Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 51 (2020) 1192–1199

Procedia MANUFACTURING

www.elsevier.com/locate/procedia

30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021)
15-18 June 2021, Athens, Greece.

Is Deep Learning ready to satisfy Industry needs?

Paolo Tripicchio*, Salvatore D'Avella

Department of Excellence in Robotics & AI, TeCIP Institute, Scuola Superiore Sant'Anna, Pisa, Italy

Abstract

The impact that Artificial Intelligence is having in modern society is undeniable. Many companies are now using AI to improve the throughput and automate their processes. But the challenge is that Artificial Intelligence is both a source of enthusiasm and skepticism for industries. The manuscript points out the main causes of skepticism giving at the same time some possible technical solutions to exploit at the best the potentialities of AI even in those conditions in which the data are imbalanced and the object classes are not well separated. This work also emphasizes the delicate relationship between artificial intelligence, researchers, and industries, and tries to give an overview of a possible trade-off between the two parties. The document ends up proposing an 'interpretable learning' approach that can be exploited as a common language between the two players. The desirable practice would be to make AI explainable, provable, and easily understandable by the companies.

Keywords: Deep Learning; Quality Inspection; Data Imbalance; Transfer Learning;

1. Introduction

After a winter stage in the development of neural networks, the rapid growth in computing power put the reflectors on AI again. The impact that Artificial Intelligence is having in modern society and the interest in searching for its application in the various sectors is undeniable. Nowadays it is not only an interesting field of research but it is also a useful tool that industries want to employ to improve the throughput and automate their processes. But the challenge is that Artificial Intelligence is both a source of enthusiasm and skepticism for industries. This manuscript aims to point out the delicate relationship between artificial intelligence, researchers and industries, and tries to give an overview of a possible trade-off between the two parties.

There are two main reasons for that:

- Deep learning is a data-driven technology, so data are important to obtain good performance. Problems solved using AI are as good or as bad as the data they are trained on. But what does it means good or bad? and how much data are enough?
- It is difficult to explain to companies what a network does and why it does, so they see AI as a black box. Sometimes happens that for reasons related to the previous point the machine is not able to accomplish the task properly and the manager wants to know why it happens and how it is possible to integrate such processes without tangling the company Key Performance Indicators (KPI).

The manuscript addresses these key points explaining the causes of the skepticism and giving some possible technical solutions to exploit at the best and in the correct way the potentialities of AI even in those conditions in which the data are imbalanced and the object classes are not well separated.

The document finally proposes an approach for giving a common language to the two players, the researcher from one side and the industry guy from the other. The first can exploit this approach as a pipeline to follow for sharing a common language with the other party. The desirable practice would be to make AI explainable or interpretable so that the algorithm decision would be understood by the companies to facilitate decision-making.

* Corresponding author. ORCID: 0000-0003-3225-2782
E-mail addresses: p.tripicchio@santannapisa.it (Paolo Tripicchio)., s.davella@santannapisa.it (Salvatore D'Avella).

10.1016/j.promfg.2020.10.167

2. Any Scraps there?

A good part of Engineers life is usually spent working or researching for companies, with a special focus on tasks of industrial production of goods and materials. To exploit to the best such relationship, both the parties, engineers and company guys, should speak and hopefully think the same language.

In this context, starting from a typical dialogue between the two parties, this essay aims to discuss diverging opinions and possibly introduce existing solutions and a common playground where the language of Deep Learning can be understood and outcomes of the application of such technology discussed with a common view by both the parties.

In most of the situations, the demand from the industrial contractor of a DL project is always "We would like to improve our business using artificial intelligence methods. Nowadays our daily outcome for this task is *a certain amount of production units* and the task is currently performed by human personnel". Perfect, it seems that there is some margin of improvement, at least trying to remove human personnel from repetitive and boring tasks and employing them elsewhere.

Then, it usually comes a visit to the manufacturing plant, an analysis of the process to optimize, and unfortunately the phase of the project where the engineers state their request to the company to proceed with the development. "From our preliminary analysis, We concluded that the project could benefit from a Deep Learning architecture able to classify production defects and perform an automatic quality assessment of the product. To train the DL model, We estimate that a sufficient number of data samples that allows achieving the requested performance metrics would be in the order of tens of thousands of samples". "Perfect", says the company guy, "We have hundreds of thousands of good samples and We can provide them to you easily". Here, people that work on the subject should have already spotted the critical issue.

"We are happy to hear that, but", a little pause to refresh in mind what to say to the customer and trying no to break their excitement, "the number of samples we had in mind should represent equally both the good products and the scrap ones".

This is the exact moment when it is possible to scout discomfort in the eyes of the listener. "But We dont have so many scraps. Our company would have shut down if We had such an amount of defective products!".

This is the first problem that should be faced when dealing with industrial production. Differently from laboratory or research setups where classification tasks are run over well defined and distinguished classes, industries that operate on the market have specific needs, and they usually want to classify defects on their products. This, from one side, means that the separation between classification targets is not ideal and most of the samples share features between them (see figure 1). Secondly, industries are efficient and the number of defective products that are produced during months could be counted on the fingers of a hand.

Despite any point the engineering team will present in their favor, the company guy will always answer that "Human personnel can do this, even having seen only a few scrap examples and even if the defects have not been quantitatively defined. They are always able to decide whether a product is scrap or not, just from experience". Nothing to say, it makes sense. And it should make sense also that a solution to the issue could be found in the AI domain. This, in turn, is not implying that an AI expert should provide a network to rule them all. This is not possible by the way, as the no free lunch theorem states [1] any two ML algorithms are equivalent when their performance is averaged across all possible problems, but, a specific solution can be tailored to satisfy each specific project needs.

Possible approaches that have been used in the literature to overcome the limitation imposed from poor availability of data are discussed hereafter.

3. Facing imbalanced data sets with low samples

Basically, problems regarding defect detection are classification problems. Following the first approaches combining traditional and machine learning techniques [2], methods based on deep learning have been encouraged after the development of AlexNet [3]. Convolutional Neural Networks(CNN) have been widely adopted for diverse applications in industries ranging from object detection for pick and place [4] to automating optical quality inspection [5]. Such networks, however, often display an intolerable problem that is their need for large amounts of labeled data necessary to properly train their parameters. In fact, a prerequisite for the training of CNNs is the availability of adequate training data. What does it mean? The ideal case is the one in which ten of thousand samples for each class are available, the inputs for each class are balanced and the classes are well separated each other. In this way, it is possible to provide as input a representative set of examples of the entire input space and the network cannot be confused by similarities among classes or by an uneven distribution of the inputs. But unfortunately, in a real industrial scenario, for example for defect detection in a product line, it is possible to obtain many good samples but too few scraps. It is then really easy to fall into the problem of imbalanced datasets. The worst-case scenario is, for sure, when the dataset is imbalanced with very few scrap samples and the classes are very close to each other in the input space. Figure 1 clarifies visually this concept. Defects like scrap number 3 are easily separable while other kinds of defects pose problems in the classification. In the following are presented the main state of the art techniques and strategies that are useful in such situations.

3.1. Sampling Strategies

In the case of imbalanced datasets, having at disposal only a few samples for the scrap class, it could be convenient to use as many goods as the available scraps you collected. This approach is called ***undersampling*** in literature. This is feasible, of course, if the number of scrap examples is sufficient for the training task. On the other way around, in order not to reduce the good examples class, it is possible to present to the network,

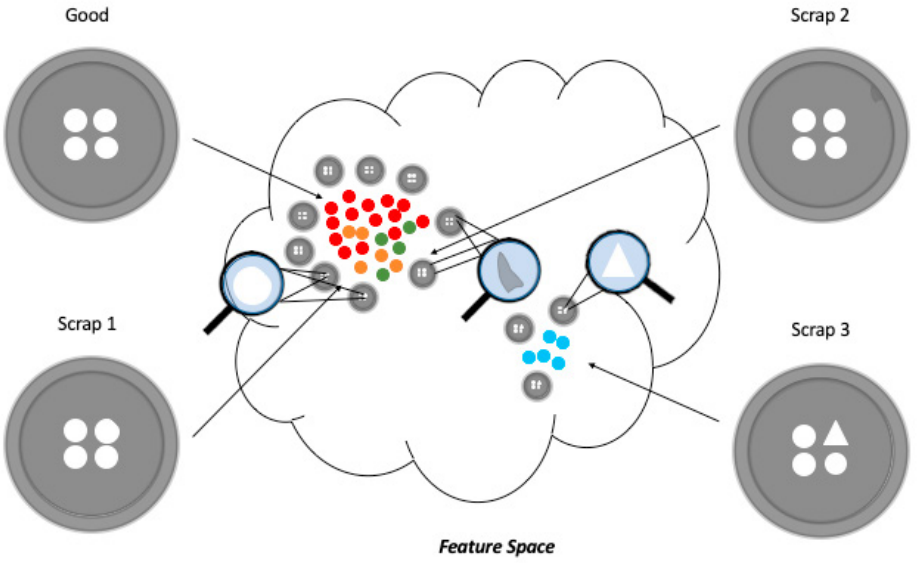

Fig. 1. Imbalanced Data-set with few scrap samples. Classes Scrap 1 and Scrap 2 are close to the Good class, while the Scrap 3 class is easy separable from the others in the feature space.

multiple times, the available scraps trying to reach the same number of good samples. This second approach is known in the literature as ***oversampling***. This is a bit dangerous since it is easy to overfit the network given the poor representation of the input space that usually cannot cover completely the possible cases. However, there are situations in which they are precious means to improve the performance of the classifier as the work proposed by [6]. Figure 2 gives a graphical visualization of the two techniques.

3.2. Data Augmentation

Another technique, improving in robustness, is data augmentation. Traditional approaches involve cropping, rotation, mirroring, scaling, color-shift [7]. They have been exploited in many defect inspection methods [8, 9]. In this case, the samples are augmented based on the available data and it is possible to generate only samples with a strong correlation between them with the risk of overfitting on a small dataset. But if the augmentation is done properly, it could improve the performance of the classifier. Figure 3 shows some of the most common affine transformation used for modifying the images and enlarging the input data.

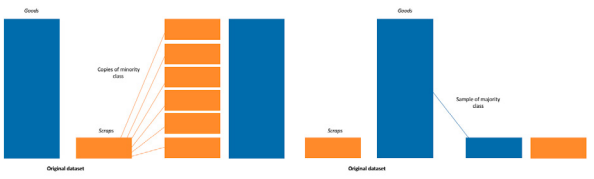

Fig. 2. Oversampling (left) and Undersampling (right)

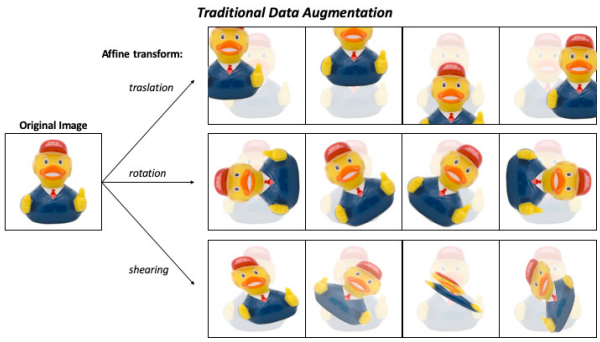

Fig. 3. Typical affine transform for data augmentation

Other techniques for augmenting the data-set have been experimented like propagating the input data through an encoder-decoder network where different transformations featured with random noise are applied [10]. Figure 4 shows roughly the idea of the architecture behind this technique.

Another method that it is worth to notice is the generation of virtual samples. It has been successfully exploited in [11] for face reconstruction. Figure 5 depicts the potentiality of the method.

In order to augment the input data, given the lack of training samples that cover the whole input feature space, one can think to create synthetic images to train the models using a Generative Adversarial Network(GAN) [12] or the most recent Conditional GAN (cGAN) [13]. This is a very promising alternative for facing the lack of sufficient training data or the case of unbalanced datasets. However, this is very time consuming and requires to take into account all possible configurations and boundary con-

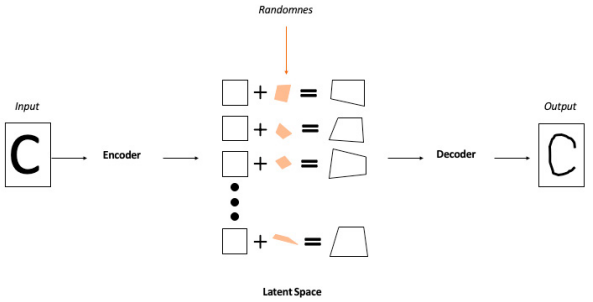

Fig. 4. Data augmentation exploiting encoder-decoder propagation

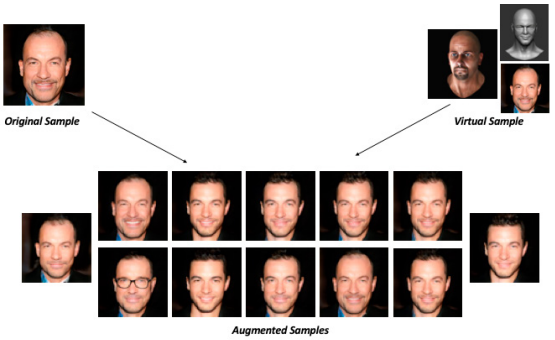

Fig. 5. Data augmentation using virtual samples. The real and augmented samples were generated by TL-GAN (transparent latent-space Generative Adversarial Network) [15]

ditions for generating samples as close as possible to real ones. For instance, if the intention is to augment the number of defects, there is the need for focusing on the defected part of the image leaving untouched the defect-free area. The problem is that modifying just the defect part could lead to a blending and blurring phenomenon in the defect neighbor pixels generating a not realistic sample.

3.3. Transfer Learning

As humans can learn from experience and transfer the concept learned between different application scenarios, similarly, a DL architecture can be trained using the so-called transfer learning paradigm. The working principle is to train the network on a larger dataset for learning how to extract the features and then adjust only the final classification layers using the appropriate dataset for the specific task. Knowledge transfer breaks the fundamental assumption that the data presented to the network during the training phase must be in the same feature space of the ones presented in the inference phase. A feature extractor obtained using transfer learning would be able to extract generic convolution features that can be exploited in different tasks. So, if we are interested in detecting, for instance, surface scratches on a product, and the amount of available sample images for the scratched products is not sufficient for proper training, it will be possible to train the network with examples of other products that present a scratched surface. The learned kernels can then be transferred to the final architecture implementation. A successive fine-tuning procedure will be responsible for successfully transferring the learned kernels to the domain of interest [14]. For these reasons, transfer learning achieves better classification results reducing the amount of training time and training labeled data required. Successful implementation of such an idea in an industrial context can be found in [5]. In this work, the authors were able to train a classification network starting with few hundreds of scrap images still obtaining accuracy and recall metrics of 97.22% and 100% respectively.

With well-optimized processes, it is often not possible to obtain a sufficiently large set of scraps samples for training the CNN for classification and most of the time the training objective moves from defect classification to anomaly detection. Deep metric learning uses deep neural networks to directly learn a similarity metric, rather than creating it as a byproduct of solving a classification task [16]. They are well suited for tasks in which the amount of object classes is perhaps endless and classification is not applicable. This approach would not require defective samples for training if the geometry and surface appearance of a part are well defined and distinct from each other. The idea is to calculate the distance of the proposed sample to an ideal prototype. Unfortunately, textured objects present surface appearance and properties that are stochastic.

3.4. Outliers

Another important aspect to take into account is to present correct examples for the learning process. As it happens with humans, when learning new concepts if the concept is not clearly defined it could lead to fuzzy assumptions that in the end could produce wrong outcomes. This, in turn, is reflected in the choice of a proper dataset for training. In particular, when dealing with data provided by some sensing technology installed in an industrial site, it is necessary to verify the correctness of the data samples and avoid possible sources of classification mistakes ”cleaning” the dataset. The cleaning process should detect possibly the presence of outliers (wrong data association of a sample with a class) and also spot possible ”borderline” samples that could confuse the learning process. This may often happen in industrial processes where the definition of a certain class is not given with quantitative metrics but with a qualitative evaluation. In fact, it is unfortunately common that different quality experts in the same industrial process classify the same product as belonging to different classes. If the same concept is imbued into the DL architecture, the learning process will probably worsen the decision process.

In most of the cases, in order to let the neural network work properly, the training could benefit of a pre-processing stage on the input data, encoding the input in the most suitable form for the network. In those cases, the help of professionals of the sector is needed for interpreting well the input to feed into the network for filtering and pre-processing the data or for performing some optimization.

This is extremely important in situations where the introduction of new machinery or the change of one or more com-

ponents or procedures in a manufacturing scenario (i.e. change of material producer, change of parameters, etc.) induces in the produced pieces some undesired alteration. These artifacts will certainly affect quality inspection results. In these cases, the use of a data filtering technique is recommendable. In [17] a pre-filtering stage has been introduced by the authors to smooth out the presence, on a metallic surface, of nuances due to a change in the welding process.

3.5. Performance Metrics and Loss functions

When the dataset is heavily unbalanced, it is necessary to consider correct metrics to evaluate the performance of the trained algorithms. In this context, authors in [18] propose a balanced accuracy statistic computed as:

$$Balanced\ Accuracy = \frac{Recall + Specificity}{2}. \quad (1)$$

A different approach could be to directly scale the confusion matrix terms based on the relative support of each class as in [17].

Other studies propose the modification of the loss function to account for class imbalance. In particular, the *binary cross-entropy loss* is a common choice for classification tasks. Starting from such a formulation, a balanced cross-entropy is introduced in [19]. In this solution, the part corresponding to the dominant class in the loss function is multiplied by the fraction of the less dominant class.

A second alternative is proposed in [20] where the authors address the problem of class imbalance by reshaping the standard cross-entropy loss such that it down-weights the loss assigned to well-classified examples. The resulting loss is called *Focal loss*.

In the balanced cross-entropy loss, while the importance of positive and negative examples becomes balanced, the method does not differentiate between easy/hard examples. The Focal loss instead, focus the training on hard negatives. It is a dynamically scaled cross-entropy loss, where the scaling factor decays to zero as the confidence in the correct class increases.

4. Explainable ML

Differently from the first AI systems that were composed of few neurons and very few connection layers, and thus, easily interpretable, modern AI systems employ more complex networks like Deep Neural Networks (DNNs) that are instead opaque decision systems since it is hard to understand what happens under the hood even for the network designers. The power of these neural networks stems from the combination of learning algorithms and the interactions of thousands of neurons in hundreds of layers within a huge parametric space. Such a combination of factors makes modern networks considered as black-box models [21]. The trend is to make light on the dark aspects of complex networks, giving transparency to the mechanisms by which the model works [22].

According to Breiman [23] accuracy generally requires complex prediction methods, thus simple and interpretable functions can not make the most accurate predictors. Black-box models are not interpretable unless the data have low dimensionality.

Given the diffusion of bigger datasets in many application domains, black-box supervised learning models, like neural networks, complex trees, random forests, local kernel-weighted methods, and many others, are being commonly employed in favor of more transparent linear regression models for capturing nonlinear behaviors. The advantage of using black-box models is gaining efficiency and practicality since these models are more accurate and can learn the features of the input samples autonomously. Such an advantage is paid at the price of loss in interpretability of the predictor variables on the predicted response. However, for many applications in different contexts like medicine, finance, and industry, understanding the effects of the predictors is crucial. As the use of black-box machine learning models increases, so it does the demand for transparency from the various stakeholders in AI [24].

This demand is dictated from one side by regulation requirements, from the other side to get insights from the models and possibly make scientific/business findings or understand where a process went wrong. To reach this specific goal, it is of utter importance to understand which variables mainly contribute to the outcome of a prediction, to retrieve an input-output relationship for each important variable, and to model the interaction between them.

The recent works on the interpretability of neural networks concern the input and the output layers. Perhaps, this is prevalently due to the fact that these layers have a precise meaning. Indeed, in computer vision, the input layer usually represents the values of three channels (red, green, and blue) for every pixel in the input image, while the output layer shows the class labels and their associated probabilities. Nevertheless, to get the final result in the output layer, the data traverses the hidden layers that constitute the real potentiality of the neural network model because, at each layer, the network learns a new representation of the input, depending on the activation of the neurons. The difficult part stays in explaining the reasons behind such activations since the networks usually use abstract vectors.

A common way of qualitatively interpret the representation learned by the first layer of a deep architecture is by visualizing the filters learned by the model. Many times, these filters represent stroke detectors on digit data, or Gabor filters (edge detectors) on natural image patches [25]. If early layers encode low-level features like edge or curve, later layers would learn higher-level features like mouth, nose, or eyes in face recognition applications. The research community is split on whether this is true. Many researchers see a meaningful and understandable relationship among neurons as an almost trivial fact [26, 27]; many others do not believe that latent variables could be meaningful [28, 29].

One of the main problems that make it difficult to visualize and understand neural networks is the relationships among the variables and the extremely high-dimensionality of the parameter space.

In the last years, many works have been proposed to improve the interpretability of a particular statistical learning procedures output. Rao and Potts [30] proposed a method that visualizes the decision boundary of bagging decision trees. Tzeng [31] tried to visualize the layers of neural networks to study the dependencies between the inputs and the outputs and give information about the classification uncertainty. Jakulin et al. [32] analyzed the interpretability of support vector machines using nomograms that provide a graphical representation of the contribution of the variables. Breiman [33], through randomization of out-of-bag observations, estimated a metric for the importance of the variables for Random Forests (RF). The most popular method for visualizing the results of predictors with black box supervised learning models is the Friedman's Partial Dependence (PD) Plots [34]. For data with small correlations, PDPs can reliably estimate the relationships between the predictors and the fitted response in terms of nonlinearities, directions, and interactions. Other diagnostic tools have been proposed by extending PDP. Accumulated Local Effects (ALE) plots [35] do not require PDP unreliable extrapolation with correlated predictors and are less computationally expensive. Individual Conditional Expectation (ICE) plots identify interactions analyzing the connection of the predicted response on individual features for each sample point [36]. The ANOVA decomposition of ICE plots measures the variable importance and is consistent with the global sensitivity indices of Sobol [37]. To make the model easily explainable, Additive Index Models (AIM) with neural networks ridge functions have been presented as eXplainable Neural Networks (xNN) [38]. AIM decomposes a complex function into the linear combination of multiple component functions, and as such represents a good model for explainability. Current research investigates upon the naive version of xNN to enhance its explainability capabilities [39]

All these methodologies aim at making the prediction process explainable. However, the focus of the proposed approach is not on the interpretation of the process but the interpretation of the outcomes. The industrial staff usually is not interested in how the computation is performed, they are interested in understanding the meaning of a prediction that a simple percentage value (usual network output) cannot represent by itself. The next section will introduce for such a purpose a possible approach to face the problem.

5. A Common Language

What has been discussed in the previous paragraphs depicts a complex and intricate scenario where industrial needs, aims, and desiderata, together with engineering knowledge and mathematical approaches should converge and find a common goal, vision, and understanding.

The current solutions employed at industrial production facilities for quality and defect analysis involve classical computer vision inspection techniques, capturing images of the products in the analysis at several stages on the production line. Typical graphical user interfaces and usually stored performance data includes the current image of the product at the specific stage, quantitative measurements of the characteristics of the product and current efficiency of the process typically measured as Overall Equipment Effectiveness(OEE). These are the information that industry people understand and are used to deal with. Black-boxes architectures that provide just a probability of belonging to a certain class without a context are difficult to understand. In the specific, given that the aim is usually to rapidly adjust some machinery parameters to tackle production defects avoiding long stops of the working cells.

Given these premises, it is obvious that it would be optimal if any DL system integrated into an industrial context could provide the information that industry workers are accustomed to processing. For sure the efficiency estimation can be provided easily but other information that provides semantic and quantitative information like the exact position of a defect, the length of a certain component, etc. are hard to be directly coded into a machine learning algorithm especially when the samples for the training phase are limited as discussed in the text.

The approach proposed here is to set up a particular DL architecture that allows taking advantage of both state-of-the-art machine learning techniques and knowledge representation properties typical of classical computer vision approaches. The main idea is to let the user understand why a product has been classified as scrap with some measures.

To make it simple, in the following, the problem of quality inspection of daily contact lenses is discussed. In the case of contact lenses, the producer could be interested in several characteristics of the product to judge if the quality of the product is acceptable or if it should be discarded. Suppose that these characteristics are the circular shape, the measure of its radius, and to asses the presence of occlusion patterns over the central surface of the lens itself in order to avoid the selling of bad looking products. This kind of measurement and identification could be carried out with classical computer vision. However, given a large amount of variation of the lenses product, customizing the computer vision tasks for each possible combination of colors and patterns will be expensive. A DL approach could instead satisfy the industry needs generalizing the concept of a good quality product against a scrap product. In this case, unfortunately, the information about the measures of interest are lost in the process. The solution, proposed here, is to approach the problem still with an ML architecture but having as output an intermediate and simplified representation of the input data so that simple computer vision techniques could be applied without the need for customization for every variation of the product. This particular encoder-decoder architecture is a compromise that will allow the industrial people to understand and correctly assess the output provided by the DL software. Figure 6 shows an example of an architecture that given as input the image of a contact lens acquired by a camera, it produces a simplified representation of the lens as output. This special encoding allows measuring with ease the characteristics of the lens product in a way that is completely clear both to the industry partner and to the engineering one. Different variations of good contact lenses are converted to the same encoding by the network. The new encoding allows the easy computation of

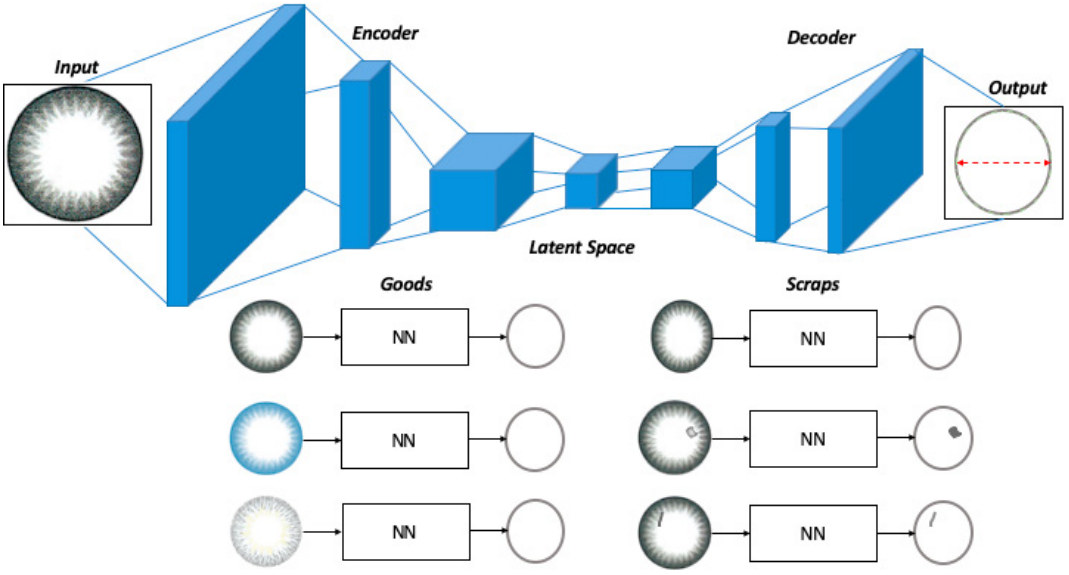

Fig. 6. Proposed encoding architecture applied on a contact lenses case. From a latent space representation, a simplified version of the input is generated in order to compute with classical computer vision techniques quantitative measurements and detect defects.

the measures of interest, and defects on the products are immediately visible and can be spotted with simple computer vision techniques.

5.1. Discussion

Small deviations on an acquired product image could result in misclassification in a typical classification network. For this special reason, a particular focus has been put in the last years on the study of adversarial inputs [40] that could drive unwanted results in the classification process. When a slight modification appears in the product surface under analysis, if this kind of alteration has not been considered during the training stage of the machine learning algorithm, the membership class probabilities could increase in the wrong class and decrease in the desired (correct) one. The result is that the system produces a class label that is not correct, and no clue is given on how this particular decision has been made. On the contrary, employing the proposed architecture, the presence of an alteration in the input should result in an alteration in the output image that is still interpretable by human operators and probably also by classical computer vision techniques measurements. This situation satisfies both the engineers that can still analyze the input and produce automated results and the industry personnel that can understand visually what led to the result presented by the inspection system.

The presented case is simple by design to give a clear and understandable example of the idea of the proposed methodology applied to a real industrial problem. Furthermore, the presented methodology is not application-specific and is designed to be generally applicable to other industrial common cases. The key idea behind the approach is to substitute a network that directly gives the decision as output with a network that instead processes the input to provide a simplified version of it in such a way that the user can reasonably (directly or by using classical computing) take the same decision.

6. Conclusions

This manuscript discusses problems faced by AI solution developers during the integration of such technologies in industrial manufacturing facilities. It starts analyzing the divergence in opinions between developers and users of the systems aiming at identifying the sources of skepticism in the adoption of Machine Learning Solutions. It continues presenting possible approaches to solve the problem of data imbalance that is crucial for correctly train a deep learning architecture, and a short introduction to explainable machine learning is provided to complete the discussion. An approach providing a sort of common language to both AI designers and Industry people is finally introduced. The approach presents a conceptual architecture that will facilitate understanding and improve the explainability of Deep Learning systems.

References

[1] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, April 1997.

[2] Lorenzo Peppoloni, Massimo Satler, Emanuel Luchetti, Carlo Alberto Avizzano, and Paolo Tripicchio. Stacked generalization for scene analysis and object recognition. In *IEEE 18th International Conference on Intelligent Engineering Systems INES 2014*, pages 215–220. IEEE, 2014.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C.

Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[4] Salvatore DAvella, Paolo Tripicchio, and Carlo Alberto Avizzano. A study on picking objects in cluttered environments: Exploiting depth features for a custom low-cost universal jamming gripper. *Robotics and Computer-Integrated Manufacturing*, 63:101888, 2020.

[5] P. Sassi, P. Tripicchio, and C. A. Avizzano. A smart monitoring system for automatic welding defect detection. *IEEE Transactions on Industrial Electronics*, 66(12):9641–9650, 2019.

[6] Bee Wah Yap, Khatijahhusna Abd Rani, Hezlin Aryani Abd Rahman, Simon Fong, Zuraida Khairudin, and Nik Nik Abdullah. An application of oversampling, undersampling, bagging and boosting in handling imbalanced datasets. In Tutut Herawan, Mustafa Mat Deris, and Jemal Abawajy, editors, *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, pages 13–22, Singapore, 2014. Springer Singapore.

[7] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. *ArXiv*, abs/1805.09501, 2018.

[8] Tian Wang, Yang Chen, Meina Qiao, and Hichem Snoussi. A fast and robust convolutional neural network-based defect detection model in product quality control. *The International Journal of Advanced Manufacturing Technology*, 94(9):3465–3471, Feb 2018.

[9] J. Chen, Z. Liu, H. Wang, A. Nez, and Z. Han. Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network. *IEEE Transactions on Instrumentation and Measurement*, 67(2):257–269, Feb 2018.

[10] Javier Jorge, Jesús Vieco, Roberto Paredes, Joan-Andreu Sánchez, and José-Miguel Benedí. Empirical evaluation of variational autoencoders for data augmentation. In *VISIGRAPP*, 2018.

[11] Biao Leng, Kai Yu, and Jingyan QIN. Data augmentation for unbalanced face recognition training sets. *Neurocomput.*, 235(C):10–14, April 2017.

[12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[13] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. cite arxiv:1411.1784.

[14] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *CoRR*, abs/1411.1792, 2014.

[15] S.Guan. Tl-gan: transparent latent-space gan. 2018. Available at https://github.com/SummitKwan/transparent_latent_gan.

[16] R. Ren, T. Hung, and K. C. Tan. A generic deep-learning-based approach for automated surface inspection. *IEEE Transactions on Cybernetics*, 48(3):929–940, March 2018.

[17] P. Tripicchio, G. Camacho-Gonzalez, and S. D'Avella. Welding defect detection: Coping with artifacts in the production line. *The International Journal of Advanced Manufacturing Technology*, 2020.

[18] Jeffrey P Mower. Prep-mt: predictive rna editor for plant mitochondrial genes. *BMC bioinformatics*, 6(1):96, 2005.

[19] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[20] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[21] Davide Castelvecchi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016.

[22] Zachary C Lipton. The mythos of model interpretability. *Queue*, 16(3):31–57, 2018.

[23] Leo Breiman et al. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.

[24] Alun Preece, Dan Harborne, Dave Braines, Richard Tomsett, and Supriyo Chakraborty. Stakeholders in explainable ai (2018). *arXiv preprint arXiv:1810.00184*, 122.

[25] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[26] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

[27] Andrej Karpathy, Justin Johnson, and Fei-Fei Li. Visualizing and understanding recurrent networks. *CoRR*, abs/1506.02078, 2015.

[28] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features, 2019. cite arxiv:1905.02175.

[29] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. *International Conference on Learning Representations*, 2019.

[30] J Sunil Rao and William JE Potts. Visualizing bagged decision trees. In *KDD*, pages 243–246, 1997.

[31] Fan Yin Tzeng and Kwan-Liu Ma. Opening the black box -data driven visualization of neural networks. In *VIS 05*, 12 2005. VIS 05: IEEE Visualization 2005, Proceedings ; Conference date: 23-10-2005 Through 28-10-2005.

[32] Aleks Jakulin, Martin Možina, Janez Demšar, Ivan Bratko, and Blaž Zupan. Nomograms for visualizing support vector machines. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 108–117, 2005.

[33] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[34] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[35] Daniel W Apley. Visualizing the effects of predictor variables in black box supervised learning models. *arXiv preprint arXiv:1612.08468*, 2016.

[36] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65, 2015.

[37] Xiaoyu Liu, Jie Chen, Joel Vaughan, Vijayan Nair, and Agus Sudjianto. Model interpretation: A unified derivative-based framework for nonparametric regression and supervised machine learning. *arXiv preprint arXiv:1808.07216*, 2018.

[38] Joel Vaughan, Agus Sudjianto, Erind Brahimi, Jie Chen, and Vijayan N. Nair. Explainable neural networks based on additive index models, 2018.

[39] Zebin Yang, Aijun Zhang, and Agus Sudjianto. Enhancing explainability of neural networks through architecture constraints, 2019.

[40] Nicolas Papernot, Patrick McDaniel, Ananthram Swami, and Richard Harang. Crafting adversarial input sequences for recurrent neural networks. In *MILCOM 2016-2016 IEEE Military Communications Conference*, pages 49–54. IEEE, 2016.