Analysis and Mitigation of the Imbalance Impact on an Industrial Image Classification Dataset

Willian Jeferson Andrade
CPGEI, UTFPR
Curitiba, Brazil
willianandrade@alunos.utfpr.edu.br

André Eugenio Lazzaretti
CPGEI, UTFPR
Curitiba, Brazil
lazzaretti@utfpr.edu.br

Ricardo Eiji Kondo
PPGEPS, PUCPR
Curitiba, Brazil
ricardo.e.kondo@gmail.com

Heitor Silvério Lopes
CPGEI, UTFPR
Curitiba, Brazil
hslopes@utfpr.edu.br

Valmir de Oliveira
CPGEI, UTFPR
Curitiba, Brazil
valmir@utfpr.edu.br

***Abstract*—One of the most impacting problems in building machine and deep learning models for automated vision systems in industrial environments is the large variety of products in a real production line, which means that the dataset generated from the systems will be, most probably, imbalanced. Different approaches have been studied in the recent literature to reduce the effects of imbalanced data, still without a recommendation of a more adequate methodology for industrial scenarios. Hence, this paper compares three approaches for reducing the effects of imbalanced classes using a dataset of real images collected in an industrial production line: class removal, weight compensation, and data augmentation. We use a convolutional neural network as the backbone for the classifier of proposed method. Several comparisons are presented, emphasizing the advantages and limitations of each approach. Results show that data augmentation is the most promising approach for the evaluated dataset, improving the results and allowing the real-world application of the proposed method.**

***Index Terms*—Imbalanced dataset, Data Augmentation, weights compensation, Classification, Convolutional Neural Networks.**

I. Introduction

Industrial production systems have been modernized in recent years with the emergence of the Industry 4.0 concept. The collection of production data and the subsequent analysis are already present in several industrial branches. Such a procedure leads in the concept of a Cyber-Physical Production System (CPPS), derived from software and hardware applications in exchanging data between these systems. CPPS is known as an Industry 4.0 trend focused on flexibility for new products and new requirements [1]. According to [2], CPPSs have significant potential to further improve the condition-awareness of manufacturing machines and processes, reduce operational downtime, improve automation and product quality, and respond more timely to dynamically changing customer demands.

In the particular case of computer vision for industrial applications, in the context of CPPSs, which is the main focus of this work, the central idea is to provide image acquisition, defect detection, and classification. This technology is widely used because of its fast, accurate, non-destructive, and low-cost characteristics. Machine vision identifies objects mainly based on their color, texture, and geometric features. Hence, image acquisition quality, the number of acquired images, and the image processing algorithm determine the correct detection of defects and classification accuracy.

In the industrial environment, the images from the vision system can be easily collected and stored on a server. However, for many categories of industries, the problem is that there is a large variety of products in a real production line. This means that the dataset generated from the vision system will probably be imbalanced. According to [3], in industrial scenarios, the distribution of data across different classes is highly skewed, e.g., an instance can be 1000 times less frequent than another class.

In [4], the authors suggested that solutions for imbalanced classification problems can be categorized into two major approaches: Synthetic Minority Oversampling Technique (SMOTE) and Adaptive Synthetic Sampling Approach (ADASYN). On the one hand, SMOTE and its variants attempt to re-balance the dataset by generating synthetic samples of minority classes. ADASYN, on the other hand, uses a weighted distribution for different minority class examples according to their difficulty in learning.

In this context, several works have been addressed in the recent literature for industrial problems using SMOTE and ADASYN-based techniques, as presented in [5]. Still considering [5], datasets with faulty instances are difficult to obtain, since machines are normally operating in a healthy state. Generating artificial fault data in the laboratory, on the other hand, can be too expensive and, most of the time, might not present the reality of an event of an actual fault.

Regarding artificial data generation, one can emphasize the approach presented in [6]. The authors proposed to work with image fault detection and the recurrent problem of imbalanced datasets. Their work used a dataset of X-ray images from a public database with four different imbalanced classes of faults. They used Random Oversampling (ROS), Random Un-

dersampling (RUS), and SMOTE to create three new balanced datasets. The next step included a Histogram of Oriented Gradients (HOG) and deep Convolutional Neural Networks (CNN) for the proposed classification procedure. The best result was with the CNN model with SMOTE re-sampling, reaching an accuracy of 97.2%.

Similarly, in [7], the authors discussed an imbalanced image dataset of defects in the strip steel industrial process. Images were acquired through a computer vision system, which creates imbalanced datasets due to the low occurrence of defects. The contribution of their work was using a transfer learning with the VGG19 model combined with different sorts of algorithms such as Online Hard Example Mining and Adversarial Fast Region-based Convolutional Neural Network to make a real-time classification. Some difficulties included the speed of the production line, the quality of images, and the tuning of the chosen algorithms for best accuracy results.

In [8], on the other hand, the authors used a two-stage transfer learning training strategy for improving the accuracy on minority samples and a multi-scale convolutional neural network to extract multi-scale features of input images. They combined a CNN with a VGG16 transfer learning for training the dataset. They also included some additional stages for capturing image details that are normally not perceptible for a normal CNN architecture, thus improving the final results.

Considering the above described works, all they presented a particular set of techniques to reduce the effects of imbalanced data. Nonetheless, when we intend to classify real images collected by a computer vision system in the industry, some techniques may have better results than others. Therefore, the present paper compares three approaches to reducing the effects of imbalanced classes using a dataset of real images collected in a production line: (1) class removal, (2) weight compensation, and (3) data augmentation. We used a CNN as the backbone for the proposed method and present different comparisons, emphasizing the advantages and limitations of each approach.

The paper is organized as follows. Section II describes the methodology used to develop this work, while Section 3 presents the results obtained. In the last section, conclusions and future work are presented.

II. Methods

This Section details the proposed method and its most relevant steps. As previously discussed, we compare three approaches aiming at reducing the effects of imbalanced classes, using a dataset of real images acquired in a real industrial production line. Fig. 1 shows an overview of the workflow, which is detailed as follows.

A. Process Description

Before presenting the details of the proposed methodology, it is relevant to discuss some details of the industrial process related to our problem. However, it is worth mentioning that some details from the process and images were omitted due to the company's rights and manufacturer confidentiality.

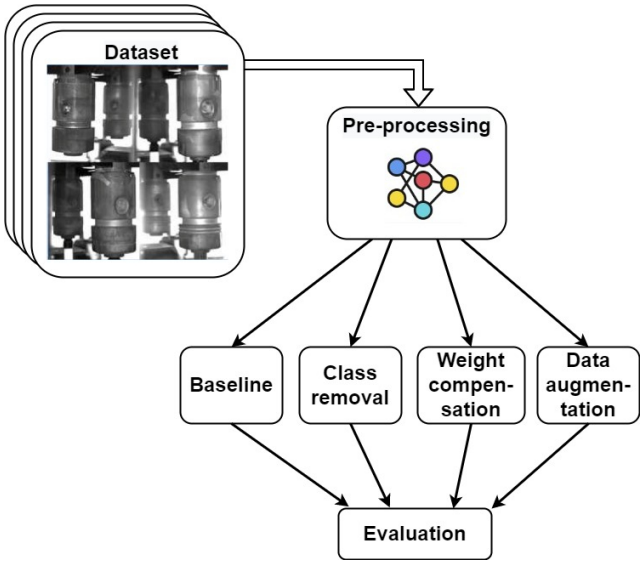

Fig. 1. Overview of the proposed workflow.

In general, when there is a new production order for a new part number, a setup must be performed at the production machines due to the differences among product families. The part is manually fed into a rotary table that carries the part into the machine. An industrial robot takes out the part from the table and position it in front of the inspection vision system. Then, the part is quickly verified for the vision system, which is trained for product mixture detection. However, due to light intensity differences caused by oiled parts, surface treatment variance and other conditions in the production line, the final result (Good or Bad) can be erroneous, which is not desired for an industrial process. Therefore, the ultimate motivation for the automation and analysis presented in this work is to improve the results from the visual analysis.

B. Dataset

The dataset used in this work is a real database stored in a server of manufacturing company. Fig. 2 shows some of the images in the dataset.

The dataset has 9 imbalanced classes, some of them with more than 1000 images, most of them around 1000, and one of them with less than 50 samples. It is shown that there is a large difference among classes. Table I shows how the data is distributed within classes.

C. Pre-processing

Image pre-processing is one of the steps for data preparation to make it more "understandable" to the neural network input layer. In this work, the first process was the normalization of image pixels in the range $\{0..1\}$. Next, the whole dataset was randomly divided into training and test, with a 70%/30% ratio.

D. Convolutional Network Architecture

CNNs were proposed and tested for the first time in 1998 to handle two-dimensional inputs, such as images, in which

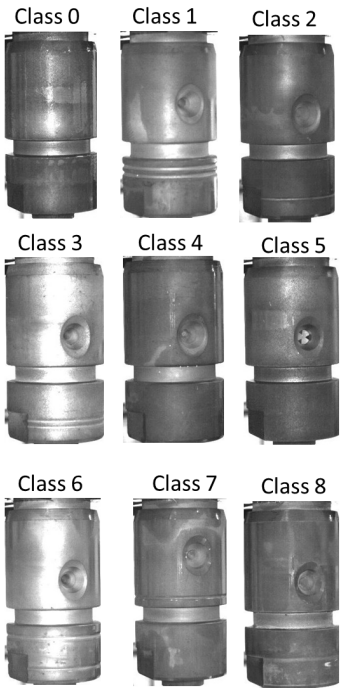

Fig. 2. Samples of the mechanic parts of several classes.

TABLE I
CLASS DISTRIBUTION.

Class	Images
Class 0	1569
Class 1	1116
Class 2	1283
Class 3	1306
Class 4	917
Class 5	924
Class 6	44
Class 7	1568
Class 8	1582

feature learning was achieved by stacking convolutional and pooling layers [9], [10]. As detailed in [11], CNN is a feed-forward neural network that consists of convolutional layers and fully connected layers, as well as associated weights and pooling layers. In general, CNNs have the ability to learn different levels of representation for high-dimensional data, so that they can learn abstract, essential, and high-order features from raw data. The convolutional layer applies a set of number filters to obtain the feature maps of input images. The pooling layer downsamples the data to reduce the feature dimensions of the input. Finally, the fully connected layers are responsible for computing the class score, i.e., the classification. Table II shows the architecture of the CNN chosen for this work.

E. Class Removal

We selected three of the most recurrent approaches to deal with imbalanced data. The simplest way to deal with the class imbalance problem is to treat the minority class of the dataset as an outlier and, then, eliminating it. In this case, there is a class with less than 50 images which is around 5% of the next minority class.

TABLE II
CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE.

Layer	Outputs	Parameters
Conv2D	6384788	224
MaxPool2D	3192398	0
Conv2D-1	31923916	1168
MaxPool2D-1	15911916	0
Conv2D-2	15911932	4640
MaxPool2D-2	795932	0
Dropout	795932	0
Flatten	149152	0
Dense	128	19091584
Dense-1	9	1161
Output	9	–

F. Weights compensation

The weights compensation method penalizes mistakes in samples of a given class i using weights different from one. Hence, higher class-weight means that more emphasis is given on that class. In the imbalanced context, classes that are less frequent than others, increase the class weight relative to classes that are more frequent. Table III shows the corresponding percentage of each class correlated to class 8 as majority. They are calculated as $w_j = \frac{S}{C \cdot S_j}$, where w_j is the weight for class j, S is the total amount of samples for all classes, C is the total number of unique classes in the dataset, and S_j is the total amount of images in the respective class.

TABLE III
WEIGHTS FOR CLASS.

Class	Weight
Class 0	0.73
Class 1	1.02
Class 2	0.89
Class 3	0.88
Class 4	1.25
Class 5	1.24
Class 6	26.03
Class 7	0.73
Class 8	0.72

G. Data Augmentation

Data augmentation is a compelling method to reduce over-fitting problems [12]. Data augmentation introduces artificial images to the dataset by either warping or oversampling. Data warping introduce transformations in the existing images, however preserving the labels. On the other hand, oversampling creates synthetic instances and adds them to the training set [13]. In this work, we use the SMOTE approach. The main idea is to create new artificial images for each class until each one reaches the same number of images in the majority class. The original dataset contained more than ten thousand images, distributed in nine classes. With the data augmentation oversampling process, all classes will increase the total number of images to more than fourteen thousand.

In the industrial production line, a robot shows the parts to the vision system to take the photos. The robot has a high accuracy in terms of positioning. Based on this fact, changes in position and rotation are considered negligible. By inspecting some images, the only features that seem to be more likely for data generation is brightness, blur, and noise. Fig. 3 shows how subtle are the differences between an original image and a synthetic image with brightness, blur, and noise augmentation.

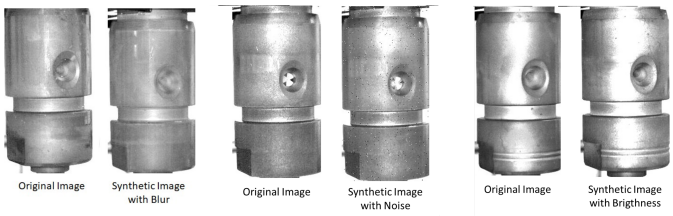

Fig. 3. Brightness, blur, and noise augmentation.

H. Evaluation metrics

We used different metrics to evaluate our results. The first one is the Precision, defined as:

$$Precision = \frac{TP}{TP + FP}, \tag{1}$$

in which, TP are the True Positives and FP the False Positives. The next metric is the Recall or Sensitivity, formulated as:

$$Recall = \frac{TP}{TP + FN}, \tag{2}$$

where FN correspond to the False Negatives. With those metrics, one can define the F1-score, which is simply the harmonic mean between precision and recall:

$$F1score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{3}$$

III. Results

In this Section, the results of the application of the four methods presented before (Section II) are explained in details.

A. Baseline

It is essential to have a baseline model for future comparison with the other methods that will be tested. For this baseline, we adopt the original distribution of the dataset. The classification results are shown in Table IV, and the corresponding confusion matrix is shown in Figure 4.

By analyzing the confusion matrix of Fig. 4, it is observed that class 8 presented the most relevant prediction errors. There is a significant misclassification between this class and classes 1 and 3. By inspecting the images in each of these classes, we can notice that the family of products analyzed have close similarities, as shown in Figure 2. Possibly, this is the main cause of prediction errors and relatively low F1-Score shown in Table IV.

TABLE IV
Baseline results.

Class	Precision	Recall	F1-Score	#Images
C0	0.998	0.989	0.993	452
C1	0.889	0.824	0.855	341
C2	0.997	0.995	0.996	376
C3	0.934	0.932	0.933	395
C4	1.000	0.982	0.991	281
C5	0.978	0.996	0.987	268
C6	0.941	1.000	0.970	16
C7	0.991	0.998	0.994	441
C8	0.841	0.889	0.864	476
Macro avg	0.952	0.956	0.954	3046
weighted avg	0.950	0.949	0.949	3046

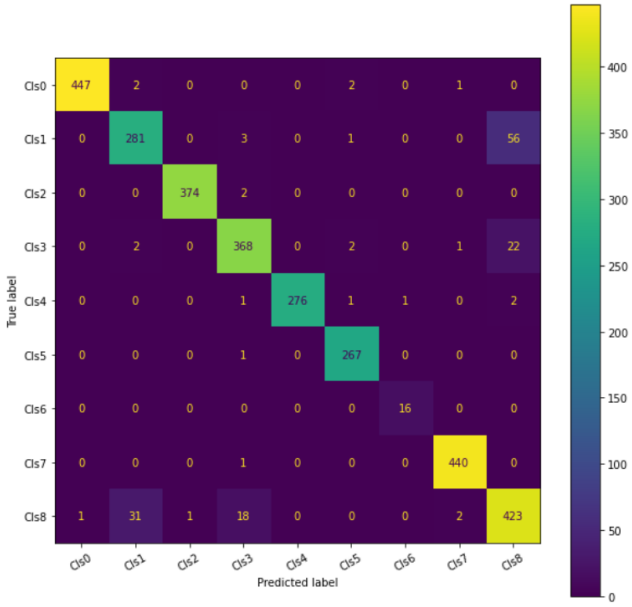

Fig. 4. Confusion Matrix - Baseline.

B. Class removal approach

Once the minority class is much smaller than the other classes, it is possible to remove that class from the original dataset. Table V shows the results for the class removal approach.

TABLE V
Results for the class removal approach.

Class	Precision	Recall	F1-Score	#Images
C0	0.998	1.000	0.999	452
C1	0.881	0.979	0.928	341
C2	0.997	0.997	0.997	376
C3	0.950	0.906	0.927	395
C4	0.993	0.979	0.986	281
C5	0.985	0.996	0.991	268
C7	0.989	0.998	0.993	441
C8	0.940	0.893	0.916	476
Macro avg	0.967	0.969	0.967	3030
weighted avg	0.966	0.966	0.966	3030

By inspecting the confusion matrix in Fig. 5, we can notice

that there are more minor errors than in the baseline, primarily for class 1. However, this approach was not to successfully sort out the class 8, which still have several prediction errors. Although the class removal approach results seem to show better results, a real dataset from a factory production line could take samples from another period where the minority class might be more expressive and could not be removed.

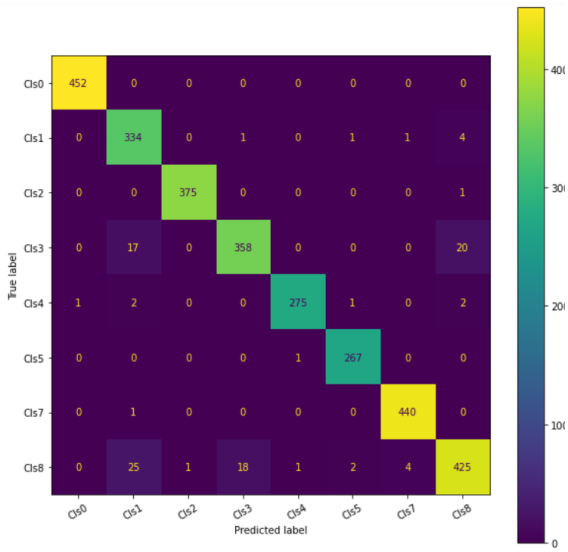

Fig. 5. Confusion matrix - Class Removal.

C. Weights compensation approach

The next approach compensates the difference of instances among classes by using relative weights, based on the majority class. First, it is needed to check how many samples there are in each class. Then, the difference in the percentage of each class related to the majority one is calculated, which will be the weight compensation. Table VI shows the results for this approach. It is observed that results are somewhat similar compared to the baseline.

TABLE VI
RESULTS FOR THE WEIGHT COMPENSATION APPROACH.

Class	Precision	Recall	F1-Score	#Images
C0	0.996	0.998	0.997	452
C1	0.902	0.921	0.911	341
C2	0.995	0.995	0.995	376
C3	0.967	0.954	0.961	395
C4	1.000	0.989	0.995	281
C5	1.000	0.989	0.994	268
C6	0.889	1.000	0.941	16
C7	0.998	1.000	0.999	441
C8	0.916	0.916	0.916	476
Macro avg	0.962	0.974	0.968	3046
weighted avg	0.969	0.969	0.969	3046

The confusion matrix (Fig. 6) shows that improvements were achieved for class 1, in which almost all images were classified accurately. However, class 8 resulted in the worst results. In the baseline, 423 images were correctly classified, but only 210 out of 476 were correctly classified in this method, less than 50%.

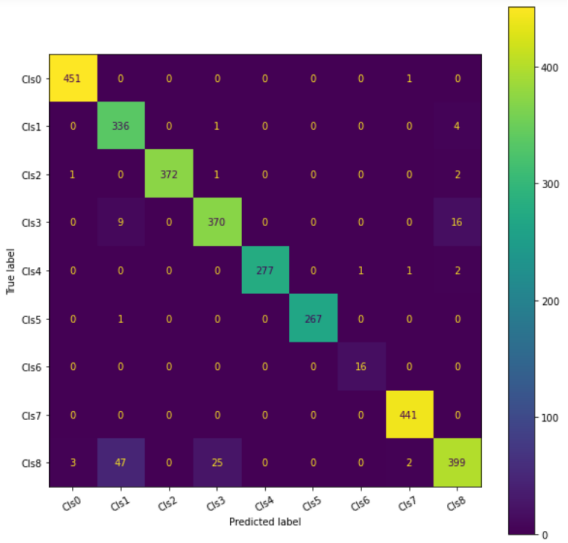

Fig. 6. Confusion matrix - Weights Compensation.

D. Data augmentation approach

Data augmentation introduces new artificial images to the dataset by either warping or oversampling [13]. Data warping transforms existing images, such as their label is preserved. On the other hand, oversampling creates synthetic instances and adds them to the training set. We selected the synthetic minority oversampling technique (SMOTE) in this work. Results of classification with data augmentation are shown in Table VII.

TABLE VII
RESULTS FOR THE DATA AUGMENTATION APPROACH.

Class	Precision	Recall	F1-Score	#Images
C0	0.998	0.991	0.994	474
C1	1.000	0.998	0.999	474
C2	0.986	0.980	0.983	475
C3	0.996	1.000	0.998	475
C4	0.976	0.942	0.959	474
C5	0.996	1.000	0.998	475
C6	1.000	0.998	0.999	474
C7	1.000	0.996	0.998	474
C8	0.929	0.974	0.951	475
Macro avg	0.987	0.986	0.986	4286
weighted avg	0.987	0.986	0.987	4286

In the confusion matrix of Fig. 7, it is observed an improvement in the classification accuracy. In this approach the classifier was not severely biased towards the majority class. However, there are some classification errors between in classes 4 and 8 (25 images). A deeper analysis of this issue reveals that the products from those classes have very similar shapes, which might sometimes make images hard to distinguish. Notwithstanding, with this approach almost all prediction errors were reduced to zero, except for a single

class. The problem of predicting class 8 was reduced, which was more complex to achieve in the previous methods.

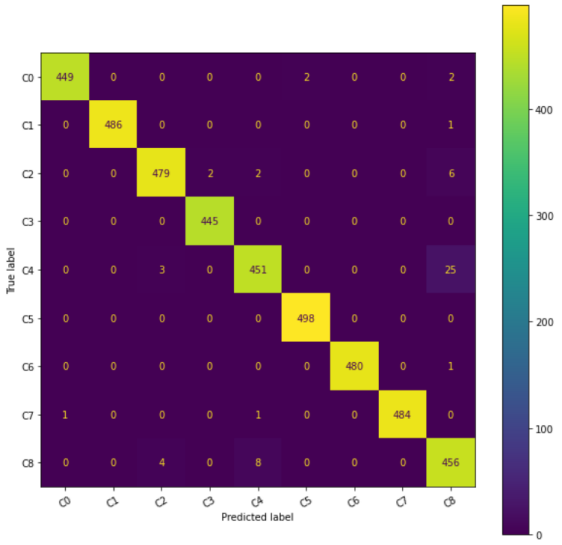

Fig. 7. Confusion matrix - Data Augmentation.

E. General comparisons

Table VIII summarizes all the F1-score values for each approach. We can notice that the best result was achieved by using data augmentation. Nevertheless, the weights compensation method results were relatively close, suggesting that this methodology could, also, be useful for solving the industrial image classification problem.

TABLE VIII
AVERAGE F1 SCORE.

Method	F1-Macro	F1-Weighted	#Images
Baseline	0.954	0.949	3046
W/O minority class	0.967	0.966	3030
Weights compensation	0.968	0.969	3046
Data augmentation	0.986	0.987	4286

IV. Conclusion and Future Work

In this paper we compared tree approaches for mitigating the imbalance problem of an industrial image classification problem. This is a real problem daily faced in production lines. We used a CNN to extract features and classify images into 8 different (but, quite similar) classes.

First, a simple baseline was created, using the raw imbalanced dataset. Although, at first sight, the results seem to be good, they are not satisfactory considering the industrial production standards. The first, and simplest, approach used was removing the minority class. Next, a weight compensation to balance the loss function for minority classes was applied to change the imbalanced dataset to a balanced one. Finally, the data augmentation was used to balance the number of examples per class. Overall, the data augmentation showed the best results in terms of F1-score, with the weights compensation method achieving similar results, but less images are required.

An immediate future work will test the model in the real environment of the production line, and collect more data for future analysis and comparison to the current solution. Actually, the computer vision system has an error of around 3% of false rejection, which directly influences productivity and performance indicators. If this model would be able sort out this problem with the precision shown, production will increase, leading to higher production volumes, which is the final target for this work.

References

[1] M. Seitz, F. Gehlhoff, L. A. Cruz Salazar, A. Fay, and B. Vogel-Heuser, "Automation platform independent multi-agent system for robust networks of production resources in industry 4.0," *Journal of Intelligent Manufacturing*, 2021.

[2] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *Journal of Manufacturing Systems*, vol. 48, pp. 144–156, 2018.

[3] E. Lin, Q. Chen, and X. Qi, "Deep reinforcement learning for imbalanced classification," *Applied Intelligence*, vol. 50, no. 8, pp. 2048 – 2052, 2020.

[4] A. Duan, L. Guo, H. Gao, X. Wu, and X. Dong, "Deep focus parallel convolutional neural network for imbalanced classification of machinery fault diagnostics," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 11, pp. 8680–8689, 2020.

[5] T. Zhang, J. Chen, F. Li, K. Zhang, H. Lv, S. He, and E. Xu, "Intelligent fault diagnosis of machines with small & imbalanced data: A state-of-the-art review and possible extensions," *ISA Transactions*, vol. 119, pp. 152–171, 2022.

[6] W. Hou, Y. Wei, Y. Jin, and C. Zhu, "Deep features based on a DCNN model for classifying imbalanced weld flaw types," *Measurement*, vol. 131, pp. 482–489, 2019.

[7] X. Wan, X. Zhang, and L. Liu, "An improved VGG19 transfer learning strip steel surface defect recognition deep neural network based on few samples and imbalanced datasets," *Applied Sciences*, vol. 11, p. 2606, 03 2021.

[8] J. Liu, F. Guo, H. Gao, Z. Huang, Y. Zhang, and H. Zhou, "Image classification method on class imbalance datasets using multi-scale cnn and two-stage transfer learning," *Neural Computing and Applications*, vol. 33, pp. 14 179–14 197, 2021.

[9] J. Wang, Y. Ma, L. Zhang, R. X. Gao, and D. Wu, "Deep learning for smart manufacturing: Methods and applications," *Journal of Manufacturing Systems*, vol. 48, pp. 144–156, 2018.

[10] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5990–5998, 2018.

[11] J. Yang, S. Li, Z. Wang, H. Dong, J. Wang, and S. Tang, "Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges," *Materials*, vol. 13, no. 24, 2020.

[12] N. Aquino, M. Gutoski, L. Hattori, and H. Lopes, "The effect of data augmentation on the performance of convolutional neural networks," in *Proc. XIII Brazilian Congress on Computational Intelligence*, Niteroi, Brazil, 2017, pp. 1–12.

[13] C. Shorten, Khoshgoftaar, and T. M., "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, pp. 2196–1115, 2019.