





A Generative Adversarial Network approach for automatic inspection in automotive assembly lines

Joceleide D. C. Mumbelli , Giovanni A. Guarneri , Yuri K. Lopes , Dalcimar Casanova ,
Marcelo Teixeira 

Abstract—In manufacturing systems, quality of inspection is a critical issue. This can be conducted by humans, or by employing *Computer Vision Systems* (CVS) which are trained upon representative datasets of images to detect classes of defects that may occur. The construction of such datasets strongly limits the use of CVS methods, as the variety of defects has combinatorial nature. Alternatively, instead of recognizing defects, a system can be trained to detect non-defective standards, becoming appropriate for some application profiles. In flexible automotive manufacturing, for example, parts are assembled within a reduced set of correct combinations, while the amount of possible incorrect assembling is enormous. In this paper, we show how a CVS can be extended with a *Deep Learning*-based approach that exploits a *Generative Adversarial Network* (GAN) to detect non-defective production, eliminating the need for constructing defect image datasets. The proposal is tested over the assembly line of Renault, in Brazil. Results show that our method returns better accuracy in inspection, compared with the current CVS solution, besides generalizing better to different components inspection without having to modify the method.

Note to Practitioners—In manufacturing, the variety of possible defects limits the training of computational algorithms to automatically detect them. In this paper we show how industry practitioners can conduct training only upon non-defective images, thus treating any other case as a defect and dispensing the construction of specific image datasets to represent defects. The approach is tested over a real automotive assembly line and the results are shown to be sound from an applied point of view.

Index Terms—Automatic inspection; Deep learning; Generative Adversarial Networks; Automotive manufacturing.

I. INTRODUCTION

As *Manufacturing Systems* (MSs) [1], [2], [3] progress towards digitalization, they require more flexibility to process multiple types of products over the same plant. An example emerges from the automotive domain, where different models of vehicles are manufactured, each one using its own set of components. Environments like this create new challenges and requirements for classic control and automation practices, such as the need for visual inspections to be integrated at many steps of manufacturing and over many types of assembly sets [4]. Inspections aim to guarantee that manufactured products are zero-defect and meet certain security and quality requirements.

In practice, several types of defects can be inspected, e.g., component placement, position, soldering and type consistency. The inspection task is in general conducted by a human

agent, so those faulty components are sent back to rework, while the others progress through the process. As this is error-prone and unproductive, there are many attempts to replace human inspection with automatic decisions taken by CVSs [5]. A CVS includes a capture camera and embedded computing that allow inferring visual details into an image.

Unfortunately, the accuracy of a CVS still fails to match visual inspections for most applications, despite the efforts in the literature [6]. The main barrier is that the result of a CVS inspection varies according to the problem nature, suffering from environmental influences, such as noise, lighting, and angle of image capturing [7]. In the classic view of CVS-based inspections, each type of defect requires a specialized computational method to be detected, implying substantial efforts in research, engineering, and implementation.

Novelties in *Deep Learning* (DL) have improved CVS and made them more applicable and accurate for industry [8], [9], [10], with promising impacts, e.g., on the automotive domain [11], [12], [13]. However, classical DL approaches (i.e., based on supervised training) still rely on the dataset quality and its representativeness. They are usually trained over datasets that include a significant number of images, representing each type/class of defect that may occur in manufacturing. When the diversity of defects is wide and combinatorial, the construction of such a dataset imposes a strong limitation for the use of most DL CVS in a supervised way, making them weakly generalizable to industry.

Alternatively, instead of training the DL method in a supervised way, with a reasonable set of image samples for each possible defect, a system could be trained to detect only non-defective standards, so that any situation otherwise is considered a defect. This becomes appropriate for applications where the number of defects is (much) greater than the non-defective cases. In flexible automotive manufacturing, for example, parts are presumably assembled within a reduced set of correct combinations, while the number of possible incorrect assembling is enormous. Thus, training a CVS to recognize a variety of incorrect cases may be an exhaustive, sometimes unfeasible, task that delays the production flow and allows errors to survive post-production.

Computationally, this scenario can be seen as an anomaly detection task, and the *Generative Adversarial Networks* (GANs) are nowadays the most promising unsupervised learning method to address it [14]. GANs dispense the construction of image datasets to represent defects, as their training requires only correct images, which are in general easier to find on factory floors. Therefore, GANs are expected to generalize

Joceleide D. C. Mumbelli, Giovanni A. Guarneri, Dalcimar Casanova, and Marcelo Teixeira are with the Federal University of Technology – Paraná. E-mails: joceleide.dcm@gmail.com, {giovanni,dalcimar,mtx}@utfpr.edu.br. Yuri K. Lopes is with the Department of Computer Science of the Santa Catarina State University. E-mail: yuri.lopes@udesc.br.

better for the automatic inspection of several types of defects, evidencing practical appeal for different areas, such as agriculture [15], optimization problems [16], general manufacturing [17], [18], machining [19], and surface inspection [20]. Yet, applications for visual inspection in the automotive domain are still incipient.

This paper extends a production line verification CVS currently used by Renault, in Brazil. This system faces limitations related to the number of points that need to be verified, the speed required for image processing, and the verification angle. Those difficulties, in conjunction with the classical CVS embedded in the capturing system, reduce the quality of inspection, affecting the advantages of the CVS. The approach proposed here is a DL-based inspection system that exploits a GAN to improve the automatic inspection tests along the flexible assembly line of Renault. By dismissing defective image examples in the training phase, our approach recognizes more defective types with a single model, evidencing better performance in comparison with the current CVS system. Tests suggest an increase of 2,13% in the mean accuracy for detecting defects that, comparably, are also detected by the current CVS, and identification of countless other defects that currently have not been detected.

Structurally, the manuscript is organized as follows: a literature review is presented in Section II; Section III discusses the related concepts; Section IV introduces the main results; while conclusions and perspectives are discussed in Section V.

II. RELATED WORK

In industry, human inspections have been gradually replaced by robots that use embedded CVSs to perceive the environment, collect and process information, and take decisions accordingly. A sketch of this relation is shown in Fig. 1.

A CVS takes pictures of a certain region of interest in objects that include points to be verified. Images may be processed with or without automatic learning. Feature extraction via contour detection [21], for example, checks the edges of an image to determine the classes within which it belongs. Its main gap is to depend on human effort to complete each test.

Recent advances in *Machine Learning* (ML) and DL have shown the potential to eliminate human dependency, to some extent. This has been demonstrated by industrial inspections such as in aircraft fuselage [22], cement cracks [23], conformances in automotive assembling [13], and metallurgical checks [24], [25]. Inspections like these are simple to be done when defects have minor diversity, i.e., when they belong to a limited set of types, facilitating them to be mapped by images and learned after training by employing supervised learning. In some applications, e.g., automotive manufacturing, defects have high diversity, challenging the construction of a dataset that reasonably reflects this.

One option is to complement the dataset with images generated artificially to represent a defect. DL itself can be used to generate images that improve the defects diversity. *Recurrent Neural Networks* (RNNs) [26], *Restricted Boltzmann Machines* (RBMs) [27], [28], and *Variational Autoencoders* (VAEs) [29], are examples of such techniques. They work well

for low resolution images, but may be unsuitable when defects are sensitive and require more precise images.

In this case, an alternative is to assume that, instead of recording every possible manufacturing defect, one can record and learn/model only over correct cases, so that any other possible variation is a defect. Computationally this defect is called an anomaly/outlier on the dataset. More specifically, when training data is not polluted by outliers, and we are interested in detecting whether a new observation is an outlier, we refer to this task as novelty detection. Novelty detection is a semi-supervised task, as it is known that the training step includes only normal (i.e. non-defective) samples. A DL method that implements this idea of novelty detection as a semi-supervised task is the GANs. This method simultaneously trains a generator, to produce fake images, and a discriminator, to distinguish between real and fake images. Images generated by GANs have in general good quality and are qualified to enrich image datasets for classification tasks subject to a high diversity of defects.

Variations of GANs include: *Deep Convolutional Generative Adversarial Networks* (DCGANs), which uses convolution to ensure stability and convergence during training [30]; *Wasserstein Generative Adversarial Networks* (WGANs) [31], which uses the Wasserstein distance to effectively solve the problem of gradient disappearance during training; *Cycle Generative Adversarial Networks* (CycleGANs) [32], which allows images from two domains to be generated without paired images; among other variations [14].

In industry, GANs have been tested in steel defect classification [33], palm print recognition [34], people identification [35], vehicle license plate recognition [36], medical image synthesis [37], [38], texturing industry [39], etc. However, applications in the automotive domain are still emerging and are limited to detecting longitudinal errors of sensors in autonomous vehicles [40], or to generate sensor errors as an attempt to test safety in advanced driver assistance systems [41]. No results on GANs detection over assembly lines have been found so far. Thus, this paper exploits GANs for: (i) increasing image datasets that map critical manufacturing defects in vehicles production; and (ii) learning how to differentiate defective and non-defective images without having images for all possible defects that may occur.

III. BACKGROUND

Parallel and multidisciplinary advances in engineering have allowed the industry to produce flexibly, on-demand, quickly, and under reduced costs [42]. Adapting the factory floor to these requirements is non-trivial and presupposes the integration of multiple and complex technologies. Among them, there are cognitive approaches, which aim to enrich automatic production systems with abilities hitherto only perceived by humans, such as context recognition, touch, hearing, and vision [43], [44], where the latter is of interest here.

An application domain that benefits from visual inspections is the automotive manufacturing, where the quality of the manufactured product is associated with aesthetics, safety, and human life risks [45], besides being linked with the high added

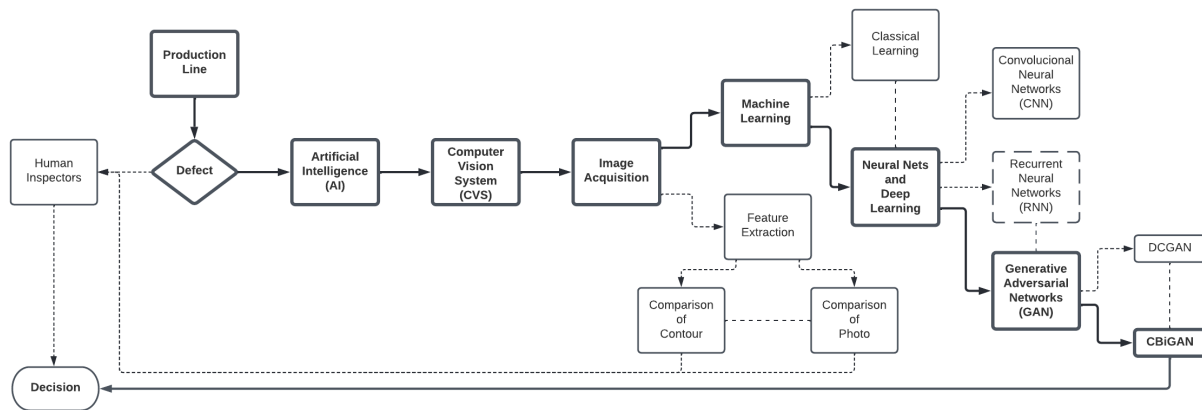


Figure 1. Overview on the detection of manufacturing defects based on human-centered and automatic approaches. In bold, we identify the path pursued in this research, while the dashed lines highlight options in the literature that are not exploited here.

value of the product. Thus, any anomaly in manufacturing implies restarting a long, complex, and expensive chain of rework, sometimes after the product is already in use.

Ensuring the quality of a vehicle depends directly on the quality of the parts, their compatibility of type, and how they are assembled. Defects can emerge physically (e.g., the morphology of a part), or logically (e.g., the use of a non-defective part in the wrong vehicle). Detecting and preventing possible defects is decisive [46], and this can be done automatically or not. Usually, the diverse mix of features makes visual inspection difficult to be fully automated, and it remains human-centered. As such, it is monotonous, not rarely harmful to the human health [47], besides being imprecise.

A way to reduce human dependence is by adopting CVSs [48]. A CVS includes the orderly steps of acquisition, pre-processing, feature extraction, segmentation, and classification, with the purpose of reproducing human reasoning, to some extent. The first two steps extract and process images obtained from the environment, making them suitable for identifying features, after mathematical treatment, and isolating segments containing such features. Then, classifier algorithms identify patterns in those segments, usually by evaluating similarities between regions of interest and predefined templates, comparing the values of pixels within an acceptance threshold.

In general, commercial CVSs do not generalize well for real problems in manufacturing. Although image acquisition systems include modern equipment, they require manual setup of templates for classification, both for pre-processing methods and for lighting and distance adjustments. These are strong limitations for a CVS, as experts are needed anyway, and their profile may vary from one application to another. Also the manual definition of thresholds for classification requires experienced knowledge [49], [50], [51].

The use of DL-based techniques has shown potential to reduce these limitations and automate some of the expert-dependent tasks [52], [53]. Yet, this alternative still faces a crucial problem: the number of possibilities in which a defect can be created and interpreted. When a manufacturing system requires this type of multiple-objects flexibly-assembled inspection, the use of CVSs becomes practically unfeasible due

to their weak generalization capabilities to recognize multiple and unexpected defects [54], [55].

The proposal in this article emerges from this reasoning and hypothesizes that those limitations can be overcome, to some extent, by applying modern learning systems based on *Generative Adversarial Networks* (GANs) [39].

A. Generative Adversarial Networks

A *Generative Adversarial Network* (GAN) is part of a deep neural network architecture that consists in training two models (players) to take decision by competing against each other. One player, called *generator* (G), is a neural network that generates new (fake) data instances, while the other, called *discriminator* (D), evaluates their authenticity. By analogy, G acts like an intruder who tries to create and spread false leads to make its identification harder, while D is a detective who tries to filter the leads that make sense.

In order to make the false leads, G applies random noises to generate new data as real as possible to those in a training dataset. This aims to fool D , which then has to decide whether or not each data instance belongs to the training dataset, or it has been created by G . The Fig. 2 illustrates a GAN scheme.

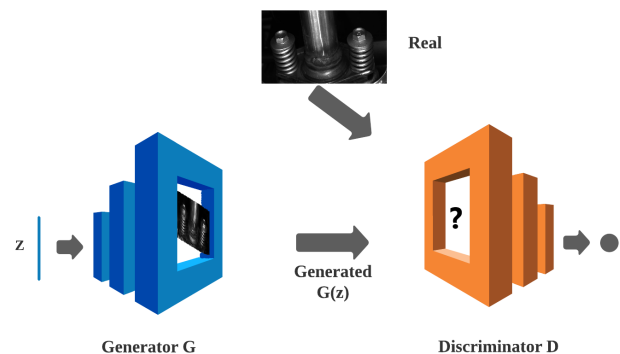


Figure 2. General overview of a GAN architecture.

The game challenge can then be summarized as follows: given a dataset with training data samples, a generator of fake samples G , and a discriminator of instances D , consider that D is trained to maximize the chances of assigning a correct label for samples coming indistinctly from the dataset or G ; inversely, G is trained to minimize the hits of D .

Remark that the rules for G and D to compete can be intuitively seen as a two-player *minimax* game, inherited from the *Games Theory* [56]. It assumes that there is always a rational solution to a well-defined conflict between two individuals whose interests are opposite. By following this reasoning, G and D can be seen as two opposite players that take optimized decisions through a value function $V(G, D)$ [57], as follows:

$$\begin{aligned} \min_G \max_D V(D, G) &= \mathbb{E}_{\mathbf{x} \sim p_d(\mathbf{x})} [\log(D(\mathbf{x}))] \\ &+ \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1-D(G(\mathbf{z})))] \end{aligned}$$

where: \mathbf{x} is the real data sample, with distribution $p_d(\mathbf{x})$, that approximates the expected value $\mathbb{E}_{\mathbf{x}}$; and $p_z(\mathbf{z})$ is a latent variable associated with an input noise vector with random uniform distribution $z \sim U(-1, 1)$, that approximates the expected value $\mathbb{E}_{\mathbf{z}}$. Then, $G(\mathbf{z})$ generates data from the input noise $p_z(\mathbf{z})$, while $D(\mathbf{x}) \in [0, 1]$ discerns how likely its input is to be true, or inversely fake.

The GAN training converges when *Nash-equilibrium* is reached in the minimax (zero-sum) game [58], i.e., when the actions of one player do not change depending on the actions of the opponent. Here, this means that the GAN generator $G(\mathbf{z})$ produces realistic images and the discriminator $D(\mathbf{x})$ outputs random predictions (probabilities close to 0.5) [59].

However, GANs are typically trained using gradient descent techniques that are designed to find a minimum for a cost function, instead of finding the Nash-Equilibrium, as it may lead the search not to converge [60]. In other words, achieving Nash-equilibrium often proves difficult due to training instability [61], and approaches such as Wasserstein GANs arise.

1) *WGAN*: The Wasserstein GANs (WGANs) are alternatives for training conventional GANs that tend to improve the learning stability. This also prevents problems like the mode collapse, and provides meaningful learning curves that are useful for debugging and hyperparameter searches [31].

Here, $D(\mathbf{h}) \in \mathbb{R}$ is an auxiliary scalar function, which is used to calculate the Wasserstein distance that replaces $D(\mathbf{x})$, in the minimax game, so that:

$$\begin{aligned} \min_G \max_D V(D, G) &= \mathbb{E}_{\mathbf{h} \sim p_d(\mathbf{h})} [D(\mathbf{h})] \\ &- \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [D(G(\mathbf{z}))]. \end{aligned}$$

In this way, the D function moves from a classifier to a critic, producing an authenticity score. This tends to assign high scores to real samples, and low scores to simulated samples. More details on WGANs can be found in [62], [63].

In addition to the WGANs, other variations of GANs have emerged to optimize the original theory, such as the BiGANs and the CBiGANs, explained in the following.

2) *BiGAN*: These are GANs improved to optimize the latent space, exposing this space to the discriminator along with the images generated from it [64]. A codifier module, denoted $E(\mathbf{h})$, is introduced and trained in conjunction with G , in a way to map the real samples to their respective latent spaces. The discriminator $D(\mathbf{h}, \mathbf{z}) \in [0, 1]$ is trained to discern whether the couple (\mathbf{h}, \mathbf{z}) comes from a real or generated image. The minimax problem for BiGANs can then be introduced as follows:

$$\begin{aligned} \min_{G, E} \max_D V(D, E, G) &= \mathbb{E}_{\mathbf{h} \sim p_d(\mathbf{h})} [\log D((\mathbf{h}), E(\mathbf{h}))] \\ &+ \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1-D(G(\mathbf{z})), \mathbf{z})]. \end{aligned}$$

Thus, fooling D causes G and E to minimize the difference between the ordered pairs $(G(\mathbf{z}), \mathbf{z})$ and $(\mathbf{h}, E(\mathbf{h}))$.

3) *CBiGAN*: The *Consistency BiGAN* (CBiGAN) [39] aims to combine features from WGAN and BiGAN. This approach improves the modeling of the latent space by exposing it to the discriminator (BiGAN) and also produces authenticity scores (not classification) from WGAN. In this way the CBiGAN tackles anomaly detection as a one-class classification problem, assuming for the training only non-anomalous samples. Given a test sample, the CBiGAN labels it as normal/nonanomalous/defect-free (considering a negative class) or anomalous (otherwise).

CBiGAN can be seen as a GAN that captures the distribution of latent space $Z = \mathbb{R}^n$. The generative model is a BiGAN and the loss function is modeled as Wasserstein distance. Then, the new minimax problem can be stated as:

$$\begin{aligned} \min_{G, E} \max_D V(G, E, D) &= \mathbb{E}_{\mathbf{h} \sim p_d(\mathbf{h})} [D(\mathbf{h}, E(\mathbf{h}))] \\ &- \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [D(G(\mathbf{z}), \mathbf{z})], \end{aligned}$$

where:

- $G : Z \rightarrow h$ is the generator that produces false images of latent variable associated with an input noise vector.
- $E : h \rightarrow Z$ is an encoder that map real samples to the corresponding latent space and trained together with G .
- $D : h \times Z \rightarrow \mathbb{R}$ is the discriminator/critic that produces authenticity scores to each samples using Wasserstein distance.

This model is of particular interest in this work, since our training is performed only on non-anomalous/defect-free samples. They are easily acquired in the industrial environment, while samples with defects are more rare and variable.

IV. PROPOSED APPLICATION

This section introduces our approach to detecting defects that remain after checking in the experimental CVS at the Renault manufacturing. A condensed version of the real process is shown in Fig. 3. This environment is called *image island*. It was created by Renault to carry out the final inspection of the underside of vehicles, just before they are attached to their painted bodywork. It consists of a CVS that includes robotic arms, two high resolution cameras, and a computing system. Assume that the production line aims to deliver 60 cars per hour, and the cycle time is 54 seconds. Therefore, at each

point on the line, employees or equipment have 54 seconds to complete their work, while the remaining 6 seconds are spent for a vehicle to pass from one checkpoint to another.

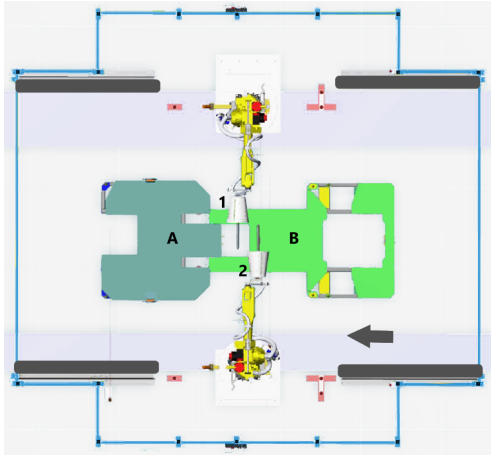


Figure 3. Representation of an image island at Renault, with a car positioned under the two cameras that capture images from points of interest.

Each car that passes through the island is seen as two parts, A and B, that are positioned in front of two cameras, 1 and 2. They both collect images of certain regions of interest, first from A and then from B. Remark that cars belong to different models, so that the image processing consists in automatic inspection of many components of several car models.

For the automated checking, assume that A and B are conceptually split in 4 sections. Each section has a number of points to be verified, which varies from one section to another. In total, at most 11 points are checked through the 4 sections, which may vary according to the model. The line does not stop during the verification, so that a sensor detects the presence of a new vehicle and starts the verification process automatically.

When a car enters the inspection cell, a PLC controller receives an “ok” signal and informs the robot controller of the coordinates of the point to be checked. When the arm reaches that correct position, its controller reports this to the PLC, which finally authorizes the camera controller to collect an image of the item to be verified. The lighting ring switches on and off for each image capturing to achieve the best possible shooting condition. After the image is captured, the native verification algorithm is triggered for that particular point and returns a Boolean value associated with the anomalous/defect or non-anomalous/defect-free assembled component. The test result is informed to the main PLC, which interfaces it through a human-computer interface. The same procedure repeats for the other verification points.

The main problem with the current system is that it uses classic CVS methods that are limited in training due to the lack of images representing defects. As a result, it returns a substantial rate of false negatives (FN) (i.e., defect-free images detected as defective), and some false positives (FP) (i.e., images for which it has not been possible to be sure about correctness, but the equipment detected as correct).

Implications of mistaken inspections can be severe. All cars detected as defective need to leave the line and go through a

thorough manual inspection. Cars detected as defective that are really defective, would have to be reworked anyway, so they are an inevitable problem; cars that are not really defective but have been classified as such (i.e. FN ones – these are more rare events in this case study, but still) are also a problem because they are removed from the line unnecessarily, consuming time and resources. However, the main problem comes from FP classification, i.e., with defective cars that are detected as normal by the CVS. This type of verification error is quite unacceptable in practice, as it takes effort and time, as well as increases production costs, besides jeopardizing the final integrity of the product and human lives. The next section quantifies the error rate of the existing CVS.

A. Problem quantification

The production line was observed for a time window during which 10 cars were inspected. Among these cars, 7 were of the same model and included 11 verification points in their underbase. The current automatic inspection system resulted in 77 verified points, from which 62 were evaluated as non-defective, and 15 as defective.

After manual inspection, it was confirmed that the 15 points initially classified as defective were actually correctly assembled, i.e., FN points. This outcome corresponds to an error rate of 19.5%, which is impractical and makes the equipment unfeasible to be used, as implied in 15 completely unnecessary rechecks.

Tables I and II show respectively the confusion matrix and the accuracy for the checked points. Defect-free inspections are represented as positive classifications (P), while a defect is expressed as negative (N), which are further associated with a Boolean value to form the following assertions:

- TP = Positive evaluated as positive (true defect-free).
- TN = Negative evaluated as negative (true defect).
- FP = Negative evaluated as positive (false defect-free).
- FN = Positive evaluated as negative (false defect).

Component	TP	TN	FN	FP
C20	102	0	1	1
C60	105	0	1	0
C100	270	0	0	0
C147	256	1	5	3
C231	96	0	28	0
C259	238	0	34	2
C267	232	10	32	0
C287	101	5	8	17
C329	40	0	0	1
C369	247	0	0	0
C492	254	4	14	0

Table I

CONFUSION MATRIX OF THE CURRENT CVS MODEL. IT CONSIDERS EACH COMPONENT OF A GIVEN CAR MODEL THAT WAS CHECKED OVER A CERTAIN TIME WINDOW.

Table I reveals a large amount of classification errors (FN and FP), reaching 34 FN cases for component number C259, and 17 FP cases for component C287. These were the most extreme cases of errors observed on the dataset.

For testing, we selected the sets of components that included a reliable amount of TN images, which were the components C147 and C287. Component C369 was also selected for

testing, as it refers to a part similar to others available outside the factory environment, which facilitates us to better compare our results. Table II shows the accuracy of the tested items.

Component	Accuracy
C147	96.88
C287	76.41
C369	100.00

Table II

ACCURACY OF THE CURRENT CVS WHEN CHECKING THE COMPONENTS OF A GIVEN CAR MODEL OVER A CERTAIN TIME WINDOW.

Components C147 and C287 have low accuracy when it comes to industrial standards. Although C369 has acceptable accuracy, it is a component that has not been exposed to any possibility of failure, and the result is unclear when exposed to defects. Therefore, the practical use of the equipment is compromised by the lack of classification confidence.

B. Image dataset and experimental setup

Our experiments exploit two distinct setups: one including images collected directly from the assembly line; and another, called controlled setup, constructed with images of real workpieces, that were collected outside the factory floor. For this second setup, workpieces have been physically modified to simulate a number of possible anomalies to be detected by the proposed inspection system. The two setups are described as follows.

1) *DATASET 1 - real images*: The first dataset (DS_1) is composed of images collected from the real production environment. With the factory in full production, the images were captured from the island by cameras attached to robotic arms. This dataset includes 3 distinct partitions, each one referring to a distinct type of component, each type is further divided into train and test data, as follows:

- Component C147: Steering Case Pin Rubber.
- Component C287: Gearbox Sensor.
- Component C369: Exhaust Pipe Screw.

Samples of each type of component are shown in Figures 4a, 4b, and 4c. The configuration of training and testing datasets is presented in Table III. All images in the training set are TP images, i.e., without defects, while the testing dataset includes indistinctly TP and TN images.

Partition	Profile	C147	C287	C369
Train dataset	Defect-free Real	165	48	176
	Defect-free DA	1650	1344	1760
Test dataset	Defect Manual	65	63	46
	Total	100	83	71
	Total dataset	265	131	247

Table III

DS_1 PARTITION SETTINGS FOR TRAINING AND TESTING.

As the capturing camera is positioned considerably distant from the components (about 40 cm) the resulting images include fragments that do not belong to the zone under analysis (e.g., background and borders). Thus, a preliminary region of interest (ROI) was defined to allow focusing on particular objects under consideration. The result of this preprocessing phase can be seen in Figures 4d, 4e, and 4f.

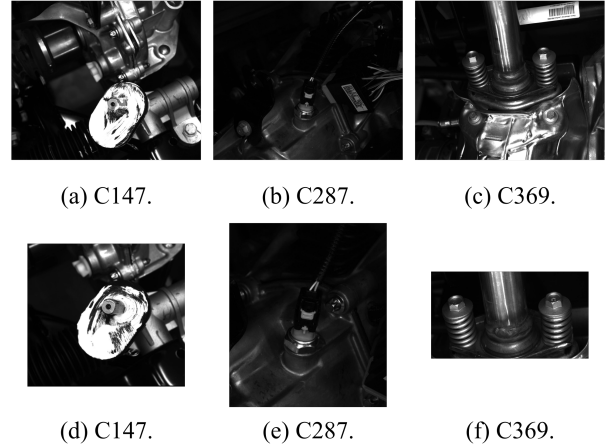


Figure 4. Original images (a), (b), and (c), for the components C147, C287, and C369, captured directly from the assembly line, and respective ROIs version (d), (e), and (f).

For the training dataset, a data augmentation (DA) [55] technique was applied to increase the number of training images. We applied variations in rotation, width and height shifts, zoom scale, and filling modes. The final number of images of each component after DA can be seen in Table IV.

Partition	Profile	C147	C287	C369
Train	Defect-free Real	165	48	176
	Defect-free DA	1650	1344	1760
Total dataset		1815	1392	1936

Table IV

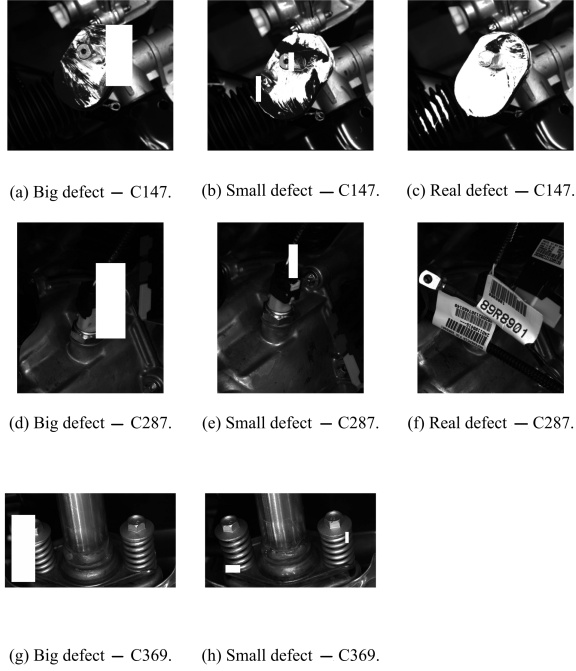
TRAINING DS_1 AFTER DA.

Remark that training our method requires only non-defective images to be available, which is a huge advantage in comparison with other approaches in the literature. In fact, TP images are usually more frequent and easier to obtain, in contrast with DL-based approaches that require collecting a reasonably large set of TN images, including all possible types of defects.

After the training phase, it is necessary to evaluate the accuracy and stability of the trained model when subject to an independent dataset that may include both TP and TN images. As images are captured in a real environment, real defects are less frequent events. Therefore, to fairly test the model face to more complex classification tasks, we conducted a manual DA over the test images so that more instances of TN parts are produced. The imposed defects are similar to real defects, observed in real TN images, extended with new features of possible defects. The defects have been separated into small and large size defects, as listed in Table V. Examples of TN images in the testing dataset with real defects (when they exist) and manually added defects, can be seen in Fig. 5.

2) *DATASET 2 - controlled images*: The second dataset (DS_2) was composed of images of parts collected by a camera apart from the island. This controlled image generation allows to represent a large number of possible defects and test whether or not the model is capable of recognizing them.

Table VI summarizes the number of images in DS_2 . As DS_2 consists of only 1 part, and this part is the same as C369 of DS_1 , the total of images contained in DS_2 is equal to DS_1

Figure 5. Production line DS₁ parts defects

Partition	Class	Profile	C147	C287	C369
Train	Defect-free		1815	1392	1936
	Defect-free		35	20	25
Test	Defect	Real	04	22	00
		Manual small	32	19	23
		Manual huge	29	22	23
		Total	65	63	46
Total			1925	1475	2007

Table V

NUMBER OF IMAGES IN TRAINING AND TEST DS₁ AFTER DA.

for this part. Analogous to DS₁, DA was also applied to the training images in DS₂ to make them comparable.

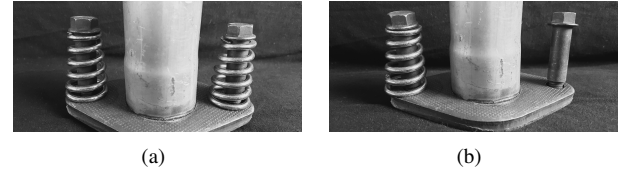
Partition	Profile	CR
Training dataset	Defect-free	1936
Test dataset	Defect-free	25
	Defect	46
	Total	71
Total dataset		2007

Table VI

DS₂ PARTITION SETTINGS FOR TRAINING AND TESTING DATASETS.

From the controlled dataset DS₂, we can simulate a wide diversity of defects. The test dataset did not undergo any computational manipulation, so the defects present in this DS₂ partition are defects that would actually pass through the assembly line, i.e., the photographed part was defective, with no need to create defects via software. Examples of images used for training and testing are shown in Fig. 6.

3) *Experimental analysis*: To assess the performance of both experiments, we used the 2-way holdout (training/test split) method, with confidence interval via normal approximation. We feed the training data to the method to learn from, and then estimate the performance over unseen data, i.e., the test was entirely conducted over images not used for training. No cross-validation or hyperparameter adjustment schemes were

Figure 6. Images with (a) and without (b) defects, from the dataset DS₂.

applied to perform the model selection since this is already known in the literature [39], and also due to the involved computational cost.

As our primary evaluation, we measured the accuracy of correct classifications between TP and TN assembly parts. Accuracy is a simple metric that represents the number of correct model predictions. It can be defined as the number of correctly classified test cases, divided by the total number of test cases. We then compare these results with those returned by the CVS installed on the real manufacturing line. Figures 7, 8, and 9 show examples of the images created by CBiGAN. The network was trained with images of parts from the industrial environment, i.e., the generated images are very similar to factory floor images.

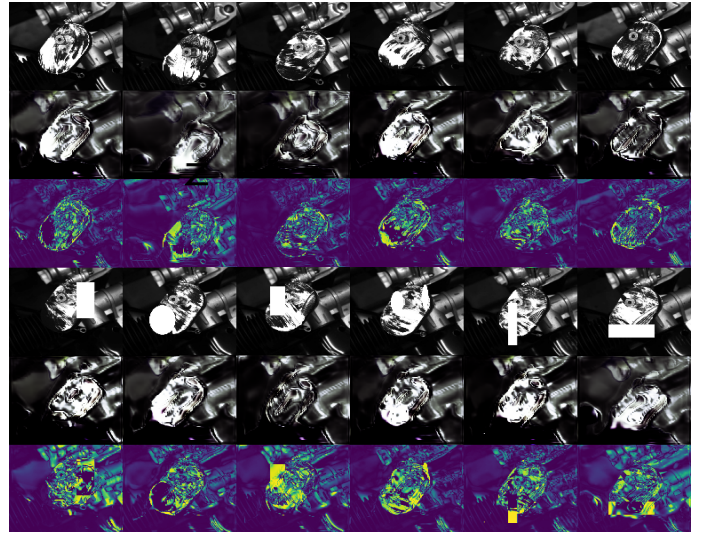


Figure 7. Image produced by CBiGAN for part C147, which shows in its lines the training and test images, in addition to the images generated by CBiGAN and the difference between the input and generated images.

Figures 7, 8, and 9 are seen as two partitions split horizontally. Each partition is formed by three lines: the line at the top shows images of parts belonging to the test dataset (one different part in each column); the line in the middle has images of these parts recreated by the network; and the third line shows the difference between them. The same follows for the second partition, at the bottom.

Table VII shows the accuracy of our tests, which can be compared with Table II to reveal their differences with respect to the real equipment performance to classify each type of part.

An accuracy increase from 96.88% to 98.4% was observed to classify the component C147. The improvement seems to be minor for this case, which is explained by the low amount of images of defects of this component. Differently,

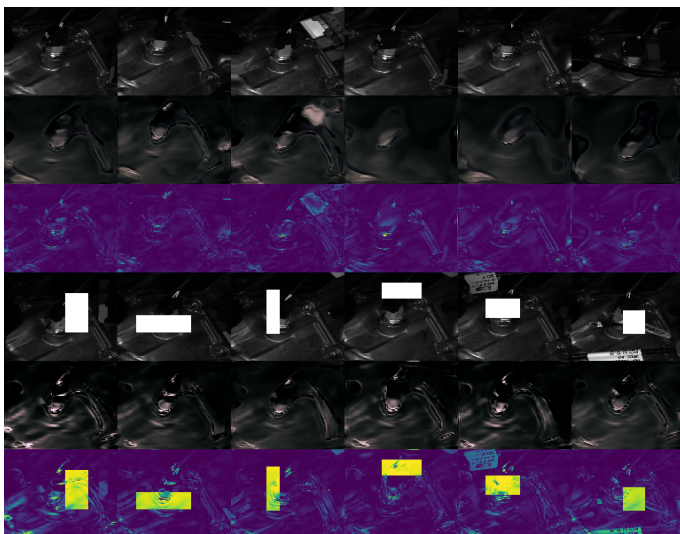


Figure 8. Image produced by CBiGAN for part C287, which shows in its lines the training and test images, in addition to the images generated by CBiGAN and the difference between the input and generated images.

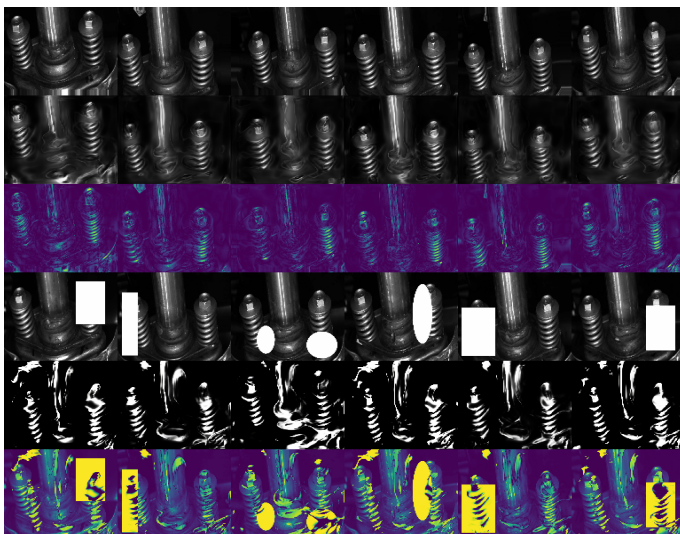


Figure 9. Image produced by CBiGAN for part C369, which shows in its lines the training and test images, in addition to the images generated by CBiGAN and the difference between the input and generated images.

Component	Accuracy %
C147	98.4
C287	85.9
C369	100.0

Table VII

ACCURACY ACHIEVED IN THE TEST WITH CBiGAN FOR EACH SET OF COMPONENTS.

when classifying components C287, the accuracy increases from 76.41% to 85.9%, i.e., more than 10%. This could be further improved by a more robust training step, including more images of real defects to feed the model. For component C369 the obtained accuracy was on the order of 100%, both for DS_1 and DS_2 .

Upon comparison, we conclude that our model was capable of improving, or at least maintaining, the accuracy of the real vision system for all the image profiles we tested. We

highlight, however, that our model is further advantageous in the sense that it also identifies abnormal components and is able to identify all possible defective components, as it considers defective anything different from a good component used for training. We claim this is a significant improvement from a practical point of view.

We now test how our model performs when processing DS_2 . This is expected to be harder, as DS_2 was constructed with the purpose of simulating a broader diversity of defects. Fig. 10 shows examples of images generated by the CBiGAN using DS_2 .

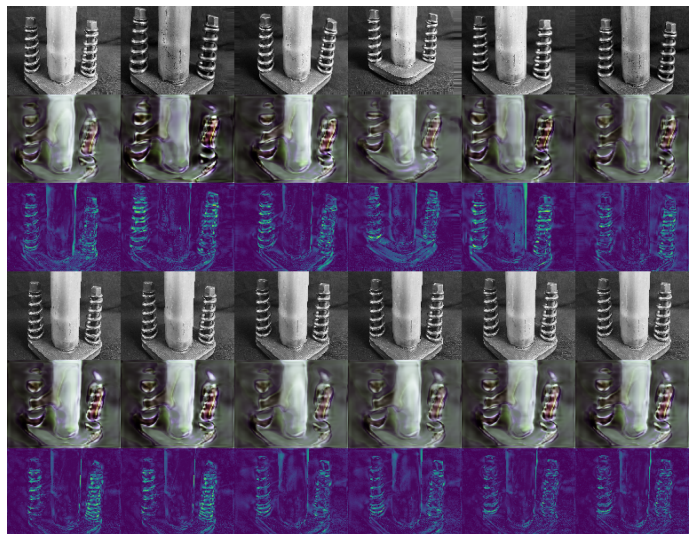


Figure 10. In the picture we can see some real images, some produced by CBiGAN, and the different between them for the DS_2 set of components.

As DS_2 is composed of the same C369 component as DS_1 , we were able to compare how CBiGAN behaves with controlled images. In our tests, the result was considered excellent, keeping accuracy of 100%. Although more real tests on the factory floor are needed, it is a good indication that the method can bring real benefits to the identification of defects.

V. CONCLUSIONS AND PERSPECTIVES

In industry, ensuring the quality of products is increasingly important and challenging, especially in flexible manufacturing plants. In the automotive industry, for example, a single production line is usually responsible for manufacturing several models of cars, each one with its own set of multiple components to be assembled. When a car shows after-sales defects, it denigrates the brand image and increases costs due to *a posteriori* repairs. In this context, applying only manual inspection as the quality control mechanism may not be the most effective way to ensure quality, and require support from automated visual inspections.

This type of solution, however, has two strong limitations: (1) it requires a specialized solution for each type of defect to be detected, in each product type; and (2) it needs input data (e.g., images) of all types of defects to be possibly recognized, from all components to be inspected. This work presented a method that allows tackling both limitations at the same time. The proposed GAN model only needs defect-free images

to be trained, thus solving limitation (2) and, additionally, it also solves limitation (1) as the same model can be used for different components, requiring only one retraining with the images of the component to be inspected.

The proposal was validated using 2 different scenarios. The first exploited a real industrial manufacturing environment with multiple components. The second used images with acquisition and simulation of controlled defects. In scenario 1, the same method was used to inspect 3 different components, without the need for adaptations in the GAN. The result evidenced an increase in accuracy of inspection and identification of defects, compared with the current system used in the factory floor. In scenario 2, it was concluded that a wide variety of defects can be identified without the need for them to be part of the training step.

The obtained increase in the quality of inspection results in substantial gains along the manufacturing process, as each anticipated error implies less manual re-checking and avoids having to stop the production line. In future research, we aim to extend the GAN-based approach to cover other types of inspection, such as conformance tests, which is the kernel for successful applications in flexible manufacturing.

ACKNOWLEDGEMENTS

This research was supported by the Brazilian National Council of Scientific and Technological Development (CNPq), under grant number 309946 /2020-4, and partially by CAPES (Coordination for the Improvement of Higher Level or Education Personnel), financial code 001, FINEP (Funding Authority for Studies and Projects), Araucária Foundation, and Renault.

REFERENCES

- [1] D. Kiran, *Production Planning and Control: A Comprehensive Approach*. Butterworth-Heinemann, 2019.
- [2] Y. Yin, K. E. Stecke, and D. Li, "The evolution of production systems from industry 2.0 through industry 4.0," *International Journal of Production Research*, vol. 56, no. 1-2, pp. 848–861, 2018.
- [3] B. Esmailian, S. Behdad, and B. Wang, "The evolution and future of manufacturing: A review," *Journal of Manufacturing Systems*, vol. 39, pp. 79–100, 2016.
- [4] A. C. Caputo, P. M. Pelagage, and P. Salini, "Modeling errors in kitting processes for assembly lines feeding," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 338–344, 2015.
- [5] M. E. A. Boudella, E. Sahin, and Y. Dallery, "Kitting optimisation in just-in-time mixed-model assembly lines: assigning parts to pickers in a hybrid robot-operator kitting system," *International Journal of Production Research*, vol. 56, no. 16, pp. 5475–5494, 2018.
- [6] X. Feng, Y. Jiang, X. Yang, M. Du, and X. Li, "Computer vision algorithms and hardware implementations: A survey," *Integration*, vol. 69, pp. 309–320, 2019.
- [7] M. Quintana, J. Torres, and J. M. Menéndez, "A simplified computer vision system for road surface inspection and maintenance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 608–619, 2015.
- [8] L. Li, K. Ota, and M. Dong, "Deep learning for smart industry: Efficient manufacture inspection system with fog computing," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4665–4673, 2018.
- [9] A. Luckow, M. Cook, N. Ashcraft, E. Weill, E. Djerekarov, and B. Vorster, "Deep learning in the automotive industry: Applications and tools," in *2016 IEEE International Conference on Big Data (Big Data)*, pp. 3759–3768, IEEE, 2016.
- [10] M. Mazzetto, L. F. Southier, M. Teixeira, and D. Casanova, "Automatic classification of multiple objects in automotive assembly line," in *24th IEEE International Conference on Emerging Technologies and Factory Automation*, pp. 363–369, IEEE, 2019.
- [11] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriulka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, *et al.*, "An empirical evaluation of deep learning on highway driving," *arXiv preprint arXiv:1504.01716*, 2015.
- [12] D. Pomerleau, "Rapidly adapting artificial neural networks for autonomous navigation," in *Advances in neural information processing systems*, pp. 429–435, 1991.
- [13] M. Mazzetto, M. Teixeira, Érick Oliveira Rodrigues, and D. Casanova, "Deep learning models for visual inspection on automotive assembling line," *International Journal of Advanced Engineering Research and Science*, vol. 7, no. 1, pp. 473–494, 2020.
- [14] F. Di Mattia, P. Galeone, M. De Simoni, and E. Ghelfi, "A survey on gans for anomaly detection," *arXiv preprint arXiv:1906.11632*, 2019.
- [15] D. Wang, R. Vinson, M. Holmes, G. Seibel, A. Bechar, S. Nof, and Y. Tao, "Early detection of tomato spotted wilt virus by hyperspectral imaging and outlier removal auxiliary classifier generative adversarial nets (or-ac-gan)," *Scientific reports*, vol. 9, no. 1, pp. 1–14, 2019.
- [16] D. Mukherjee, A. Guha, J. M. Solomon, Y. Sun, and M. Yurochkin, "Outlier-robust optimal transport," in *38th International Conference on Machine Learning (M. Meila and T. Zhang, eds.)*, vol. 139 of *Proceedings of Machine Learning Research*, pp. 7850–7860, PMLR, 18–24 Jul 2021.
- [17] F. A. Saiz, G. Alfaro, I. Barandiaran, and M. Graña, "Generative adversarial networks to improve the robustness of visual defect segmentation by semantic networks in manufacturing components," *Applied Sciences*, vol. 11, no. 14, 2021.
- [18] A. Kusiak, "Convolutional and generative adversarial neural networks in manufacturing," *International Journal of Production Research*, vol. 58, pp. 1–11, 09 2019.
- [19] A. Deshpande, A. Minai, and M. Kumar, "One-shot recognition of manufacturing defects in steel surfaces," *Procedia Manufacturing*, vol. 48, pp. 1064–1071, 01 2020.
- [20] R. S. Peres, M. Azevedo, S. O. Araújo, M. Guedes, F. Miranda, and J. Barata, "Generative adversarial networks for data augmentation in structural adhesive inspection," *Applied Sciences*, vol. 11, no. 7, 2021.
- [21] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 5, pp. 530–549, 2004.
- [22] T. Malekzadeh, M. Abdollahzadeh, H. Nejati, and N.-M. Cheung, "Aircraft fuselage defect detection using deep neural networks," *arXiv preprint arXiv:1712.09213*, 2017.
- [23] Y.-J. Cha, W. Choi, and O. Büyükoztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.
- [24] D. Mery, D. Hahn, and N. Hirschfeld, "Simulation of defects in aluminium castings using cad models of flaws and real x-ray images," *Insight-Non-Destructive Testing and Condition Monitoring*, vol. 47, no. 10, pp. 618–624, 2005.
- [25] D. Mery and D. Filbert, "Automated flaw detection in aluminum castings based on the tracking of potential defects in a radioscopic image sequence," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 6, pp. 890–901, 2002.
- [26] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *International Conference on Machine Learning*, pp. 1747–1756, PMLR, 2016.
- [27] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.
- [29] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
- [30] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
- [31] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, pp. 214–223, PMLR, 2017.
- [32] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE international conference on computer vision*, pp. 2223–2232, 2017.
- [33] A. M. Deshpande, A. A. Minai, and M. Kumar, "One-shot recognition of manufacturing defects in steel surfaces," *Procedia Manufacturing*, vol. 48, pp. 1064–1071, 2020.

- [34] G. Wang, W. Kang, Q. Wu, Z. Wang, and J. Gao, "Generative adversarial network (gan) based data augmentation for palmprint recognition," in *2018 Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–7, IEEE, 2018.
- [35] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," in *IEEE international conference on computer vision*, pp. 3754–3762, 2017.
- [36] X. Wang, Z. Man, M. You, and C. Shen, "Adversarial generation of training examples: applications to moving vehicle license plate recognition," *arXiv preprint arXiv:1707.03124*, 2017.
- [37] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data augmentation using gan for improved liver lesion classification," in *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pp. 289–293, IEEE, 2018.
- [38] H.-C. Shin, N. A. Tenenholtz, J. K. Rogers, C. G. Schwarz, M. L. Senjem, J. L. Gunter, K. P. Andriole, and M. Michalski, "Medical image synthesis for data augmentation and anonymization using generative adversarial networks," in *International workshop on simulation and synthesis in medical imaging*, pp. 1–11, Springer, 2018.
- [39] F. Carrara, G. Amato, L. Brombin, F. Falchi, and C. Gennaro, "Combining gans and autoencoders for efficient anomaly detection," in *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 3939–3946, IEEE, 2021.
- [40] H. Arnelid, "Sensor modelling with recurrent conditional gans," Master's thesis, Chalmers University of Technology, 2018.
- [41] E. L. Zec, H. Arnelid, and N. Mohammadiha, "Recurrent conditional gans for time series sensor modelling," in *Time Series Workshop at International Conference on Machine Learning, (Long Beach, California)*, 2019.
- [42] A. A. F. Saldivar, Y. Li, W.-n. Chen, Z.-h. Zhan, J. Zhang, and L. Y. Chen, "Industry 4.0 with cyber-physical integration: A design and manufacture perspective," in *21st international conference on Automation and computing*, pp. 1–6, IEEE, 2015.
- [43] H. M. Do, K. C. Welch, and W. Sheng, "Soham: A sound-based human activity monitoring framework for home service robots," *IEEE Transactions on Automation Science and Engineering*, pp. 1–15, 2021.
- [44] M. Teixeira, J. E. Cury, and M. H. de Queiroz, "Exploiting distinguishers in local modular control of discrete-event systems," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1431–1437, 2018.
- [45] A. Pacana and K. Czerwińska, "Improving the quality level in the automotive industry," *Production Engineering Archives*, vol. 26, 2020.
- [46] K. Czerwińska, D. Siwiec, A. Pacana, and D. Malindžák, "Analysis of non-compliance of industrial robot arm parts," *Zeszyty Naukowe. Organizacja i Zarządzanie/Politechnika Śląska*, 2019.
- [47] A. Genaidy, A. Al-Shedi, and R. Shell, "Ergonomic risk assessment: preliminary guidelines for analysis of repetition, force and posture," *Journal of human ergology*, vol. 22, no. 1, pp. 45–55, 1993.
- [48] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [49] KEYENCE, "Keyence." <https://www.keyence.com.br/>, March 2020.
- [50] WENGLOR, "Wenglor." <https://www.wenglor.com/>, March 2020.
- [51] COGNEX, "Cognex." <https://www.cognex.com/>, March 2020.
- [52] I. D. Apostolopoulos and M. Tzani, "Industrial object, machine part and defect recognition towards fully automated industrial monitoring employing deep learning. the case of multilevel vgg19," *arXiv preprint arXiv:2011.11305*, 2020.
- [53] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: a survey," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 84–100, 2018.
- [54] S. Niu, B. Li, X. Wang, and H. Lin, "Defect image sample generation with gan for improving defect recognition," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1611–1622, 2020.
- [55] X. Jiang and Z. Ge, "Data augmentation classifier for imbalanced fault classification," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1206–1217, 2021.
- [56] G. Owen, *Game theory*. Emerald Group Publishing, 2013.
- [57] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.
- [58] D. M. Kreps, "Nash equilibrium," in *Game Theory*, pp. 167–177, Springer, 1989.
- [59] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *IEEE conference on computer vision and pattern recognition*, pp. 5485–5493, 2017.
- [60] I. J. Goodfellow, "On distinguishability criteria for estimating generative models," *arXiv preprint arXiv:1412.6515*, 2014.
- [61] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, pp. 2234–2242, 2016.
- [62] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.
- [63] H. Petzka, A. Fischer, and D. Lukovnicov, "On the regularization of wasserstein gans," *arXiv preprint arXiv:1709.08894*, 2017.
- [64] A. Berg, J. Ahlberg, and M. Felsberg, "Unsupervised learning of anomaly detection from contaminated image data using simultaneous encoder training," *arXiv preprint arXiv:1905.11034*, 2019.