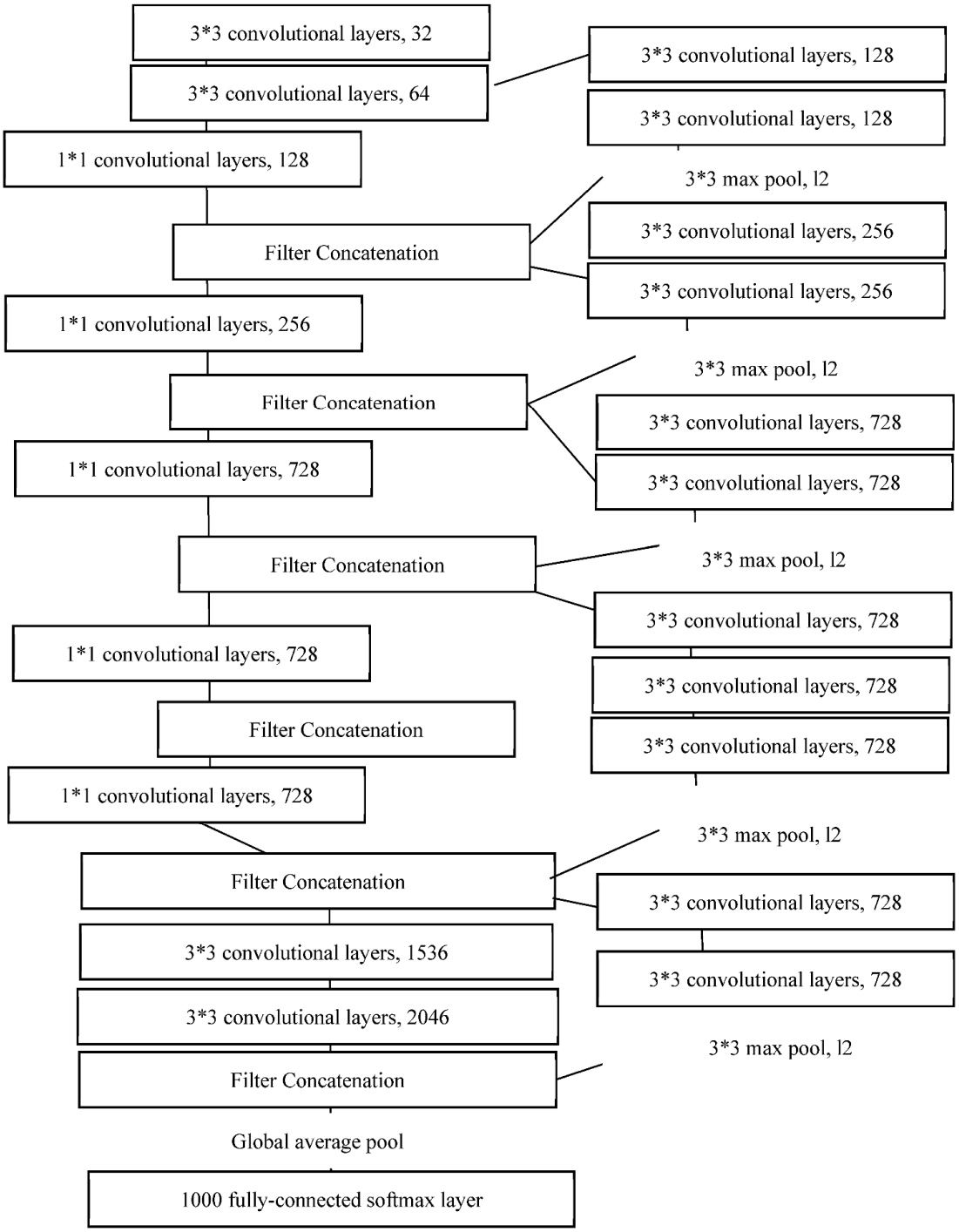


Figure 6. The Xception architecture.

3.1.4. Evaluation Criteria for a Model

A confusion matrix is a two-dimensional matrix used to describe the performance of a classification model (Figure 7). The evaluation item is based on a two-dimensional prediction and actual result, which can be classified into four types: false negatives (FN), true negatives (TN), false positives (FP), and true positives (TP). Subsequently, FN, TN, FP, and TP are used to estimate the accuracy, precision, recall rate, and F1 score. Each criterion is discussed in detail below.

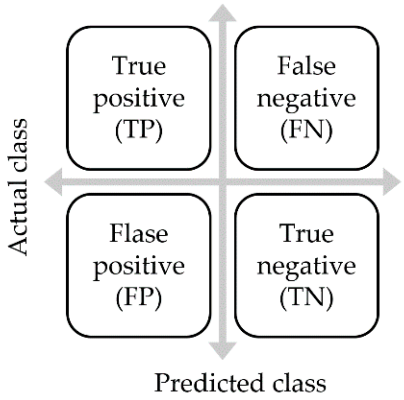


Figure 7. Schematic showing a confusion matrix.

- Accuracy rate

The accuracy rate refers to the proportion of correct predictions. It is estimated by dividing the aggregation of correct assessments by the sum of all assessments. The accuracy rate is also the degree of closeness to the correct value and can be regarded as the degree of veracity. The accuracy formula parameters include the number of FNs, TN (NUM_{TN}), FP (NUM_{FP}), and TP (NUM_{TP}) [Equation (2)].

$$\text{Accuracy rate} = \frac{NUM_{TP}+NUM_{TN}}{NUM_{TP}+NUM_{TN}+NUM_{FP}+NUM_{FN}} \quad (2)$$

- Precision rate (positive predictive value)

Accuracy and precision are the two main criteria for model evaluation. The precision rate indicates the accuracy of predictions for a positive assessment and can be regarded as the degree of reproducibility. As shown below, its value is estimated by dividing the actual positives by the sum of all positive assessments.

$$\text{Precision rate} = \frac{NUM_{TP}}{NUM_{TP}+NUM_{FP}} \quad (3)$$

- Recall rate (true positive rate)

The recall rate is a statistical measure of the performance of a classification function. It is estimated as the proportion of actual positives identified correctly in an overall assessment that should have been predicted as positive [Equation (4)].

$$\text{Recall rate} = \frac{NUM_{TP}}{NUM_{TP}+\ NUM_{FN}} \quad (4)$$

- F1-score

The F1-score estimation combines precision rate and recall rate in its model. It refers to the harmonic mean of the model's precision and recall rates [Equation (5)].

$$\text{Recall rate} = \frac{2 \times (Precision\ rate \times Recall\ rate)}{Precision\ rate\ + Recall\ rate} \quad (5)$$

3.2. Proposed Real-Time Anomaly Detection System

This study proposed an approach for users to manage anomaly detection. Therefore, the MQTT protocol and API were applied to convey the data to a human. Figure 8 illustrates the architecture of the proposed system. First, we observed that the MQTT transfer of the real-time message, based on a specified topic, enabled the real-time data connectivity network. Then, the API receives the message and report generation, notification, and alarm automatically, producing results on the SCADA dashboard. Consequently, the proposed system eases the user interaction with the system's configuration and data connectivity.

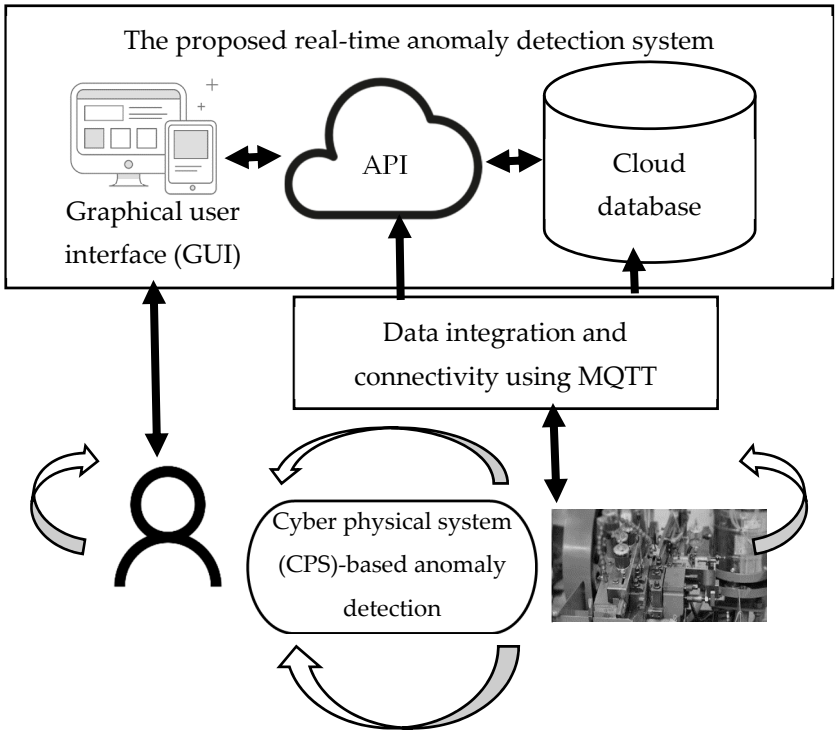


Figure 8. Architecture of the proposed real-time anomaly detection system.

3.2.1. Real-Time Anomaly Data Connectivity via the MQTT Protocol

This study used the MQTT protocol to create a specification format for communicating modules using a supervisory control module's protocol instruction (Figure 9). IBM previously developed the MQTT, which has become an ISO standard (ISO/IEC PRF 20922) messaging protocol when using objects, devices, and sensors on a transmission control protocol/internet protocol (TCP/IP). It is applied to ensure a stable connection environment, using the broker as a hub for exchanging messages with the client, and providing flexible content for users to create their topics and messages. As a result, MQTT brokers can receive messages on different topics from different clients. First, the MQTT client sends a message using the "publish" instruction and receives the message by subscribing to the MQTT topic. Then, the broker filters the messages and dispatches them according to the topic. Notably, each message must contain a topic, which the broker could use to forward the message to any client who subscribes to this topic. In this study, the module was connected via the MQTT broker, since each anomaly event had its topic and message content in the JavaScript object notation (JSON) format (see Table 3). Then, real-time anomaly event information was sent to the real-time dashboard to update the value, during which the system receives the real-time data information from the central broker. Different users have their alarm or notifying threshold values for making such decisions. Following these customized requirements, the current state of the manufacturing industry can be supervised.

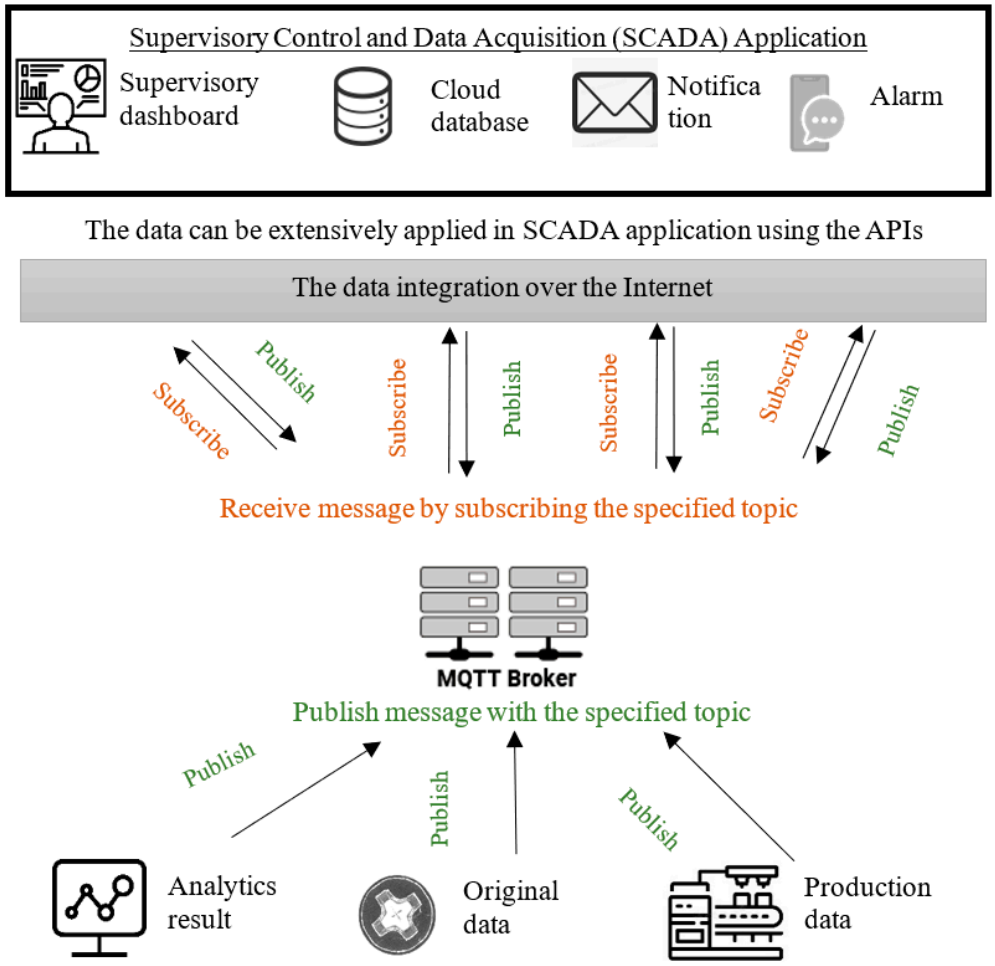


Figure 9. Real-time SCADA using MQTT.

3.2.2. Automatic Notification for Anomaly Detection

This study used a representational state transfer API (RESTful API) to develop a user notification mechanism. The RESTful API is based on the RESTful architecture. Specifically, RESTful is a communication architecture that is commonly used in the development of web services (WS). The API here is the middleware between the user and the service. Accordingly, the RESTful API can be used to access the WS by sending a hypertext transfer protocol (HTTP) request. Notably, RESTful APIs use request types to access the application, which includes "GET", "POST", "DELETE", and "PUT". These request types refer to retrieving, passing, deleting, and updating instructions for data applications. Specifically, while the "GET"-type request is applied to retrieve resources, the "POST"-type request is applied to pass and create resources. Moreover, while the "DELETE"-type request is used to delete information, the "PUT"-type request is applied to pass information, change the state, or update information. Based on this background, we built an API server and created a query argument and response content. Investigations revealed that while the proposed APIs could be managed using Swagger, the analytics module provided the APIs, allowing users to access the trained model directly through HTTP. We also observed that since this study consistently uses the JSON format as the API response format, it could be applied extensively to WS or mobile applications. Additionally, in this study, the received argument while calling API and the response result of API must encode and decode using message-digest algorithm 5 (MD5) separately. Hence, we protected the API connectivity and communicating content using MD5 encryption and decryption, and as a result, this study proposed a report API notification (Figure 10). Therefore, a user can configure the threshold value of an alarm and know if the value exceeds the threshold value. After the

API receives the real-time MQTT message, it automatically generates an anomaly report and sends the alarm notification via email and LINE Bot.

Table 3. Anomaly detection MQTT protocol.

Topic	Message
iot/temp/factory	"device" { defectiveProductCount: number}
iot/defectedProductCount/{inspection_stage}	"product" { name: "product name," defectedTypeNo.: 1, count: number }

iot/anomalyReport/{inspection_stage}	"header"
	{ rptname: "anomaly detection report,"
	reportTime: timeStamp
	}
	"report body"
	{ " product"
	{
	alarmThrehold,
	name: "product name,"
	defectedTypeNo.: 1,
	count: number,
	imageSrc:[image1Url, image2Url . . .]
	}
	"productionOwner"
	{
	name:" employee's name,"
	empId:" employee's id,"
	email:" employee's email,"
	phone:" employee's phone,"
	}
	},
	"footer"
	{ otherNotification:discription
	}

iot/anomalyBotConnect/{inspection_stage}	"botInfo"
	{ accessToken: "anomaly detection report,"
	userId: timeStamp
	}
	"message body"
	{
	" product"
	{
	alarmThrehold: number,
	name: "product name,"
	defectedTypeNo.: 1,
	count: number,
	description: other notification information
	}
	}

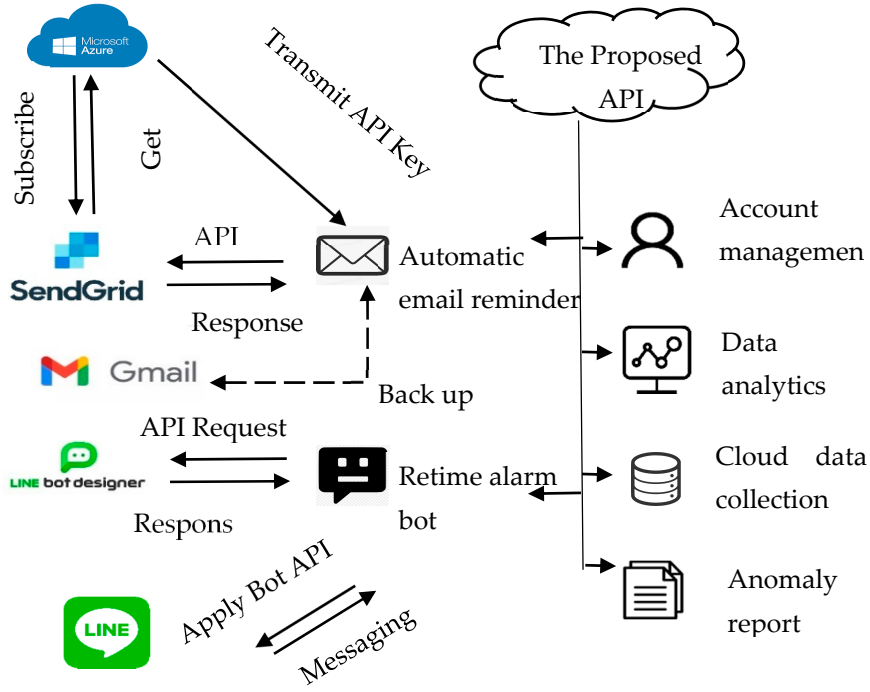


Figure 10. Architecture of automatically generated reports and reminders via API.

4. Experiment

The experiment comprised image data analytics and system development aspects. Figure 11 shows the experimentation procedure. First, the data were preprocessed, feature extracted, split, and trained sequentially during data analytics. Furthermore, during system development, the front-end, such as the user interface, and the back-end (e.g., the database, API, data communication), were implemented with a corresponding programming language and existing third-party APIs.

4.1. Use Case

This study selected a screw manufacturing factory as its real case environment to demonstrate how the proposed method uses data analytics in anomaly detection. The proposed method could identify the defection type during the surface screw cap head inspection. Subsequently, since our use case focused on improving product quality, we used the AOI system for product inspection. The use case provided the screw product and developed an AOI system to realize the digital transformation. Small-sized manufacturing enterprises may face complex digital transformation (e.g., funds for purchasing advanced equipment and complicated software integration) to fulfill Industry 4.0. Thus, this study automatized and intellectualized the inspection process based on an existing system, using data from this AOI system.

4.2. Dataset Description

Several sources have been adopted for data analysis to address the inspection efficiency issue and potential mistakes from human operations in the screw manufacturing industry. To this end, we used the screw cap head data as a source to address the anomaly problem. The image data were obtained from the AOI system. Figure 12 shows the experiment's captured images used as training dataset instances. These values represent defection types from 1 to 5, respectively (Figure 12). In the surface inspection stage, each screw product first used a photographic sensor to vertically capture one screw's cap head image from above. The resolution of the image is 140 × 140 pixels, and in a bitmap (.bmp) format. Furthermore, the size of each captured image is 57.4 KB. Then, based on this background, the inspectors viewed each screw cap head image on the AOI system's dashboard and judged whether the

surface of the screw's cap head was qualified according to their experience. The captured image was finally collected into an AOI system.

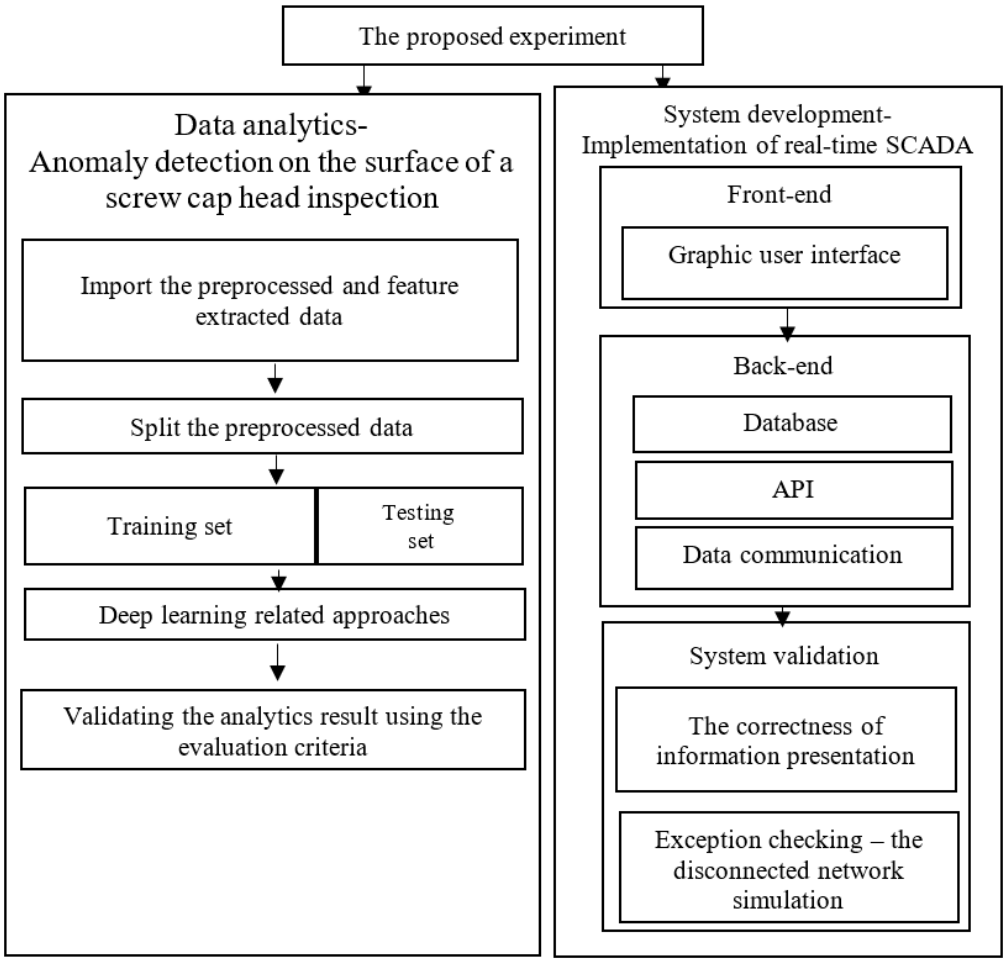


Figure 11. Schematic showing the experimental procedure.

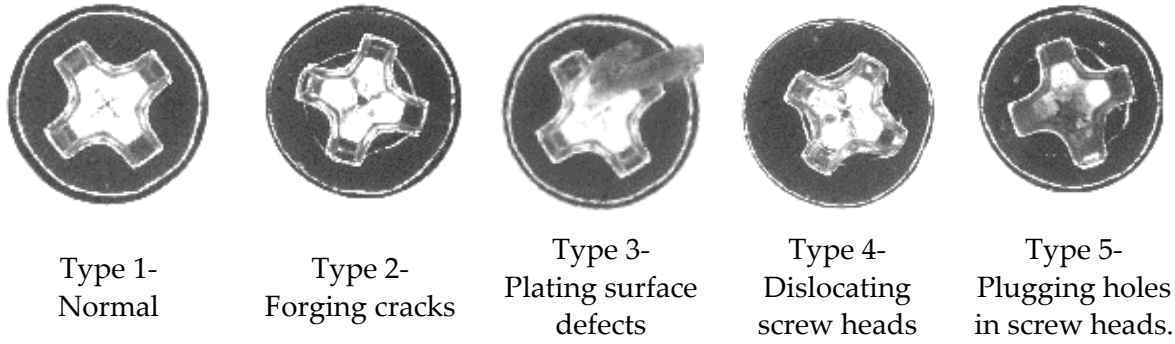


Figure 12. Instances of defective products.

The identifying target was the defection type, e.g., normal, forging cracks, plating surface defects, dislocation of screw heads, and plugging holes in screw heads. However, this use case was an anomaly detection case that identifies the anomaly type of a screw product following inspection of the surface of its cap head. Although the dataset comprised 5500 instances with image features, the training sample size was related to the classification performance. Thus, this study divided the training dataset into different batch sizes and investigated the difference between the training sizes. Table 4 describes the datasets.

Table 4. List of datasets.

The Analytics Target	Manufacturing Process Stage	Description	Total Instances	Training Batch Size	Test Size
The five defection types	The surface of the cap head	Data with different image features for each instance	5500	1000 2000 3000 4000 5000	500

4.3. Experimental Environment and Tools

The experimental data used in this study was the image-type data. Therefore, it needed to be collaboratively used with the graphics processing unit rather than single-use with the central processing unit. Furthermore, the image data needed a larger physical space and cache size. Thus, specified random access memory and a hard disk were used to support the data analytics. Table 5 demonstrates the specifications of the experiment environment. While the experiment was run in an environment consisting of the processing, storage, and graphic process units (see Table 5), this study used Python programming to process and analyze the image data. Finally, node.js programming language was used to develop communication functions, whereas the .NET core and JavaScript were used to develop the user interface and access the mailing and LINE Bot APIs.

Table 5. Description of the experimental environment.

Main Part	Specification
Central processing unit	AMD (8-Core) (4.7 G)
Main board	Asus TUF X570-PLUS(ATX)
Random access memory	Kingston (128 GB)
Hard disk	WD SN750SE 500G/Gen4
Graphics processing unit	NVIDIA RTX3080-10G
Power supply unit	Asus ROG STRIX (1000 W)

5. Results

This study describes the results from the viewpoint of data analytics and system implementation. Our investigations showed the different deep-learning approach results during the data analytics. The batch size was also considered, after which we investigated the relevance between the training data size and classification performances. Finally, we demonstrated how the information could communicate between data analytics and the user interface, using Bots, API, and MQTT in the system implementation.

5.1. Anomaly Detection on the Surface of a Screw Cap Head during Inspections

This study used preprocessed data to identify the type of defection on the surface of the screw cap head (Tables 6–8). However, the manufacturing data might be a difficult source for sufficient data collection during the off-season or due to uncertain issues. Hence, this study investigated the effect of the training data size on classification performance. Investigations revealed that Inception and Xception had higher accuracy than VGG-16 when using 1000 instances. However, VGG-16 had higher precision using 2000 instances. Furthermore, while Inception had similar accuracy and precision between 4000 and 5000 training instances, Xception had a similar result because its framework network is similar to Inception's. Plating surface defects ($Precision_{plating}$ = 0.85) and plugging holes in screw heads ($Precision_{plugging}$ = 0.91) were two detection types that were difficult to identify in the three deep learning models.

Table 6. VGG-16 results.

Training Data Size	1000 (200 each type)	2000 (400 each type)	3000 (600 each type)	4000 (800 each type)	5000 (1000 each type)
Avg. accuracy	0.899	0.920	0.924	0.944	0.949
Avg. precision	0.908	0.930	0.924	0.950	0.954
Avg. recall rate	0.900	0.920	0.924	0.944	0.950
Avg. F1-score	0.896	0.918	0.924	0.944	0.950

Table 7. Inception results.

Training Data Size	1000 (200 each type)	2000 (400 each type)	3000 (600 each type)	4000 (800 each type)	5000 (1000 each type)
Avg. accuracy	0.928	0.930	0.932	0.958	0.960
Avg. precision	0.936	0.936	0.936	0.960	0.962
Avg. recall rate	0.928	0.930	0.932	0.958	0.960
Avg. F1-score	0.926	0.928	0.932	0.960	0.960

Table 8. Xception results.

Training Data Size	1000 (200 each type)	2000 (400 each type)	3000 (600 each type)	4000 (800 each type)	5000 (1000 each type)
Avg. accuracy	0.926	0.922	0.930	0.956	0.958
Avg. precision	0.938	0.934	0.934	0.960	0.962
Avg. recall rate	0.926	0.922	0.930	0.956	0.958
Avg. F1-score	0.926	0.922	0.930	0.956	0.960

Subsequently, this study used analysis of variance to estimate the correlation between different training data sizes and evaluation criteria. The results showed that 5000 instances significantly improved accuracy over other training data sizes. Therefore, while VGG-16's result showed that using 5000 instances improved precision more than using other training data sizes (see Table 8), Xception's and Inception's results showed no significant difference in accuracy between using 4000 and 5000 instances (Tables 6–8).

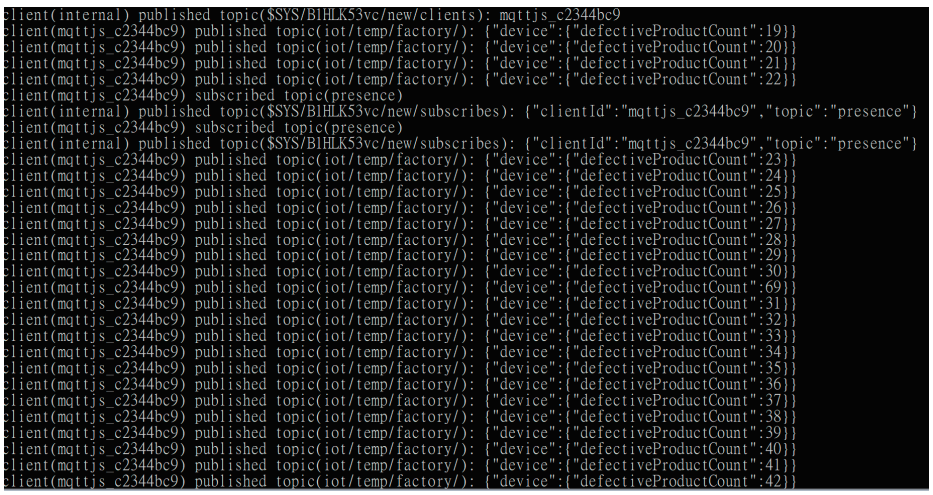
Summarily, the results obtained using the deep learning methods were compared with those of VGG-16, Inception, and Xception, indicating that the deep learning network achieved more than 94% accuracy on the surface of the screw cap head. We also observed that although the average value of the F1 score using VGG-16, Inception V3, and Xception was over 0.94, the result represented the training model, which had great precision and recall. Furthermore, the training model using the above algorithms had an average of 94% recall rate, meaning that the training model successfully retrieved a higher fraction of defective instances. As a result, Inception V-3 and Xception had better classification performances, as evaluated by the accuracy, precision, recall rate, and F1-score.

5.2. Implementation of Real-Time SCADA

The trained model, deployable using various deep learning approaches, was deployed as a RESTful API that can be used for anomaly detection in edge- or cloud-based analytics. After conducting an anomalous inspection, the supervisory control was fully automated. Subsequently, the real-time SCADA dashboard displayed the current number of screws with defected surfaces, after which the supervisor designed a production plan according to the real-time result to avoid delayed shipment. This study used MQTT and a broker server to develop a real-time cyber network for messaging and generating real-time information for the SCADA and notification applications. We observed that while MQTT

technologies played an important role in data connectivity, data consistency was important for communication in the industrial internet of things environment. Subsequently, this study validated the data consistency by stopping and restarting the broker server to simulate the network's disconnect and reconnect status. Figure 13 shows that the real-time defective product counter was correctly uploaded to the cloud while the MQTT client was reconnected to the MQTT broker. Investigations also revealed that the data during the disconnected status successfully re-messaged the cloud MQTT broker and could, thus, consistently communicate between the client sides. The last value of the defective product counter was 21, after which the network was disconnected. Further, a new value, 22, was uploaded when the network reconnected (Figure 13).

(**a**) MQTT broker—Disconnect (Interrupt)



(**b**) MQTT broker—Reconnect

Figure 13. Data consistency validation results.

Also, in this study, the proposed system provided a flexible format for the user to design an MQTT topic and anomaly detection message. The real-time yield could be calculated and displayed on the dashboard while SCADA receives the real-time message published from the anomaly event (Figure 14). Additionally, this study proposed APIs to automatically generate the anomaly report and send it to the supervisor by email. Interestingly, the third-party RESTful APIs cooperated with the MQTT protocol to provide an auto-notification using the chat tool, Bot. Then, the supervisor could set the alarm or notify the threshold. Although the value of the defective product was over the threshold, investigations showed that they could receive anomaly reports through emails, including an alarm message from the chat Bot (Figures 15 and 16). Moreover, this study proposed or used APIs based on the RESTful API framework. The RESTful APIs have different calling types, and this study used the "POST"-type to transmit information and create a resource. In addition, we proposed managing APIs through Swagger. In this study, the anomaly detection APIs enabled users to directly access the trained model via HTTP and provided

an efficient way for the cloud system to access the function. A system can send an API request through the HTTP media type and respond to information in the typical API design. Similarly, this study used the JSON format as its MQTT and API protocol response formats, which is widely used in web and mobile applications.

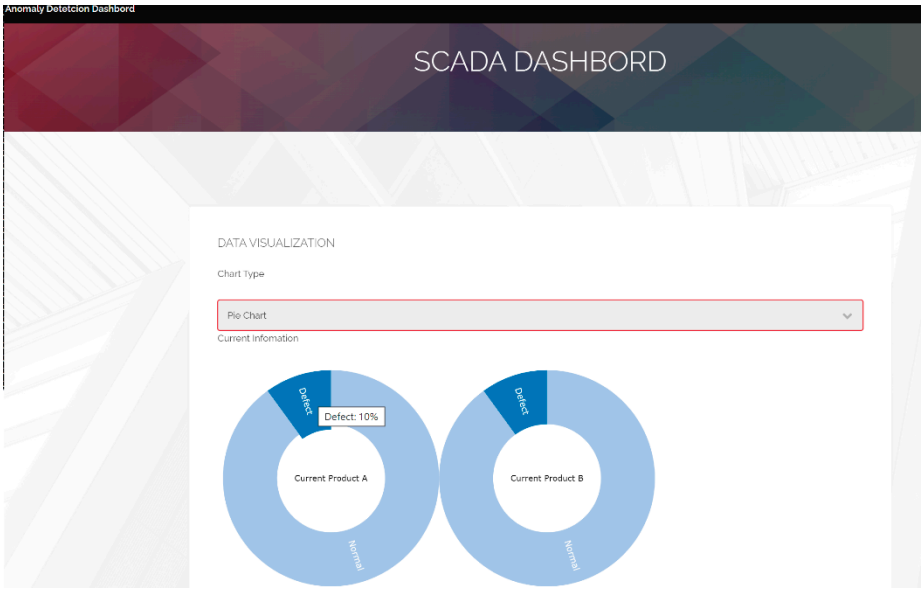


Figure 14. The real-time anomaly detection SCADA dashboard.

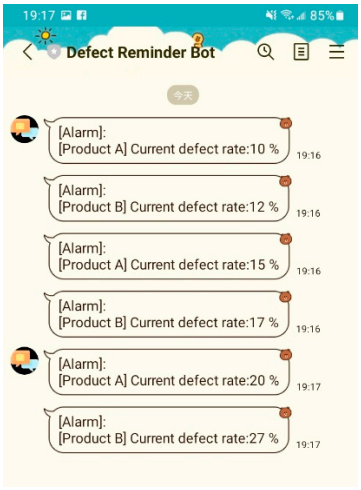


Figure 15. Real-time anomaly notification via Bot.

Figure 16. Real-time anomaly notification via email.

6. Discussion

The main finding of this study was the promising advantages of using deep learning approaches in detecting a defective product, surveying the relationship between training

size and deep learning approaches, enabling real-time communication, and transmitting the inspection information from the product to its users. Previously, Haselmann et al. [22] used deep learning approaches in anomaly surface inspection and showed that the trained model had good accuracy. Some studies have reported similar results, especially that the deep learning algorithm approach has outstanding analytical results [12,13,23–25]. From the first finding, we observed that VGG-16, Inception V3, and Xception better detected the normal and the four defective types. Especially, the cap head with forging cracks had better classification results using VGG, Inception V3, and Xception than other defective types. Such defective type has an obvious contour in the screw cap head, as shown in the cap head image. From the second finding, the training batch size did not differ significantly between Inception V3 and Xception. Accordingly, Filonenko and Kurnianggoro [47] highlighted that inception-based networks, such as Inception and Xception, performed highly during training of the image dataset with sufficient scenarios. Moreover, with dataset cases, the accuracy of using VGG had different results according to the training sample size [48]. However, VGG-related networks have limited depth, which may limit the result accuracy of the training data size.

Based on the third and fourth findings, this study proposed a new approach using MQTT and RESTful APIs to enable automatic anomaly detection. The MQTT connected real-time communication, data integration, and WS, after which the real-time SCADA dashboard displayed the anomaly event with graphical representations. The proposed and third-party APIs are subsequently used to dynamically send an anomaly report, notification, or alarm. Additionally, this study successfully used the proposed method for real-time anomaly detection. The proposed approach enhances the anomaly detection's efficacy and reduces the inspection time.

Industry 3.5 technologies develop cost-effective strategies using data, network, and existing manufacturing resources to transition to Industry 4.0. However, small- and medium-sized manufacturing institutions may face difficulties migrating to industrial internet of things-based anomaly detection. Therefore, this proposed research framework may assist emerging companies or small- and medium-sized manufacturing firms in improving defect detection using data analytics and cloud technologies. This study also shows a cost-effective approach based on the existing system instead of buying an advanced intelligent AOI system. Summarily, this study provides an effective and automatic tool for informing users about defective screw information, using the SCADA dashboard, notification Bot, and anomaly detection report. Therefore, by cloud techniques and the IoT communication protocol, the availability of real-time anomalous data enables earlier detection to avoid delayed shipment and reduces inspection time.

7. Conclusions

Defective human visual inspection is a key issue during screw inspection; it enhances the risk of human faults and increases inspection time. Therefore, this article presented a cost-effective automatic anomaly detection system using data analytics and cloud technologies. From the viewpoint of data analytics, we applied VGG-16, Inception V3, and Xception to analyze data from the AOI system and detect defective surfaces of screws. The Inception and Xception produced better inspection results than VGG-16, which agrees with Constantinescu's [49] and Koppikar's findings [50]. Specifically, while Constantinescu et al. [49] used medical images, Koppikar et al. [50] used the UCF-Crime dataset. Nevertheless, both results showed that Inception V3 had a higher accuracy than VGG-16 [49,50]. This study also investigated the relationship between training data sizes and models' performance. The proposed result using VGG-16 showed that the difference in performance existed between the training dataset sizes. Kandel's findings support the proposed result and showed that the smallest batch size achieved the lowest area under curve using a certain learning rate [51]. Another study also observed that the accuracy of using VGG had different results according to the training sample size [48]. Moreover, VGG-related networks have limited depth. Thus, the results' accuracy may limit the training data size. This study had a similar

finding. The experimental results showed that while using VGG-16, using a larger training dataset had better accuracy and precision than a smaller training dataset. Compared with other defection types, plating surface defect detection was the most difficult to detect. Moreover, deep learning has high accuracy and precision for an overall evaluation. Accordingly, the results support the robustness of convolutional-related networks with transfer learning to detect the defective surface of a screw with high accuracy, recall, and precision.

From the viewpoint of system implementation, this study proposed an anomaly detection system. We also developed APIs and third-party APIs to conduct real-time SCADA and alarm notifications, automatically generating a report. However, investigations revealed that while the API and Bot could automatically notify users of the correct anomaly information, real-time data communication could normally run only when the network is disconnected and reconnected. Moreover, since it was difficult to obtain a sufficient yield of anomaly data from the manufacturing site, the limited amount of defective screw cap head image datasets served as a limitation in this study. Nevertheless, this study demonstrated the entire workflow from data analytics to system design. Therefore, in future studies, an enlarged sample size of defective product images can help extensively explore more valuable information for anomaly detection.

Funding: This research was funded by Ministry of Science and Technology grant number MOST 110-2221-E-224-047 and no. MOST 111-2221-E-224-033-MY2.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Kamat, P.; Sugandhi, R. Anomaly detection for predictive maintenance in industry 4.0—A survey. *E3S Web Conf.* **2020**, *170*, 02007. [CrossRef]
2. IBM Cloud Education. The Basics of Business Automation. IBM Official Website. Available online: https://www.ibm.com/cloud/blog/basics-of-business-automation (accessed on 12 May 2021).
3. Stojanovic, L.; Dinic, M.; Stojanovic, N.; Stojadinovic, A. Big-data-driven anomaly detection in industry (4.0): An approach and a case study. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016; pp. 1647–1652.
4. Winters, P.; Adae, I.; Silipo, R. Anomaly Detection in Predictive Maintenance. In *Anomaly Detection with Time Series Analysis*; KNIME: Zurich, Switzerland, 2014; pp. 3–9.
5. Erdmann, M. Unsupervised Anomaly Detection in Sensor Data Used for Predictive Maintenance. Doctoral Dissertation, Ludwig-Maximilians-University, Munchen, Germany, 2018.
6. Minarini, F. Anomaly Detection Prototype for Log-Based Predictive Maintenance at INFN-CNAF. Master's Thesis, University of Bologna, Bologna, Italy, 2019.
7. Alaoui-Belghiti, A.; Chevallier, S.; Monacelli, E. Unsupervised anomaly detection using optimal transport for predictive maintenance. In Proceedings of the International Conference on Artificial Neural Networks, Munich, Germany, 17–19 September 2019; Springer: Cham, Switzerland, 2019; pp. 686–697.
8. Farbiz, F.; Miaolong, Y.; Yu, Z. A cognitive analytics based approach for machine health monitoring, anomaly detection, and predictive maintenance. In Proceedings of the 2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA), Kristiansand, Norway, 9–13 November 2020; pp. 1104–1109.
9. Carrasco, J.; López, D.; Aguilera-Martos, I.; García-Gil, D.; Markova, I.; García-Barzana, M.; Herrera, F. Anomaly detection in predictive maintenance: A new evaluation framework for temporal unsupervised anomaly detection algorithms. *Neurocomputing* **2021**, *462*, 440–452. [CrossRef]
10. Zhao, P.; Kurihara, M.; Tanaka, J.; Noda, T.; Chikuma, S.; Suzuki, T. Advanced correlation-based anomaly detection method for predictive maintenance. In Proceedings of the 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), Dallas, TX, USA, 19–21 June 2017; pp. 78–83.
11. Perini, L. Predictive Maintenance for Off-Road Vehicles Based on Hidden Markov Models and Autoencoders for Trend Anomaly Detection. Doctoral Dissertation, Politecnico di Torino University, Torino, Italy, 2019.
12. Serradilla, O.; Zugasti, E.; Ramirez de Okariz, J.; Rodriguez, J.; Zurutuza, U. Adaptable and explainable predictive maintenance: Semi-supervised deep learning for anomaly detection and diagnosis in press machine data. *Appl. Sci.* **2021**, *11*, 7376. [CrossRef]

13. Davari, N.; Veloso, B.; Ribeiro, R.P.; Pereira, P.M.; Gama, J. Predictive maintenance based on anomaly detection using deep learning for air production unit in the railway industry. In Proceedings of the 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA), Porto, Portugal, 6–9 October 2021; pp. 1–10.
14. De Benedetti, M.; Leonardi, F.; Messina, F.; Santoro, C.; Vasilakos, A. Anomaly detection and predictive maintenance for photovoltaic systems. *Neurocomputing* **2018**, *310*, 59–68. [CrossRef]
15. Bose, S.K.; Kar, B.; Roy, M.; Gopalakrishnan, P.K.; Basu, A. ADEPOS: Anomaly detection based power saving for predictive maintenance using edge computing. In Proceedings of the 24th Asia and South Pacific Design Automation Conference, Tokyo, Japan, 21–24 January 2019; pp. 597–602.
16. Coelho, D.; Costa, D.; Rocha, E.M.; Almeida, D.; Santos, J.P. Predictive maintenance on sensorized stamping presses by time series segmentation, anomaly detection, and classification algorithms. *Procedia Comput. Sci.* **2022**, *200*, 1184–1193. [CrossRef]
17. Decker, L.; Leite, D.; Giommi, L.; Bonacorsi, D. Real-time anomaly detection in data centers for log-based predictive maintenance using an evolving fuzzy-rule-based approach. In Proceedings of the 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Glasgow, UK, 19–24 July 2020; pp. 1–8.
18. Decker, L.; Leite, D.; Viola, F.; Bonacorsi, D. Comparison of evolving granular classifiers applied to anomaly detection for predictive maintenance in computing centers. In Proceedings of the 2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), Bari, Italy, 27–29 May 2020; pp. 1–8.
19. Mohammadi, B.; Fathy, M.; Sabokrou, M. Image/video deep anomaly detection: A survey. *arXiv* **2021**, arXiv:2103.01739.
20. Deecke, L.; Vandermeulen, R.; Ruff, L.; Mandt, S.; Kloft, M. Image anomaly detection with generative adversarial networks. In Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Dublin, Ireland, 10–14 September 2018; Springer: Cham, Switzerland; pp. 3–17.
21. Zhou, K.; Gao, S.; Cheng, J.; Gu, Z.; Fu, H.; Tu, Z.; Liu, J. Sparse-gan: Sparsity-constrained generative adversarial network for anomaly detection in retinal oct image. In Proceedings of the 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), Iowa City, IA, USA, 3–7 April 2020; pp. 1227–1231.
22. Haselmann, M.; Gruber, D.P.; Tabatabai, P. Anomaly detection using deep learning based image completion. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; pp. 1237–1242.
23. Xu, S.; Wu, H.; Bie, R. CXNet-m1: Anomaly detection on chest X-rays with image-based deep learning. *IEEE Access* **2018**, *7*, 4466–4477. [CrossRef]
24. Khan, A.S.; Ahmad, Z.; Abdullah, J.; Ahmad, F. A spectrogram image-based network anomaly detection system using deep convolutional neural network. *IEEE Access* **2021**, *9*, 87079–87093. [CrossRef]
25. Nguyen, B.; Feldman, A.; Bethapudi, S.; Jennings, A.; Willcocks, C.G. Unsupervised region-based anomaly detection in brain mri with adversarial image inpainting. In Proceedings of the 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI), Nice, France, 13–16 April 2021; pp. 1127–1131.
26. Cozzolino, D.; Verdoliva, L. Single-image splicing localization through autoencoder-based anomaly detection. In Proceedings of the 2016 IEEE International Workshop on Information Forensics and Security (WIFS), Abu Dhabi, United Arab Emirates, 4–7 December 2016; pp. 1–6.
27. Müller, R.; Ritz, F.; Illium, S.; Linnhoff-Popien, C. Acoustic anomaly detection for machine sounds based on image transfer learning. *arXiv* **2020**, arXiv:2006.03429.
28. Zhang, X.; Wen, G.; Dai, W. A tensor decomposition-based anomaly detection algorithm for hyperspectral image. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 5801–5820. [CrossRef]
29. Ayhan, B.; Dao, M.; Kwan, C.; Chen, H.M.; Bell, J.F.; Kidd, R. A novel utilization of image registration techniques to process mastcam images in mars rover with applications to image fusion, pixel clustering, and anomaly detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 4553–4564. [CrossRef]
30. Zhuang, L.; Gao, L.; Zhang, B.; Fu, X.; Bioucas-Dias, J.M. Hyperspectral image denoising and anomaly detection based on low-rank and sparse representations. *IEEE Trans. Geosci. Remote Sens.* **2020**, *60*, 1–17. [CrossRef]
31. Verdoja, F.; Grangetto, M. Graph Laplacian for image anomaly detection. *Mach. Vis. Appl.* **2020**, *31*, 1–16. [CrossRef]
32. Vojir, T.; Šipka, T.; Aljundi, R.; Chumerin, N.; Reino, D.O.; Matas, J. Road anomaly detection by partial image reconstruction with segmentation coupling. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 15651–15660.
33. Li, Y.; Liu, W.; Huang, Q. Traffic anomaly detection based on image descriptor in videos. *Multimed. Tools Appl.* **2016**, *75*, 2487–2505. [CrossRef]
34. Wang, L.; Zhang, D.; Guo, J.; Han, Y. Image anomaly detection using normal data only by latent space resampling. *Appl. Sci.* **2020**, *10*, 8660. [CrossRef]
35. Berg, A.; Ahlberg, J.; Felsberg, M. Unsupervised learning of anomaly detection from contaminated image data using simultaneous encoder training. *arXiv* **2019**, arXiv:1905.11034.
36. Chithirala, N.; Natasha, B.; Rubini, N.; Radhakrishnan, A. Weighted Mean Filter for removal of high density Salt and Pepper noise. In Proceedings of the 2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 22–23 January 2016.

37. Suhas, S.; Venugopal, C. MRI image preprocessing and noise removal technique using linear and nonlinear filters. In Proceedings of the 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), Mysuru, India, 15–16 December 2017.
38. Kusnik, D.; Smolka, B. On the robust technique of mixed Gaussian and impulsive noise reduction in color digital images. In Proceedings of the 2015 6th International Conference on Information, Intelligence, Systems and Applications (IISA), Corfu, Greece, 6–8 July 2015.
39. Cohen, N.; Hoshen, Y. Sub-image anomaly detection with deep pyramid correspondences. *arXiv* **2020**, arXiv:2005.02357.
40. Mishra, P.; Verk, R.; Fornasier, D.; Piciarelli, C.; Foresti, G.L. VT-ADL: A Vision Transformer Network for Image Anomaly Detection and Localization. In Proceedings of the 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE), Kyoto, Japan, 20–23 June 2021; pp. 1–6.
41. Mishra, P.; Piciarelli, C.; Foresti, G.L. A neural network for image anomaly detection with deep pyramidal representations and dynamic routing. *Int. J. Neural Syst.* **2020**, *30*, 2050060. [CrossRef] [PubMed]
42. Papageorgiou, C.P.; Oren, M.; Poggio, T. A general framework for object detection. In Proceedings of the Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271), Bombay, India, 7 January 1998; pp. 555–562.
43. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001.
44. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
45. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
46. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
47. Filonenko, A.; Kurnianggoro, L.; Jo, K.H. Comparative study of modern convolutional neural networks for smoke detection on image data. In Proceedings of the 2017 10th International Conference on Human System Interactions (HSI), Ulsan, Korea, 17–19 July 2017; pp. 64–68.
48. Antoniou, A.; Storkey, A.; Edwards, H. Data augmentation generative adversarial networks. *arXiv* **2017**, arXiv:1711.04340.
49. Constantinescu, E.C.; Udriștoiu, A.L.; Udriștoiu, Ș.C.; Iacob, A.V.; Gruionu, L.G.; Gruionu, G.; Săftoiu, A. Transfer learning with pre-trained deep convolutional neural networks for the automatic assessment of liver steatosis in ultrasound images. *Med. Ultrason.* **2021**, *23*, 135–139. [CrossRef]
50. Koppikar, U.; Sujatha, C.; Patil, P.; Mudenagudi, U. Real-world anomaly detection using deep learning. In Proceedings of the International Conference on Intelligent Computing and Communication, Bengaluru, India, 7–8 June 2019; Springer: Singapore, 2019; pp. 333–342.
51. Kandel, I.; Castelli, M. The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express* **2020**, *6*, 312–315. [CrossRef]