Deep learning for big data applications in CAD and PLM – Research review, opportunities and case study

Article *in* Computers in Industry · September 2018

DOI: 10.1016/j.compind.2018.04.005

CITATIONS
2

READS
421

6 authors, including:

Jonathan Dekhtiar
NVIDIA
3 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Alexandre Durupt
Université de Technologie de Compiègne
61 PUBLICATIONS **187** CITATIONS
SEE PROFILE

Matthieu Bricogne
Université de Technologie de Compiègne
59 PUBLICATIONS **216** CITATIONS
SEE PROFILE

Benoit Eynard
Université de Technologie de Compiègne
261 PUBLICATIONS **1,553** CITATIONS
SEE PROFILE

Computers in Industry 100 (2018) 227–243

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

Review

Deep learning for big data applications in CAD and PLM – Research review, opportunities and case study

Jonathan Dekhtiar[a], Alexandre Durupt[a,*], Matthieu Bricogne[a], Benoit Eynard[a], Harvey Rowson[b], Dimitris Kiritsis[c]

[a] Université de Technologie de Compiègne, Department of Mechanical Engineering, UMR UTC/CNRS 7337 Roberval, CS 60319, 60203 Compiègne Cedex, France
[b] DeltaCAD, 795 Rue des Longues Raies, 60610 La Croix-Saint-Ouen, France
[c] Swiss Federal Institute of Technology at Lausanne (EPFL), STI-IPR-LICP, CH-1015 Lausanne, Switzerland

ARTICLE INFO

Keywords:
Deep learning
Machine learning
Computer vision
Product Lifecycle Management
Digital mock-up
Shape retrieval

ABSTRACT

With the increasing amount of available data, computing power and network speed for a decreasing cost, the manufacturing industry is facing an unprecedented amount of data to process, understand and exploit. Phenomena such as Big Data, the Internet-of-Things, Closed-Loop Product Lifecycle Management, and the advances of Smart Factories tend to produce humanly unmanageable quantities of data.

The paper approaches the aforesaid context by assuming that any data processing automation is not only desirable but rather necessary in order to prevent prohibitive data analytics costs. This study focuses on highlighting the major specificities of engineering data and the data-processing difficulties which are inherent to data coming from the manufacturing industry. The artificial intelligence field of research is able to provide methods and tools to address some of the identified issues. A special attention was paid to provide a literature review of the most recent (in 2017) applications, that could present a high potential for the manufacturing industry, in the fields of machine learning and deep learning.

In order to illustrate the proposed work, a case study was conducted on the challenging research question of object recognition in heterogeneous formats (3D models, photos and videos) with deep learning techniques. The DICE project – DMU Imagery Comparison Engine – is presented and has been completely open-sourced in order to encourage reuse and improvements of the proposed case-study. This project also leads to the development of an open-source research dataset of 2000 CAD Models, called DMU-Net available at: https://www.dmu-net.org.

1. Introduction

With the advent of phenomena such as *Cloud Computing*, *Extended Enterprise*, or *Smart Products and Manufacturing*, the manufacturing industry is facing an unprecedented increase in accessible and available data [1]. New opportunities, usages and needs arise from this observation such as Knowledge Discovery, Process Automation, Automated or Intelligent Control, Decision Support Systems, Recommendation Engine, Key Performance Indicators (KPI) Forecasting.

However, due to massive data scale, these opportunities all require to be, at least partly, automated. Recurring issues such as *massive amounts of data, high data-dimensionality* [2], *heterogeneous data aspects, and low data-quality* [3] greatly reduce, not only data integration and consumption, but also automation possibilities for statistical analysis.

Recent technological advances in terms of network throughput, data availability, computation capabilities and storage capacities allow companies and researchers to design algorithms and self-learning systems achieving industrial-grade performances. These recent advances in machine learning and deep learning offer the manufacturing industry tools and techniques reaching, at this date, some of the best performances in the current state-of-the-art to tackle the aforesaid recurring challenges.

This paper aims to:

- explore the specificities of the *engineering data* and the major challenges one may encounter when applying machine learning or deep learning in the manufacturing industry.
- relate a literature review of the existing applications in the fields of machine learning and deep learning that could be adapted to the manufacturing industry to address engineering issues.
- release and open to the research community the dataset (and possible future recurring research challenge) DMU-Net in order to insure results repeatability and statistical robustness: https://www.dmu-net.org. This dataset has also been designed to facilitate

* Corresponding author.
E-mail address: alexandre.durupt@utc.fr (A. Durupt).

https://doi.org/10.1016/j.compind.2018.04.005
Received 8 September 2017; Received in revised form 2 March 2018; Accepted 10 April 2018

research collaboration between researchers in Computer Science and Manufacturing. It should also help to disseminate achieved results and allow a clear understanding of the state-of-the-art by providing an unique scale to compare methods and models.
- detail a method illustrated with a case-study addressed with a deep learning software product focusing imagery data processing (from 3D to 2D data), highly contextualised in engineering, able to outperform the existing state-of-the-art tools and techniques in the manufacturing industry. The DMU-Net dataset constitutes the training dataset of this study. In order to facilitate the understanding of the proposed approach and fits into the open-science guidelines, every aspects of this study has been released as open source software.

The paper is structured as followed: In Section 2, a focus on the manufacturing computing context will be firstly put on. This will serve as a starting point for the proposed analysis and highlight the reasons to consider *artificial intelligence* to automate some of the engineering processes. Then a prospective study on *the current state-of-the-art on artificial intelligence* systems applicable in the manufacturing industry will be presented in Section 3. Finally, the case-study of this paper and its characteristics will be introduced in Section 4 to highlight how *deep learning* techniques could help addressing objectives of manufacturers. The applied method, commonly used in deep learning and computer vision research, will detailed in Section 5, the source code associated with the proposed case-study has been released in open-source to ease the understand and accelerate the implementation of similar software products and solutions which share common objectives.

2. Engineering and computing – massive aspects and research challenges

2.1. Available data and computer resources have a major impact on the industry

Computers took, over the years, an essential and central place in almost every engineering tasks. This trend has led the manufacturing industry to face an ever-challenging situation regarding computing issues. Data storage capacity, network traffic and computation capabilities, are all facing an exponential increase which encourages the manufacturing industry to rethink its habits and workflows manipulating engineering data at a large scale. Table 1 summarises the aforesaid situation.

Even if the accuracy of the numbers mentioned in Table 1 is questionable, there is no doubt that the industry is facing unprecedented amounts of data and network traffic all across the Information Systems (IS). It becomes increasingly challenging to handle these data on a daily basis, in such large quantities, with human capacities.

In order to deal with the present situation, process automation appears as a promising solution. This study does not only consider *automation* to be an efficient way to reduce costs and improve efficiency. *Automation* is also appraised as the only possible way to address massive scale engineering data challenges.

Thanks to recent advances in computation capabilities, it is conceivable to perform, almost in real time, complex calculations at an affordable price. Currently, artificial intelligence (AI) algorithms take full advantage of the newly available computation resources to solve complex problems that were previously out of reach.

2.2. Engineering data are mostly unstructured and heterogeneous – a challenging situation

This study focuses on any data used by domain experts and will be later referred as *Engineering Data*. These data come in very different forms and originates from various systems. This fact introduces a high degree of heterogeneity which tends to make any data usage difficult. Efficient techniques exist for number of applications using similar data, nevertheless data heterogeneity significantly impairs the possibilities for human-designed automated processes due to the broad range of engineering activities.

Two of the most common sources of data heterogeneity and their repercussions have been reviewed in the upcoming subsection:

2.2.1. Data heterogeneity – a recurrent limiting factor for big data engineering applications

2.2.1.1. Data type heterogeneity. Data type hterogeneity relates to the different types of file that one may find inside the IS.

A few examples could be: *Sketches, Drawings, Laser Scans, CAD Files, Technical Reports, etc.*

It becomes a major bottleneck in a Product Lifecycle Management (PLM) perspective when engineers need to compare the contents of files of heterogeneous (i.e. different) types but concerning the same *PLM part*.

A research question could be: *How to, effectively and automatically, compare, hundreds of files and documents that relate to a given assembly when they cannot be compared by "just" analysing the inner differences or their structure?*

2.2.1.2. Data format heterogeneity. Data format heterogeneity is inherent to the manufacturing industry. In the extended enterprise context, it is most likely that the different actors do not use the same CAD software or version. These software products mostly use proprietary formats, and this constitutes a major limitation to the automated analysis objective. Needless to say, that the *format heterogeneity* issue is more frequently problematic with engineering-specific data types. Data types, such as photo, music or video, also have inner format heterogeneity (e.g. photos: JPEG, PNG, GIF, TIFF, etc.), however they only differ in terms of performance and quality. Converting a photo from one format to another will not change the number of people present in a photograph or their relative height. Contrarily, with CAD files, it is almost certain that any conversion or translation will degrade the inner content and delete some information that might be essential for the engineers [8].

A research question could be: *How to, effectively and automatically, compare, hundreds of CAD files, in different formats, presenting different versions of the same part and extract the differences between them?*

2.2.1.3. Could STEP and JT help to address the aforesaid

Table 1
Major computing trends and their respective evolution.

Trend category	Predictor	Measured trend	Variation rate by year	Variation over 10 years
Storage		Price per GB	−22% (÷2 every 33 months)	÷12
	Kryder's law [4]	Storage capacity	+27% (2× every 36 months)	11×
Network	Cisco forecast [5]	Global IP traffic	+60% (2× every 18 months)	110×
	Nielsen's law [6]	Network speed	+50% (2× every 21 months)	57×
Computation		Price per GFLOP/s	−37% (÷2 every 18 months)	÷100
	Moore's law [7]	Computing power	+60% (2× every 18 months)	110×

Fig. 1. Handwritten digits and characters. From left to right: "*3*", "*8*" and character "*B*".

challenges?. Neutral and opened formats such as STEP [9] and JT [10] help to tackle this issue. Nevertheless, they are used and specified as *exchange formats* and not as *daily working formats*. Thus, mostly used at specific times during the engineering workflows. Moreover, due to differences in the implementations, it may be questionable to heavily rely on these formats for analytic purposes (i.e. some pieces of information can, in fact, be missing).

Additionally to heterogeneous aspects, a *Big Data* context is inherently linked to *unstructured data*. Therefore, the following section will define the latter expression and cover the effects of having a large proportion of data that comes in unstructured formats.

2.3. Unstructured data are difficult to handle with human-made processes

Engineering data can be separated in mostly two groups, *structured and unstructured data*, that are defined below.

2.3.1. Structured data

Structured data are considered as any kind of data that can be stored in form of rows and columns in systems like databases or Excel™Spreadsheets. If some data can be stored in that manner, without losing any information, then the data can be qualified as a structured data.

Data such as computer logs, excel™files, databases, csv files, ERP *(Enterprise Resource Planning)* data are good examples of *structured data*.

2.3.2. Unstructured data

Unstructured data are any data that cannot be stored in a set of rows and columns **without losing inner information**. For instance, pixel values of a photograph, as its metadata, can be stored into a spreadsheet, however the information about the content of the image would be lost if it is not extensively described in the metadata.

Data such as emails, textual documents, videos, photographs, CAD models, 3D scans, reports are good examples of *unstructured data*.

Dealing with unstructured data is a lot more challenging in a data science perspective [11–13]. It requires highly complex, expensive and time-consuming *feature extraction* processes and operations (i.e. a feature represents a descriptor (e.g. horsepower of an engine) in a data science context).

It is estimated that the average IS roughly contain around 15% of **structured data** and 85% of **unstructured data** [14]. Such an assumption seems consistent with the actual status of the manufacturing industry. Furthermore, even if a more optimistic situation is considered, with a balanced rate of 50% structured and 50% unstructured data, it still appears critical to be able to *search, mine, explore, and exploit* these data. And thus, this leads to a paradoxical situation: vast amounts of data are available for exploitation, however the processing cost would be, nowadays, prohibitive.

This observation brings us to one of the major focuses of this study: *How to mine and explore unstructured engineering data at a massive scale (with a minimal amount of manual processing, thus reducing the data processing cost) ?* With the aforesaid challenges in mind, in the upcoming section, opportunities offered by self-learning systems will be explored.

3. Artificial intelligence and self learning systems

Since the end of 1930s, researchers and companies tried to automate tasks that are nearly effortless for humans (e.g. identifying simple objects in a given picture, understanding natural language, etc.). More recently, software products based on artificial intelligence (AI) algorithms, take advantage of parallel computation capabilities and large data quantities to approach human behaviours and understanding in complex situations.

3.1. From pattern recognition to deep learning techniques

3.1.1. Pattern recognition combined with machine learning requires domain-experts hand-crafting

"The field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as classifying the data into different categories." [15]

In order to illustrate this quote from C. Bishop, let us investigate the simple problem, yet not easy, of recognising handwritten digits and characters presented in Fig. 1, there should be no doubt that the human brain could make the distinction between the three images in an instant. However, it is a fairly difficult question for a computer. Especially, if a set of *expert-designed rules* is not given the computer.

A common approach to performing such a task (e.g. handwritten character recognition) is to design algorithms able to extract features from the images. These features are simpler to analyse than the original image, and should highlight as many differences as possible between the different characters and digits. Pattern recognition methods are usually combined with machine learning (ML) models, such as Support Vector Machines (SVM), for the inference part.

This complete workflow is, nowadays, widely used in the manufacturing industry and research. Table 2 presents a few of these use-cases:

Nevertheless, as highlighted before, pattern recognition techniques require a clear understanding of the situation by the domain experts. Experts which will induce high development costs and could delay the project because of their limited availability. Even when machine learning models are used to, at least partly, automate the decision process, using these methods is expensive and requires a complete system redesign when the situation slightly change. In the following section, we will detail how *deep learning* (DL) can help to overcome these challenges.

3.1.2. Deep learning challenges the status quo: reduced development costs with improved performances

"Deep learning is in the intersections among the research areas of neural networks, artificial intelligence, graphical modeling, optimization, pattern recognition, and signal processing." [36]

Deep learning (DL) can be considered as a sub-field of ML. DL has become one of the top performing state-of-the-art techniques and methods, which outperformed traditional ML techniques for numerous applications and scientific challenges such as computer vision and natural language processing.

Machine learning models triggered a *shift of intelligence*, from understanding the decision process to a domain-expert figuring out what are the key predictors for a given situation. Some DL models are eventually able to automate this expert-driven process given a large quantity of data and thus present a definitive advantage when it comes

Table 2
Current usages of machine learning in the manufacturing industry and its state of the art.

Visual inspection [16,17]	Geometric feature recognition [18]
Manufacturing scheduling [19]	Quality management [20–22]
Predictive maintenance [23–27]	Logistics forecasting [28–30]
Engineering decision support systems [31–35]	etc.

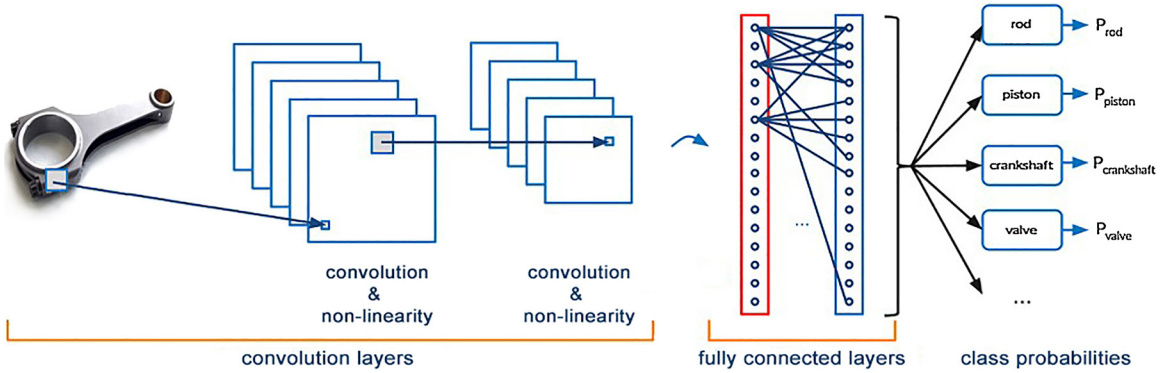


Fig. 2. Deep Convolutional Neural Network (CNN) performing an object recognition task. Inspired by [43].

to adaptability and cost reductions.

Fig. 2 presents the structure of a particular type of DEEP LEARning model, later applied in the case-study, primarily used to perform image processing: Convolutional Neural Networks (CNN). Since 2012 with the model *AlexNet* [37], CNN models outperform traditional Computer Vision techniques and methods in the most popular Visual Recognition Challenge (ImageNet) [38] and greatly helped to reach production-grade self-driving cars [39–42].

3.2. A prospective opportunity study for the manufacturing industry

Table 3 presents the results obtained after a prospective study aiming to explore the current possibilities offered by deep learning and machine learning models for the manufacturing industry. New achievements in terms of performance, precision or computation speed have been achieved since the early 2010s. Some of them could present real short-term interests for the manufacturing industry and require very little additional work to be adapted to industrial cases.

Table 3
A few usages of ML and DL, in the scientific literature, that could be adapted to manufacturing situations.

Activity recognition	Context awareness [44,45] Malicious behaviour detection [46]
Automated validation	Manufacturing defect detection [47,48] Middle-of-life automated control [49,50] (reality and virtual assets correspondence check) [a]
Computer vision	2D &3D imagery comparison [51] 3D scene understanding [52] Anomaly detection [47,48] Object classification, segmentation, identification [53–55] Pose estimation [56,57]
Data cleansing and filtering	Imagery data denoising (2D &3D) [58–60] Machine/human data automatic check [61] Outliers detection [62,63] Point cloud healing [64]
Generative engineering system	Engineering design generation [65] (manufacturing process generation) [a] Text generation [66]
Natural language processing	Automated summarisation [67] Automated translation [68,69] Information and concepts extraction [70]
Recommender systems	Best practices recommendation [71] Decision support systems [72–74] Knowledge discovery [71] Ranking systems and search engine [75–77]
Time series analysis	Anomaly detection and predictive maintenance [78–81] Demand and sales forecasting [82] Inventory and stock scheduling [82] Planning, dispatching and scheduling [83,84]

[a] Opportunities for which no academic work sufficiently adaptable or contextualized for engineering challenges was found.

This study should not, in any case, be considered as an exhaustive list of possibilities. In order not to give any hierarchical order or preference-rank, the following activities have been sorted in *alphabetical order*. Moreover, it can be stated that these activities are heavily linked and help each other. Therefore, it is normal and logical that different AI interests can lead to similar industrial objectives.

Considering the research review presented above, ML and DL algorithms appears as a well-known toolbox to solve complex engineering issues. However, recent advances in *deep learning* have led to a new horizon of possibilities and performances. Nonetheless, top performing algorithms, models and routines from the manufacturing research do not seem to take advantage of this new tool-set, to the best of our knowledge.

The upcoming section will detail, in an engineering case-study, how deep learning could be able to help manufacturers to challenge the *status quo* in terms of process automation and analytical processes. This will be a good opportunity to review the classical employed method in the data science domain and explore the different possibilities when applied to the manufacturing industry.

4. Deep learning for PLM applications – an engineering case-study

In order to emphasize the aforesaid possibilities and give a practical example, an engineering case study has been conducted with the objective to design a software product based on deep learning techniques that could support *Product Lifecycle Management* (PLM) in terms of activity and not in terms of an enterprise system.

This case-study aims to demonstrate that deep learning algorithms may help to address the previously detailed challenges:

a. To tackle the **data heterogeneity** issue in terms of **type** and **format**.
b. To be **massively scalable:**
   i. Not requiring a *complex, expensive or time-consuming* manual processing or expertise.
   ii. Being able to manage terabytes of data in a relatively short time with a reasonable cost.
c. Being as much **resistant to noisy data** as possible.
d. **Limited system-redesign costs**: if the situation changes slightly or previously unknown cases appear, the cost to adapt the system to the new situation should be kept as low as possible.

In that sense, the field of *Object Recognition* has been identified as a candidate field of research due to the difficulty to manipulate engineering imagery data and to develop autonomous systems addressing issues related to that topic. The following section will explore its major challenges and interests.

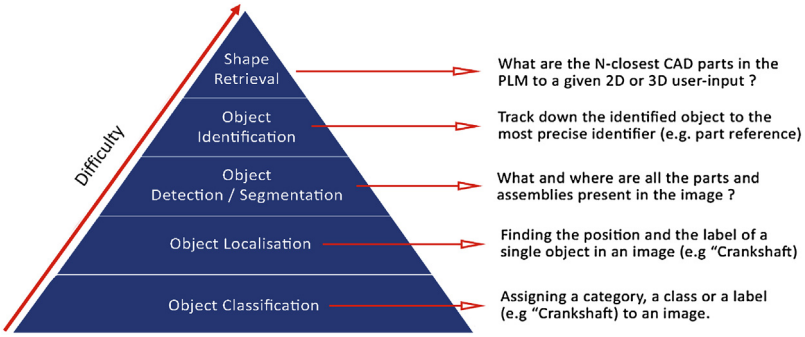


Fig. 3. Object recognition can be subdivided in five different tasks with an increasing difficulty.

4.1. Object recognition and associated research challenges

The challenge of object recognition can be summarised as follows:

"A vision system which makes use of an object model is referred to as a model-based vision system, and the general problem of identifying the desired object is referred to as object recognition. While there is no single definition of the object recognition problem, the objective is to identify a desired object in the scene and to determine its exact location and orientation." [85]

Fig. 3 presents five tasks in which the research area of *Object Recognition* can be sub-divided. Each of the following tasks aims to address slightly different objectives with an increasing difficulty yet essential to overcome in order to specify and build a system able to perform cross-comparisons between 3D and 2D data.

The challenges presented above have been well explored in the scientific literature with a manufacturing or engineering point of view [86–92]. However, none of them has been released in an open-source manner. A fact which tends to make difficult to conduct any benchmark study, if not impossible. One way to compare closed-source methods or applications would be to judge them on the same data and measure their effectiveness and accuracy with the same set of metrics. Nevertheless, they appear to use different data and metrics. This implies that it will not be possible to compare the results obtained with DICE, the case-study presented later, with the current state of the art in a manufacturing context. Nonetheless, a few deductions and assumptions can be made:

- The techniques and methods referenced above have been developed by domain experts and cannot be easily applied on new shapes or assemblies without requiring the expert to extensively re-analyse the situation. This fact tends to generate high costs to update a system based on these techniques if the use-case evolves or changes.
- With an optimistic estimate, we can assume that the methods referenced above could perform, at best, with an accuracy of 70–75%. Such a score would already be very high and would need to be validated. The best technique in 2011, seven years after the publications referenced above, was able to perform with an accuracy of 75% on the most famous computer vision challenge: ImageNet [38]. To the best of our knowledge, it seems realistic to assume that none of the above performs better than the top-performing method published several years later.
- Consequently, one of the objectives for the case-study later presented, is to perform more accurately than 75% on an object recognition challenge focused on industrial objects.

Moreover, some additional recurring issues in object recognition tasks tend to make difficult to design a robust system. These issues are presented in Fig. 4a–h. They constitute some of the major difficulties that one should overcome in order to develop a reliable 3D and 2D data comparison system in a manufacturing context.

Addressing similar objectives with the proposed approach, based on deep learning models, should allow fellow researchers to easily compare and assess the results obtained in this study with the current state-of-the-art in a manufacturing context. Nonetheless, very few studies currently focus on comparing 3D and 2D data together, the majority of them only consider 2D or 3D data.

Regarding the other challenges (e.g. scalability, and limited redesign cost), they will be addressed by the deep learning model set up to answer this issue.

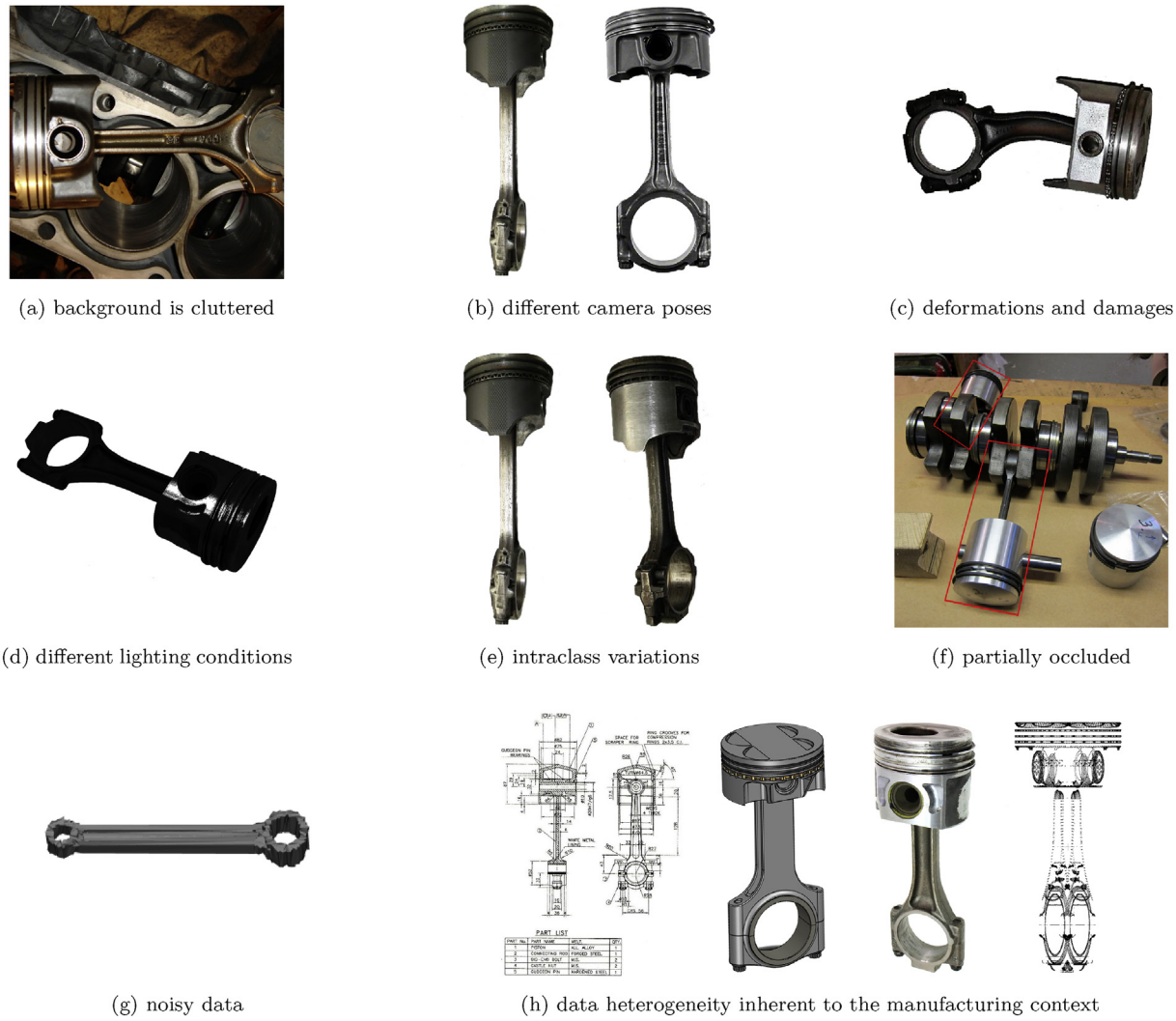
4.2. DMU-Net – a 3D CAD model dataset to enable research and insure statistical robustness

An essential part of the process is deciding, whether or not, it will be necessary to construct a dataset which will act as a knowledge-base for the deep learning model. If any dataset suiting our needs exists in the literature, it would be easier and a *good practice* to re-use it in order to facilitate results comparability.

In the upcoming sections, the essential assets of a standardised dataset will be reviewed. And thus highlighting the reasons which lead to the decision to develop a new dataset with a better fit to the considered needs: DMU-Net.

4.2.1. The reasons to use or develop a standardised dataset

a. **To enable research** by providing researchers a dataset fulfilling their needs on a specific topic.
   When it comes to using *data science* routines, algorithms or models, it is necessary to first collect a large amount of data in order to execute the *learning phase*. This phase can constitute a major bottleneck to science or engineering and may have prohibitive costs. Especially in the context of deep learning applications where the amount of input-data needs to be quite large.
b. **State-of-the-art reviewing and benchmarking**
   Being able to compare methods, algorithms and models is essential to provide the research communities and manufacturers clear insights on the best techniques to tackle a certain challenge. By making every researcher, on that specific scientific challenge, work with the same data, comparing becomes straight forward and periodic review studies and benchmarks can be conducted.
   Recurring challenges can be organized *around the dataset* during special events or conferences in order to promote the achieved work and stimulate contributions from various research groups.
c. **Insuring statistical robustness**
   Clear and unbiased metrics can be designed to judge published works and applications in open or closed-source. The amount and the variety of data provided greatly help to reduce skewness in the dataset researchers may use.
   Moreover, working with too little data may lead researchers to face *overfitting*, which would heavily reduce the generalisation power of the proposed methods (i.e. ability to take a correct decision on previously unseen data).



(a) background is cluttered (b) different camera poses (c) deformations and damages

(d) different lighting conditions (e) intraclass variations (f) partially occluded

(g) noisy data (h) data heterogeneity inherent to the manufacturing context

Fig. 4. The different situations and facts which harden object recognition tasks.

Very recently a paper has been published in Nature aiming to summarise best practices for reproducible science [93]. By building DMU-Net, a clear step toward their recommendations have been made.

4.2.2. Review of the existing 2D/3D computer vision datasets from the scientific literature

Coming from a long-time tradition in data science and computer vision fields, the **DMU-Net dataset** is released, and freely accessible for every researchers at the address https://www.dmu-net.org.

The decision to develop and release the DMU-Net dataset has been directly inspired by the datasets, and associated challenges, available for the research communities in the field of computer vision. Table 4 presents some of the most famous datasets used in computer vision research.

Table 4
Major computer vision datasets setting the ground basis of DMU-Net and directly inspiring it.

2D imagery dataset	3D imagery dataset
ImageNet [38]	ShapeNet [94]
PASCAL VOC [95]	PASCAL 3D+ [96]
MNIST [97]	AIM@SHAPE [98]
YouTube-8M [99]	Engineering Shape Benchmark (ESB) [100]

A more extensive review and benchmark of available datasets for 3D Computer Vision has been published in 2014 [101].

Aside from the Engineering Shape Benchmark (ESB) [100], none of them, consider complex and real mechanical parts coming from the manufacturing industry. Nonetheless, it appears essential to validate the proposed approach with *real data* from the manufacturing industry or very similar ones.

Concerning ESB, it presents most of the characteristics which have been previously introduced. Moreover, it is a dataset of 3D CAD Model, which perfectly fits the considered needs. However, most webpages (https://engineering.purdue.edu/PRECISE/shrec08) of the project are currently offline, and the last challenge happened in 2008. Moreover, the ESB dataset is too small for our needs, and even the available data are difficult to obtain and unmaintained.

It then appears necessary to build a new dataset, substantially larger than ESB to address the massive data aspects. In that perspective, the DMU-Net dataset was created and made available on the website https://www.dmu-net.org.

4.2.3. DMU-Net – principal characteristics

DMU-Net, in its current state, is composed of 30 classes of CAD files extracted from the open CAD Library GrabCAD: https://grabcad.com/. Common mechanical parts which present high inner-class variability were selected. The dataset is available in standard and opened formats: STEP-AP242, JT, STL, ThreeJS, X3DOM and Object File (OBJ + MTL)

Table 5
A listing of the 30 considered classes in the DMU-Net dataset.

Actuator → 37	AirFilter → 32	Battery → 46	Bearing → 128	Brake → 39
Camshaft → 46	Clutch → 61	Cooler → 30	Coupling → 43	Crankshaft → 81
CuttingTool → 41	ElectricMotor → 142	ElectronicBoard → 37	ExhaustPipe → 20	Gear → 18
GearWheel → 91	Helix → 60	Nut → 56	Piston → 87	Pulley → 109
Pump → 34	Rod → 53	RodPistonAssembly → 56	Screw → 219	ShockAbsorber → 78
SparkPlug → 87	Switch → 87	Turbo → 18	Washer → 47	WingNut → 33

Each part family or class is presented with the corresponding number of CAD models available for each category.
In total, DMU-Net is composed of 1916 CAD models distributed across 30 part families.

Fig. 5. DMU-Net dataset screenshot.

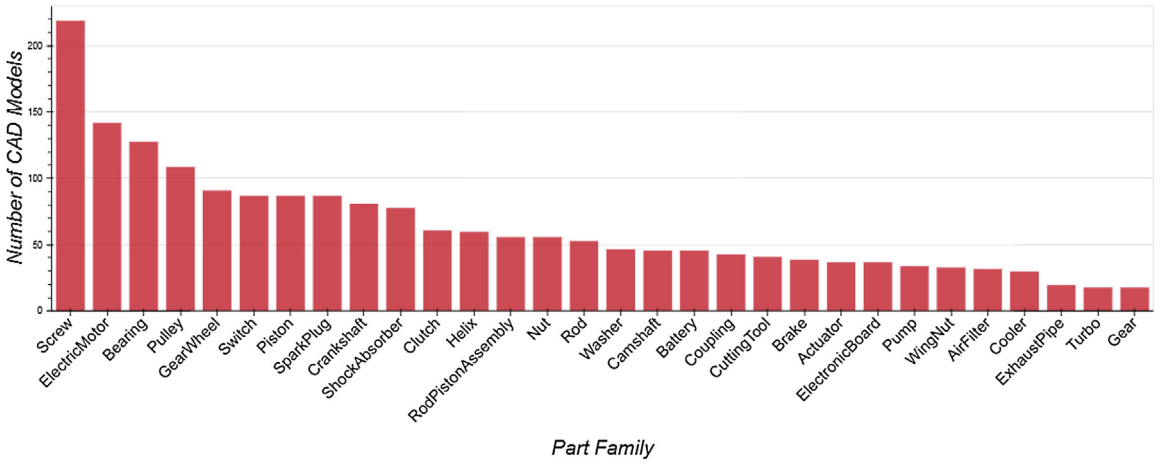


Fig. 6. DMU-Net dataset – the CAD models distribution has been intentionally skewed.

in order to keep a neutral position regarding to software vendors.

Table 5 presents the different part-families which compose the proposed dataset. The screenshot displayed in Fig. 5 supplements the latter by way of illustration.

In a similar manner to the existing diversity of CAD models stored in the PLM/PDM systems and their uneven distribution, the DMU-Net dataset was intentionally conceived presenting a skewed distribution. This situation, consistent with the reality of the manufacturing industry, tends to make the detection of the least represented classes notably more difficult.

It is common practice to split the dataset in two parts, each of them being used in a different phase of the model development. The first subpart is called the *training set*. It contains the data used to develop and train a machine learning model. The second subpart is called the *testing set*. It is used to measure the generalisation power of the model (i.e. the ability to take the correct decision on previously unseen data). This split is essential to prevent overfitting, which can be considered as *learning the noise inside the data*, and thus reducing the accuracy of the model in a production environment.

The distributions, presented in Fig. 6, describe the number of CAD models, available in the DMU-Net dataset, grouped by part-families. As stated before, the distribution of the 3D Models is unbalanced.

The following section will explore and detail the used method using deep learning models for the problem identified in the current section. The results obtained in this case-study and their reliability will also be described.

5. DICE – DMU imagery comparison engine

The whole project has been open-sourced and is available on the code sharing github.com platform. A demonstration page has been released in order to facilitate testing and reviewing without having to run

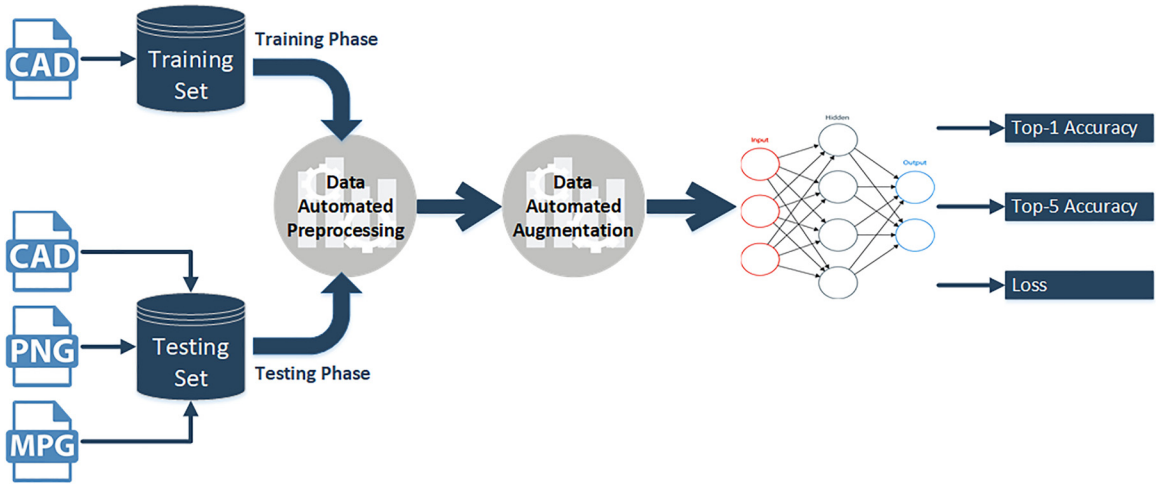


Fig. 7. The general workflow used for DICE.

the complex installation processes and try DICE in a couple of minutes.

- Source code and notebooks related to data preparation and model computation: Master Branch [1]
- Python WebAPI serving the computed model for the demonstration web page: Web API Branch [2]
- OpenCV Desktop Application running live with the camera input: RealTimeApp Branch [3]
- DICE Demonstration Webpage: Project Demonstration WebPage [4]
- Scanned 3D models used to test model resistance to noise: Master Branch-0 folder: data_3DScans [5]

The upcoming sections will describe each step of the general workflow applied in this case-study which is summarised in Fig. 7.

5.1. Data preprocessing: STEP models need to be converted to image data

Unfortunately, in the mid-2017, no deep learning model handling STEP models has been published in the current state of the art and developing one from scratch would take too much time for a case-study. Nevertheless, some models are able to handle STL geometries, however they do not adapt well to images. Nonetheless, images can be considered as a *degraded* view of a geometry with only one view point and no depth information. Thus, the type and format heterogeneity introduced by 3D models and images could be overcome by considering every CAD model as a series of images.

Fig. 8 presents the strategy used to *convert* or *translate* each CAD model into N different images.

Each rotation has been decomposed into a precise number of steps. Different configurations corresponding to various strength of preprocessing have been tested, each of them generates a defined number of automatically taken screenshots.

Rotation-2 and Rotation-1 will be referred, for the rest of this study, as R2 and R1. The formalism used to describe each level of precision is detailed below:

Example Level: [1, α, β, γ, 1] should be understood as followed:

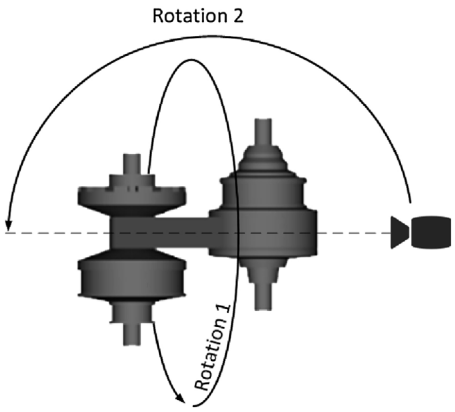


Fig. 8. Data augmentation process.

- The number of parameters in each level corresponds to the number of steps in which the R2 rotation will be decomposed. For instance, this example level takes $n_{rot2} = 5$ different parameters and thus R2 will be divided into $n_{rot2} - 1 = 4$ identical sub-rotations or steps equal to $180ř/(n_{rot2} - 1) = 180ř/4 = 45ř$. Performing $n_{rot2} - 1$ is necessary in order to *omit* the initial camera position in the counting process.
- During each step of the R2 rotation, a full R1 rotation is performed. R1 is also divided into a fixed number of steps, n_{rot1}. This number is defined by the value of the parameter corresponding to the current R2-step, and thus change over time. Each R1-step have an angle equal to: $360ř/(n_{rot1})$. In the example given above, at the first R2-step, the camera is located at the bottom of the model, R2 angle = 0ř, $n_{rot1} = 1$. At the second R2-step, R2 angle = 45ř, $n_{rot1} = \alpha$. At the third R2-step, R2 angle = 90ř, $n_{rot1} = \beta$. At the fourth R2-step, R2 angle = 135ř $n_{rot1} = \gamma$. And finally, the last R2-step, the camera is located at the top model, R2 angle = 180ř, $n_{rot1} = 1$. **Each time a rotation along R1 is performed, a screenshot is taken.**
- The first and last parameters correspond to camera positioned at the bottom of model, where R2 angle = 0ř, and at the top of the model, where R2 angle = 180ř. Having these two parameters fixed to 1 allows a screenshot to be taken from these important positions. It is unnecessary to set a value higher than 1 because it would rotate the scene and thus the additional pictures would be exact rotated copies of the first screenshot. It should be noticed that it is not necessary to define a systematic method to set the initial camera at the bottom of the model, indeed the model is fully rotated in every directions and screenshots are taken from every point of view, thus the actual

[1] Master Branch: https://github.com/DEKHTIARJonathan/DICE-DMU_Imagery_Classification_Engine/.
[2] Web API Branch: https://github.com/DEKHTIARJonathan/DICE-DMU_Imagery_Classification_Engine/tree/web-api.
[3] RealTimeApp Branch: https://github.com/DEKHTIARJonathan/DICE-DMU_Imagery_Classification_Engine/tree/RealTimeApp.
[4] Project Demonstration WebPage: https://dekhtiarjonathan.github.io/DICE-DMU_Imagery_Classification_Engine/.
[5] Master Branch – folder: data_3DScans: http://tinyurl.com/3DModelScanned.

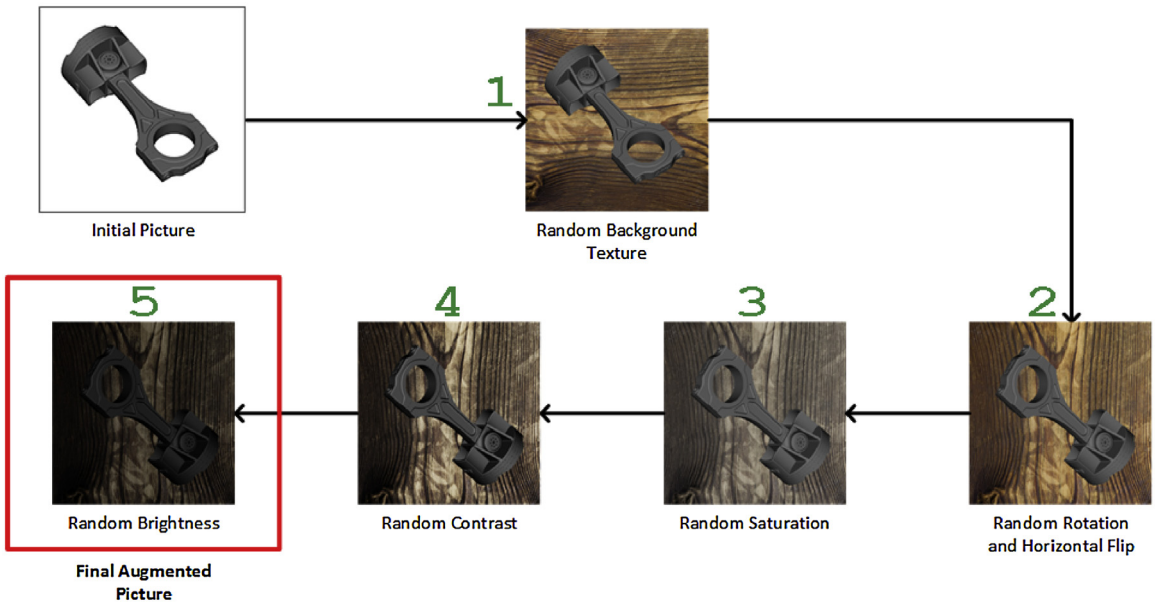


Fig. 9. Data augmentation process.

position of the "*bottom*" can be randomly chosen and fixed for each 3D model.

- Consequently, for the example given above, the following number of screenshots have been taken: $\sum_{i=1}^{5} v_i = 1 + \alpha + \beta + \gamma + 1$.

It should be noticed that because R1 is a 360ř rotation, it is redundant to perform a 360ř R2 rotation. Indeed, the screenshots obtained when the camera (R2) angle is between 0 and 180ř would be the same than between 180ř and 360ř if the orientation of the image is not taken into account.

In order to maximise the efficiency of the process, each defined preprocessing level should present these three intrinsic characteristics:

1. The **number of parameters should be an odd number**. Having this feature allows the workflow to take a series of screenshots with the camera having an R2 angle = 90ř, which is a desirable feature.
2. The **parameters values are symmetrical** from top-view to bottom-view. As the model needs to handle a wide range of CAD models, the workflow has been designed without prioritising any specific orientation. Thus giving the same results if the model is oriented bottom-up or bottom-down (if only a top and bottom can be arguably defined).
3. The **parameters values increase from bottom to middle viewpoint**. Due to an increasing R2 angle while the camera approaches the middle viewpoint (R2 angle = 90ř), R1 should be divided in smaller steps allowing to capture an increasing number of visible details.

With the formalism and rules described above, five different strength-levels are proposed and have been benchmarked:

1. **Minimal Strength**: [1, 4, 1] → 6 Screenshots per CAD Model.
2. **Low Strength**: [1, 3, 4, 3, 1] → 12 Screenshots per CAD Model.
3. **Medium Strength**: [1, 3, 6, 8, 6, 3, 1] → 28 Screenshots per CAD Model.
4. **High Strength**: [1, 3, 5, 6, 8, 6, 5, 3, 1] → 38 Screenshots per CAD Model.
5. **Maximal Strength**: [1, 3, 5, 6, 7, 8, 7, 6, 5, 3, 1] → 52 Screenshots per CAD Model.

5.1.1. *Data preprocessing – a trade off situation*

It seems natural that the more screenshots are taken the less information is lost in the data preparation process. However, this situation presents a significant trade off. With an increasing data preparation strength, an increasing number of view points will be generated and thus seen by the deep learning model during the training phase. However, the more screenshots to process, the longer the training phase will be. Moreover, the amount of additional information brought by a higher strength in preprocessing converges toward zero. This is due to the fact that close screenshots will start looking too similar and thus, a lot of redundancy will be introduced in the data which would not improve the final performances of the model and could only lead to longer computation times.

After testing each configuration, the level 4, corresponding to a "*High Strength*" preprocessing, has been chosen due to acceptable computation requirements and performances achieved by the final model.

5.2. *Data augmentation: random process to maximise model robustness to noise*

Once each STEP model, composing the DMU-Net dataset, has been reduced to 38 different images with the data preparation process, each image is then reprocessed in order to reduce the model sensitivity to specific conditions and allow a more consistent behaviour when applied to *industry situation*. Thus, the objective of this step is to randomly modify different settings of the image and create x (a parameter that needs to be determined and fixed, it will be further referenced as the "*augmentation factor*") new images, plus the original one, that present different conditions. The primary objective is to artificially force the model to differentiate background from the model, to understand that lighting conditions can vary in terms of brightness, saturation, contrast and finally that each part can come in different orientations.

Fig. 9 presents each of the five steps that are performed x times per image (x is the *augmentation factor*). Each step is randomly performed in order to mimic at best a plausible real situation.

1. **Background**: A background is randomly added. The textures have been selected from the images freely available on wildtextures.com. The objectives have been to add plausible backgrounds that could be found in a manufacturing context without being too restrictive.

Thus, the following background families have been considered: *Bark, Bricks, Glass, Leather, Metal, Paper, Rock, Stone, Textile, Wall and Wood* for a total of 142 different textures.
2. **Rotation and flip**: The image have a 50% chance to be horizontally flipped and $n_{rotation}$ is randomly picked between 0 and 3. Then the image is rotated $n_{rotation}$ times by 90ř.
3. **Saturation**: Contrarily to dull colors, it is quite rare to obtain over saturated colors with digital cameras nowadays. Thus, the saturation of each image is randomly modified from −50% (dull colors) to +20% (saturated colors). Consequently, the newly generated images are more frequently under-saturated.
4. **Contrast**: The image is randomly contrasted with a coefficient randomly selected from −50% to +50%.
5. **Brightness**: Finally, the brightness is also randomly modified with a range of −70% to +20% for the similar reasons to the saturation random modification process.

A final step of random image blurring/sharpening could have been added to the pipeline. However, the results obtained with DICE were good enough for a case study and did not require any further improvements. This data augmentation process is performed a fixed amount of times, per screenshot, and determined by the: *augmentation factor*. It is once again a trade-off between computation requirements, training time and final accuracy. Satisfying results have been obtained with an augmentation factor equal to 30.

To summarise, for each CAD Model, 38 screenshots are taken automatically from different viewpoints all around the models. Then each screenshot is processed by the data augmentation pipeline. And thus leading, for each CAD Model, to: $n_{screenshots} * (augmentation\_factor + 1) = 38 * (30 + 1) = 1178$ images (1 is added to the augmentation factor because the original screenshot is also kept unmodified in the process). All of them will be further used for the model training.

The following sections will now focus on the model selection and training process. Data will no longer be further modified.

5.3. Deep learning – model selection

First of all, using deep learning models might be a tedious and complex process. As part of a case-study, a particular attention was given not to reinvent the wheel and adapt existing and proven efficient models, from the state-of-the-art, for similar problems.

Three noteworthy deep learning models have been selected, they may not present the best results, nonetheless information about them can be easily found in the scientific literature. These models might constitute an easy first-step for anyone willing to invest time exploring deep learning and its possibilities for vision-based systems.

Table 6 presents the major characteristics of the considered models for this case-study. In order to clarify the reading, the *Input Size* columns should be understood as a 3-Dimensional Matrix which represents: Height*Width*Number_of_color_channels (i.e. 3 channels correspond an RGB picture, 1 channel for a Grayscale picture).

5.4. Deep learning – model tuning

In order to construct a deep learning model, a myriad of parameters (i.e. *weights* of the ML or DL model which are fit during the training) and hyperparameters (i.e. *higher-level* parameters which are determined by the data scientist before the training process) needs to be determined, tuned and set.

Table 7 presents a few noteworthy hyperparameters which are necessary to determine and correctly set in a deep learning model. They have been sorted in two categories: the ones on left relate to the structure of a deep learning model and those on right influence the behaviour of the model. Both of these categories are essential to obtain accurate and robust results.

Most of the time, data scientists make assumptions on certain parameters (e.g. the type of network used) or establish a range of values that a parameter can take. Thereafter, the model will be trained and cross-validated, across the possibilities considered, searching for the best combination of hyper-parameters. Some heuristics exist, in the literature, to facilitate or enable this process [105,106].

5.5. General training workflow and method applied in DICE case study

In this section, the *general workflow and method* used in this case-study will be reviewed. Moreover, a particular attention will be given to define the core concepts and specific vocabulary commonly used in the AI field.

5.5.1. Training and Testing Set:

As seen previously in Section 4.2.3, it is good practice to split the dataset in two parts: *training set* and *testing set*. It is necessary to determine how the proposed dataset will be split in two parts.

Due to a relatively limited number of CAD models currently available in the DMU-Net dataset (approximately 2000), it has been decided to perform an even split with a random selection process. Randomising the data repartition is essential in order to limit the possibilities of an inner bias, inside the *training set*, which will be later learned by the ML or DL model. With a bigger dataset, up to 80% of the data could have been included in the *training set*. Reducing this ratio down to 50% will help to detect overfitting when the amount of available data is limited. As previously explained in Section 4.2.3, overfitting can be considered as *learning the noise inside the data*, which highly reduce the effectiveness and accuracy of the model.

To summarise, for the proposed case-study, 50% of the data constitutes the *training set* and has been be used to train a DL model performing a classification task. The remaining 50% constitutes the *testing set* and has been used to evaluate the performance of the proposed approach with **previously unseen data** (i.e. capacity of generalisation) and thus insuring statistical robustness and results reproducibility.

5.5.2. Stratified K-Fold Cross validation

"Cross-validation is a statistical method of evaluating and comparing learning algorithms by dividing data into two segments: one used to learn or train a model and the other used to validate the model. In typical cross-validation, the training and validation sets must cross-

Table 6
The three deep learning models that were selected for this study.

Model name	Year	Dataset	Input size	Use of GPU	Number of layers
LeNet-5 [102]	1998	MNIST [97]	28Œ28Œ1(GrayScale)	No	7
AlexNet [37]	2012	ImageNet [38]	224Œ224Œ3 (RGB)	Yes	8 with ReLu activations
GoogLeNet [103]	2014	ImageNet [38]	224Œ224Œ3 (RGB)	Yes	22 with ReLu activations

In a broader point of view, current deep learning state-of-the-art suggests that *deeper models* provide more accurate results. For the sake of comparison, in 2015 for the ImageNet challenge [38], the record was held by ResNet-152 [104] with 152 layers and a top-5 error of 3.57% (when GoogLeNet achieved 6.7% with 22 layers in 2014 and AlexNet 16.4% with 8 layers in 2012). However, ResNet-152 will be put aside, for this study, due to its much more complex inner structure.

Table 7
Some of the key hyperparameters that need to be determined and set during the conception of a deep learning model.

Architecture related hyperparameters	Global hyperparameters
Type of Input	Loss Function (e.g. multiclass SVM loss, square hinge loss, etc.)
Network type (e.g. CNN, LSTM, RNN, etc.)	Learning rate (initial value and its variations)
Number of hidden layers	Type of regularisation (e.g. L1, L2, ElasticNet = L1 + L2, etc.)
Number of neurons by layer	Weight optimisation algorithm (e.g. Gradient Descent, etc.)
Type of layer (e.g. Convolution, Pooling, etc.)	Learning rate decay rate
How to initialise the network	Fitness metric used (e.g. precision, recall, etc.)
Activation function by neuron (e.g. Sigmoid, ReLu, etc.)	Number of *Epochs* (i.e. number of passes with the full dataset)
Dropout rate ∈[0 %, 100 %]	Conditions for early-stopping (e.g. model is not improving)

over in successive rounds such that each data point has a chance of being validated against. The basic form of cross-validation is K-Fold Cross-validation." [107]

One of the problem with K-Fold Cross validation, in a classification problem, is that some of the folds may present a very different class distribution from the original dataset. At the extreme, some folds may present only one class thus introducing bias in the metrics. Stratified K-Fold helps to address this issue by reproducing the original class distribution in the K-Folds randomly generated. Therefore, it has been decided to apply a Stratified K-Fold Cross validation process during the training process.

5.5.3. Deep learning model

The DL models considered in this study are the three Deep Convolutional Neural Networks (CNN) previously identified. In order to enable an isometric comparison of the considered models, two different metrics were recorded and their evolutions tracked: *Top-1 Accuracy, Top-5 Accuracy* defined below in Section 5.5.4. Each of the metrics have been measured during the cross-validation process (training phase) on the training set and also during the testing phase with the testing dataset.

The model tuning phase will not be explored in this publication in order to keep the focus on the results and possibilities for the manufacturing industry.

5.5.4. Top-X metrics

In machine learning and deep learning, it is common practice to use *Top-1 Accuracy* and *Top-5 Accuracy* metrics when performing a multi-class classification task. Having both of the metrics will allow us to know if the model is able to perform better if a small range of highly plausible outputs can be accepted by the end-user.

In particular in the engineering context, having only 5 possibilities to discriminate, would be, in many situations, beneficial and highly interesting. That also allows data scientist to know if the model is completely wrong or is able to give the correct answer in a relatively small batch called the *Top-5 predictions*.

5.6. Transfer learning – a method to train complex models with little amounts of data

The amount of available data still remains one of the major limiting factor that comes with any machine learning routine. Even if manufacturing companies usually have large quantities of data, cleaning and gathering these data could rapidly become an expensive process. Moreover, these data could be difficult to obtain for academic researchers due to intellectual property and confidentiality issues.

Therefore, it seems impracticable to gather many terabytes of data related to each specific engineering problem.

Nevertheless, some training techniques such as *Transfer Learning*, more extensively covered and described in the scientific literature [108–110], may help to circumvent the aforesaid issue. "Transfer learning is motivated by the fact that people can intelligently apply knowledge learned previously to solve new problems faster or with better solutions." [108]

Transfer Learning has been defined in the literature as follows: "Given a source domain D_S and learning task T_S, a target domain D_T and learning task T_T, transfer learning aims to help improve the learning of the target predictive function $f_T(.)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$, or $T_S \neq T_T$." [108]

An example, contextualised in the manufacturing industry, could be the following: Learning to differentiate "*Dogs from Cats*", for which large amounts of data are available for an inexpensive price, may greatly help for the objective to "*differentiate and classify mechanical assemblies*". This can be explained due to the fact that these different *knowledge fields* may share some low-level and fundamental knowledge such as *recognising canonical shapes* or *textures*. Only the reasoning part of the model, which is coming "*on-top*" the transferred knowledge (or layers in a neural network context), is "*re-trained*" during the training phase with the data related to the specific engineering task.

This approach has been proven efficient for many applications [109,110] and will be used to train the DICE application. In practice, DICE will use a deep learning model previously trained on one of the largest computer vision dataset: *ImageNet* [38]. The final and fully connected layers of the neural networks will be discarded, replaced and retrained in order to achieve the objectives of this study. Without *Transfer Learning*, DICE presents an uncontrolable *overfit* which greatly

Table 8
Summary of the results obtained during the DICE case-study.

		3-Stratified fold cross validation		Testing set	
Model name	Transfer learning	Top-1 accuracy	Top-5 accuracy	Top-1 accuracy	Top-5 accuracy
Random guess	Irrelevant	1/30 = 3.3%	5/30 = 16.67%	1/30 = 3.3%	5/30 = 16.67%
LeNet-5 [102]	No	14.66%	24.99%	7.4%	18.34%
AlexNet [37]	No	63.96%	92.05%	34.67%	61.15%
AlexNet [37]	Yes	58.73%	84.35%	52.50%	73.23%
GoogLeNet [103]	No	90.17%	97.51%	51.76%	67.59%
GoogLeNet [103]	Yes	84.35%	93.48%	82.81%	90.68%

The LeNet-5 [102] model has not been tested with a *transfer learning* approach due to the unavailability of a pre-trained model on the *ImageNet* [38] dataset. This is mainly due to its limited complexity which reduces the model effectiveness in complex situations.

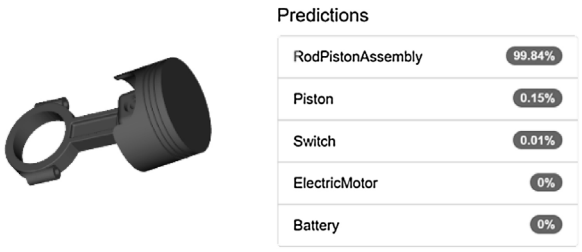


Fig. 10. Top-5 predictions for a Rod-Piston Assembly and their respective *confidence scores expressed in percentage.*

reduce the usability and usefulness of the model on *new data* as shown in Table 8 in following section.

5.7. Results

5.7.1. Result visualisation

Before going into the details of the results, Fig. 10 presents a possible predictions that can be expected from the model. The model is given an image of a mechanical assembly composed of a rod and a piston. The algorithm classifies the given picture and return the *TOP-5 predictions* with their respective score expressed as a probability percentage, a softmax function is applied to transform *prediction scores* given by the model, which does not sum up to a fixed number, to *class probabilities* which sums up to "1" over all the possible classes.

To be noted: The *class probabilities* do not sum up to 100% due to the fact that only the five most probable classes are presented and thus some of the possible classes are not displayed.

5.7.2. Case-study – results analysis

Table 8 presents the results obtained with each DL model with and without transfer learning being used.

The results obtained in this study are consistent with the one obtained with the dataset ImageNet [38,37,103]. It suggests a correct implementation of the aforesaid models and a smooth adaptability from generic research problems to specific challenges related to the manufacturing industry.

As expected, LeNet-5 model completely underperformed compared to the other models. This behaviour can be explained by the simplicity of its inner structure. Indeed, LeNet-5 is only capable of using grayscale images and a tiny grayscale image-size (28 pixels * 28 pixels * 1 channel). In comparison, LeNet-5 use around 200 times less pixels than GoogLeNet and AlexNet. Thus, LeNet-5 presents a limited capacity to infer on the data. Its classification performance is almost as bad as the random guess in the testing conditions. These structural limitations, which induce lower inference capabilities, can be explained by the limited computation resources and memory available in 1998, the year on which the LeNet-5 model was developed by Yann Lecun and his team [102]. Therefore, LeNet-5 will be put aside for the rest of the study.

Without transfer learning, GoogLeNet and AlexNet models suffer from a significant drop, by around 30%, in accuracy on testing conditions. Thus, training a deep learning model with the DMU-Net dataset from scratch tend to significantly reduce the generalisation power of the model due to the relatively limited amount of data available in the DMU-Net dataset (± 2000 CAD models). Therefore, it has been decided to apply a transfer learning process on the final model.

With transfer learning, they both perform very well with a Top-5 accuracy higher than 80% during the cross-validation process. However, GoogLeNet seems to outperform AlexNet when tested with similar conditions. This result is not surprising since GoogLeNet has been proven more efficient and accurate than AlexNet on the ImageNet dataset. Furthermore, when using Transfer Learning, it can be highlighted that GoogLeNet performs better, by around 30%, than AlexNet on the first guess. This feature is highly desirable in the objective of developing a decision support system which should perform at best for a very limited number of guesses, only one if possible.

For all the reasons stated earlier, GoogLeNet, trained with a transfer learning approach, will be the only model considered thereafter.

5.7.3. Noise resistance – artificially generated noise

In order to evaluate the impact of using noisy data, a situation which may occur in numerous industrial situations, a heavy load of random numerical noise has been generated, with the *Perlin Noise* algorithm [111], and added to a simple CAD model of a rod. It was observed that the trained GoogLeNet model was not impacted by a numerical noise and is still able to produce an accurate decision. Fig. 11a and b illustrates this behaviour by presenting virtually no impact on the decision performed.

The trained model performed decently with clean and noisy data even under a heavy load of numerical noise. The following section will focus on analysing the impact of noise captured during a 3D scanning process.

5.7.4. Noise resistance – 3D scan noise

In order to evaluate the impact of noise generated by 3D scanning, two STL models have been used to benchmark the model performance. These models were supplied by Artec Group inc. under the Creative Commons Attribution 3.0 Unported License which allows using, sharing and any modification under work attribution. The two models are available in the GitHub Repository: Master Branch – folder: data_3DScans [6]. The two scans were obtained using the *Artec Space Spider*™ scanner.

The obtained results have been displayed in Fig. 12a and b. They illustrate the behaviour of the model when fed with 3D Scans. The predictions performed are still accurate and the correct label is still present in the TOP-5 of predictions.

The trained model outputs the correct result in the first position for the screw model and in third position for the gear wheel. Thus, it can be concluded that the model is able to handle 3D scans with an acceptable degree of precision.

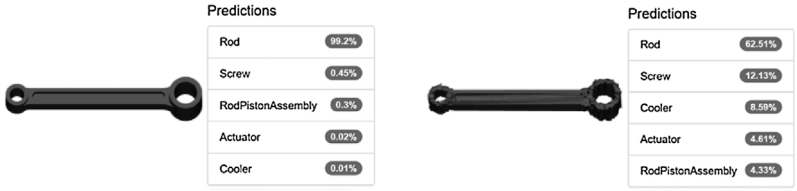
5.7.5. DICE – a proof of concept for real time industrial applications

Deep learning is a fairly computing intensive process during the training phase. However, once the model is trained, it can easily be run on cheap and non-powerful devices such as Raspberry Pis and tablet computers.

Once trained, the model has been implemented in a proof of concept software product which could be used by an operator in a factory. It uses the OpenCV Library [112] for the image acquisition part and the library Tensorflow [113] for the deep learning part. The image flow is captured through the rear camera of the tablet and processed in real time by the deep learning model locally (i.e. in a completely offline manner). The computed results are then displayed to the operator for further actions. Consequently, DICE can be used as a decision support tool for industrial visual applications.

The mechanical parts and assemblies used to test this proof-of-concept were collected from an old Renault® car during its dismantling. This situation tends to make the prediction harder by using slightly damaged or used mechanical parts. Nevertheless, relevant results were obtained in real-time. About 20 frames per second were analysed and continuously displayed back to the hypothetical user. This frame-rate is sufficient and fully compliant with the needs of an embedded industrial application. Indeed, it had been necessary to decrease the processing rate due to changes happening too rapidly and thus making the application hardly readable. For a human usage, it has been found that processing one image every half a second is a reasonable choice toward

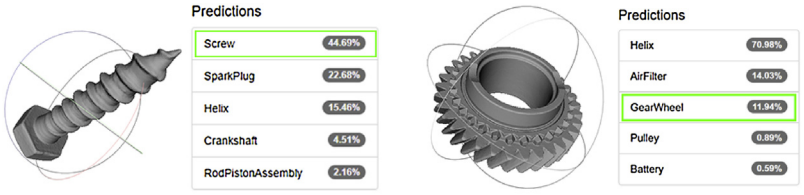
[6] Master Branch – folder: data_3DScans: http://tinyurl.com/3DModelScanned.



(a) Test performed with a clean and unmodified CAD model

(b) Test performed with a CAD model modified with random numerical noise

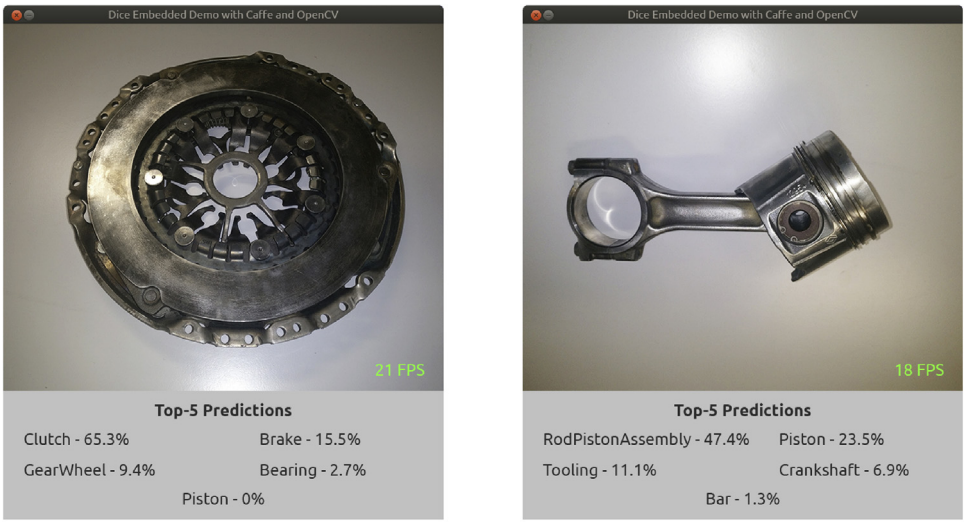
Fig. 11. Classification test performed to evaluate the model resistance to random numerical noise.



(a) Test performed with a 3D Scan of a Screw

(b) Test performed with a 3D Scan of a GearWheel

Fig. 12. Classification test performed to evaluate the model resistance to 3D scanning noise.



(a) DICE - Demo with a Clutch

(b) DICE - Demo with a Rod-Piston Assembly

Fig. 13. DICE – proof of concept application running on a tablet.

a smooth user experience. This reduction allow the application to only use 15% of the processing power available with the tablet used and thus improving battery life. In case of needing a faster processing, specific embedded devices can be used such as the NVIDIA® Jetson TK1, TX1 or TX2. They are easily available for an affordable price which depends on the version (from 400$ to 600$ in mid-2017).

Two screenshots of the DICE application are presented in Fig. 13a and b. The one on left presents the rear camera pointing at a clutch housing. Concerning the one on right, the camera is aiming at a mechanical assembly composed of a rod and a piston. The software product displays in real time the TOP-5 predictions given by the model and its refresh rate.

5.8. Case-study synthesis and discussion

The concepts which compose this method are sufficiently generic to be applied to a vast majority of problems that could be addressed with deep neural networks, especially when applied to imagery data. It might be necessary to adapt these concepts to fit at best the requirements of a similar yet not identical problematic, however the method presented could constitute an appropriate starting point.

The results obtained in this case study suggest that deep learning can be an efficient and powerful tool for the manufacturing industry. Despite the intensive training process, the use of deep learning in production should present minor difficulties due to the relatively limited computing resources required once the model is trained. Deep learning libraries, such as Caffe [114] and Tensorflow [113], have become available for Android™ mobile devices and for devices based on ARM (Advanced RISC Machine) processors such as the Raspberry Pi, thus making deep learning development for embedded applications everyday easier.

This study used one of the key and most interesting features of deep learning techniques: the ability to adapt to similar situations given new datasets large enough, thus limiting redeveloping costs when the core knowledge base evolve. In a more traditional manner, with expert systems for instance, a partial if not complete system redesign would be necessary. On the contrary with deep learning, only a model retrain, and perhaps some slight modifications on the model, would be necessary to fit to the novel, yet similar, industrial situation. This capacity will allow manufacturers to periodically re-adapt their models to the

most recent situation, with minor costs, and thus benefit from the most adapted tools to their situation at a given moment. Moreover, in contrast with classical machine learning where domain experts pre-processed the data by extracting important features (i.e. discriminative characteristics), deep learning models are able to extract, by themselves, key features from unstructured data. This fact is another key ability that could allow manufacturers to quickly apply deep learning models to a wide range of problems, especially where Expert System or classical machine learning techniques are inefficient due to a highly complex or fuzzy situation.

Even with the previous statements and observations, deep learning is a rapidly advancing field of research. Challenges that might seem not addressable today, might be covered in a few years. The research areas of Natural Language Processing, Imagery Understanding, Self-Driving cars, to mention just a few, have completely evolved and changed since 2012. They both now use, in 2017, the same language and techniques: Deep Neural Networks, even if their goals can be considered radically different. A lot of work still remain to be done to adapt the DICE proof of concept, developed in this study, to the manufacturing industry. It will be necessary to identify the specific challenges that need to be addressed in order to tackle the challenges of the industry 4.0. The DICE proof-of-concept is a simple case study and thus cannot take into account specific industrial situation.

For further works, DMU-Net needs to be extended to propose even more data to the academic researchers. International collaboration might be an efficient vector to enhance the dataset with different data sources that could be out of reach today. Regarding DICE, some difficulties, for the model, to classify real photographs compared to screenshots coming from a 3D model have been identified. This can be explained by the fact that non-realistic renderings were used during the training. It is a possible area of improvements of further research. New deep neural network architecture could also be explored, some are able to natively handle heterogeneous data, such as 3D data, photographs, sketches and drawing without having the need to prior convert them to a unique format would be valuable. Embedding Structures and Auto-Encoders might be a possible answer, with deep learning models, to this specific problem.

6. Conclusion and future works

As highlighted in Section 2, the manufacturing industry is facing many exponentially changing trends regarding network throughput, computation capabilities, data storage capacities and their respective costs for a fixed amount of data stored, exchanged, or processed. This study is based on the assumption that massive amounts of data will, in a near future, be handled by the manufacturing industry. They will be, by far, too large to be humanly manageable or will present prohibitive costs. Consequently, this study focused on solutions and tools allowing the development of decision support systems and thus reducing the amount of data to a few indicators or possibilities that an expert may use to produce a higher-level, or more context-aware, decision or operation.

In that sense, this study explored the different possibilities offered by the artificial intelligence field of research. A review of the opportunities and challenges that could be addressed by self learning systems have been presented and summarised in Table 3.

In order to compare the efficiency of a self-learning systems based on deep learning techniques compared to expert-designed systems, a benchmark dataset, DMU-Net, and a proof of concept, DICE, have been developed. The DMU-Net dataset will be enriched and improved over the coming years in order to facilitate results comparison and appeal researchers from the AI communities to work on challenges of the manufacturing industry. A possible partnership with part manufacturers is under study in order to help a faster grow in size of the dataset.

DICE addresses a modest, yet challenging, objective in term of *object recognition*. The results obtained with DICE are very promising and will require to be extended with large scale enterprise systems. The current version of the DICE project only focus on the first identified step, *object classification*, presented in Fig. 3. Thus DICE could be, in a near future, improved to consider and challenge the four remaining tasks (*localisation, segmentation, identification, retrieval*). The latter challenges constitute a much greater interest for the manufacturing industry.

In conclusion, deep learning models appear to be highly efficient to address problematics of the manufacturing industry. To pursue in that direction, possibilities offered by some deep learning models will be explored in future works. Some can be directly trained with 3D Data [115] and thus can present a real assets to address the previously identified needs and help to remove the data pre-processing necessary in the DICE approach. Embedding techniques could also be a way to address data heterogeneity and obtained an isometric representation of similar data.

To summarize, deep learning algorithms and routines may present many desirable advantages and assets to solve challenges inherent to the *industry of the future* or *industry 4.0*. Effective and versatile automation capabilities of deep learning combined with large scale processing may be an adequate answer to the previously identified problems of today industry: *data heterogeneity, massive scale, noisy data and unstructured data*. However, system interoperability still needs to be addressed in order to fully implement fully automated machine learning and deep learning workflows at large scale and deploy them seamlessly in the industry.

Acknowledgements

The authors would like to thank the French Region “Hauts-de-France” and the European Regional Development Fund (ERDF) 2014/2020 for the funding of this work.

References

[1] J. Davis, T. Edgar, J. Porter, J. Bernaden, M. Sarli, Smart manufacturing, manufacturing intelligence and demand-dynamic performance, Comput. Chem. Eng. 47 (2012) 145–156, http://dx.doi.org/10.1016/j.compchemeng.2012.06.037.
[2] D.L. Donoho, et al., High-dimensional data analysis: the curses and blessings of dimensionality, AMS Math Chall. Lect. 1 (2000) 32.
[3] C. Cortes, L.D. Jackel, W.P. Chiang, Limits on learning machine accuracy imposed by data quality, Proceedings of the First International Conference on Knowledge Discovery and Data Mining. KDD'95, AAAI Press, 1995, pp. 57–62 http://dl.acm.org/citation.cfm?id=3001335.3001345.
[4] C. Walter, Kryder's law, Sci. Am. 293 (2) (2005) 32–33.
[5] The Zettabyte Era – Trends and Analysis, (2016) http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html.
[6] J. Nielsen, Nielsen's Law of Internet Bandwidth, (1998) https://www.nngroup.com/articles/law-of-bandwidth/.
[7] G.E. Moore, Cramming more components onto integrated circuits, Electronics 38 (8) (1965) 114–117, http://dx.doi.org/10.1109/jproc.1998.658762.
[8] X. Xu, Integration based on step standards, Integrating Advanced Computer-Aided Design, Manufacturing, and Numerical Control: Principles and Implementations, IGI Global, 2009, pp. 246–265.
[9] ISO 10303-242:2014, Industrial Automation Systems and Integration – Product Data Representation and Exchange – Part 242: Application Protocol: Managed Model-based 3d Engineering, ISO Standard; International Organization for Standardization, Geneva, CH, 2014.
[10] ISO 14306:2012, Industrial Automation Systems and Integration – JT File Format Specification for 3d Visualization, ISO Standard; International Organization for Standardization, Geneva, CH, 2012.
[11] R. Blumberg, S. Atre, The problem with unstructured data, DM Rev. 13 (42-49) (2003) 62.
[12] S. Sagiroglu, D. Sinanc, Big data: A review, 2013 International Conference on Collaboration Technologies and Systems (CTS) (2013) 42–47, http://dx.doi.org/10.1109/CTS.2013.6567202.
[13] P. Buneman, S. Davidson, M. Fernandez, D. Suciu, Adding Structure to Unstructured Data, Springer Berlin Heidelberg, Berlin, Heidelberg, 1997, pp. 336–350, http://dx.doi.org/10.1007/3-540-62222-5_55 ISBN 978-3-540-49682-3.
[14] C.C. Shilakes, J. Tylman, Enterprise Information Portals, (1998) http://ikt.hia.no/perep/eip_ind.pdf.
[15] C.M. Bishop, Pattern Recognition and Machine Learning (Information Science and Statistics), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006 ISBN:0387310738.

[16] J.H. Yang, M.S. Yang, A control chart pattern recognition system using a statistical correlation coefficient method, Comput. Ind. Eng. 48 (2) (2005) 205–221, http://dx.doi.org/10.1016/j.cie.2005.01.008.
[17] H. Jia, Y.L. Murphey, J. Shi, T.S. Chang, An intelligent real-time vision system for surface defect detection, Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004, vol. 3 (2004) 239–242, http://dx.doi.org/10.1109/ICPR.2004.1334512.
[18] J.J. Shah, D. Anderson, Y.S. Kim, S. Joshi, A discourse on geometric feature recognition from CAD models, J. Comput. Inf. Sci. Eng. 1 (1) (2001) 41–51, http://dx.doi.org/10.1115/1.1345522.
[19] P. Priore, D. De La Fuente, A. Gomez, J. Puente, A review of machine learning in dynamic scheduling of flexible manufacturing systems, Artif. Intell. Eng. Des. Anal. Manuf. 15 (3) (2001) 251–263, http://dx.doi.org/10.1017/S0890060401153059.
[20] M. Noyel, P. Thomas, A. Thomas, P. Charpentier, Reconfiguration process for neuronal classification models: application to a quality monitoring problem, Comput. Ind. 83 (2016) 78–81 doi:https://doi.org/10.1016/j.compind.2016.09.004.
[21] R.E. Precup, P. Angelov, B.S.J. Costa, M. Sayed-Mouchaweh, An overview on fault diagnosis and nature-inspired optimal control of industrial process applications, Comput. Ind. 74 (2015) 75–94 doi:https://doi.org/10.1016/j.compind.2015.03.001.
[22] A. Srdoc, I. Bratko, A. Sluga, Machine learning applied to quality management – a study in ship repair domain, Comput. Ind. 58 (5) (2007) 464–473 doi:https://doi.org/10.1016/j.compind.2006.09.013.
[23] G.A. Susto, A. Schirru, S. Pampuri, S. McLoone, A. Beghi, Machine learning for predictive maintenance: a multiple classifier approach, IEEE Trans. Ind. Inf. 11 (3) (2015) 812–820, http://dx.doi.org/10.1109/TII.2014.2349359.
[24] P. Kankar, S.C. Sharma, S. Harsha, Fault diagnosis of ball bearings using machine learning methods, Expert Syst. Appl. 38 (3) (2011) 1876–1886, http://dx.doi.org/10.1016/j.eswa.2010.07.119.
[25] B. Samanta, Gear fault detection using artificial neural networks and support vector machines with genetic algorithms, Mech. Syst. Signal Process. 18 (3) (2004) 625–644.
[26] D.W. Ko Tae Joand Cho, Tool wear monitoring in diamond turning by fuzzy pattern recognition, J. Eng. Ind. 116 (2) (1994) 225–232, http://dx.doi.org/10.1115/1.2901934.
[27] P.G. Li, S.M. Wu, Monitoring drilling wear states by a fuzzy pattern recognition technique, J. Eng. Ind. 110 (3) (1988) 297–300, http://dx.doi.org/10.1115/1.3187884.
[28] R. Carbonneau, K. Laframboise, R. Vahidov, Application of machine learning techniques for supply chain demand forecasting, Eur. J. Oper. Res. 184 (3) (2008) 1140–1154, http://dx.doi.org/10.1016/j.ejor.2006.12.004.
[29] A.P. Ansuj, M. Camargo, R. Radharamanan, D. Petry, Sales forecasting using time series and neural networks, Comput. Ind. Eng. 31 (1) (1996) 421–424, http://dx.doi.org/10.1016/0360-8352(96)00166-0.
[30] Z. Tang, C. de Almeida, P.A. Fishwick, Time series forecasting using neural networks vs. box-Jenkins methodology, Simulation 57 (5) (1991) 303–310, http://dx.doi.org/10.1177/003754979105700508.
[31] F. Danglade, J.P. Pernot, P. Véron, L. Fine, A priori evaluation of simulation models preparation processes using artificial intelligence techniques, Comput. Ind. 91 (2017) 45–61 doi:https://doi.org/10.1016/j.compind.2017.06.001.
[32] D.J. Power, R. Sharda, Model-driven decision support systems: concepts and research directions, Decis. Support Syst. 43 (3) (2007) 1044–1061, http://dx.doi.org/10.1016/j.dss.2005.05.030.
[33] B. Filipic, M. Junkar, Using inductive machine learning to support decision making in machining processes, Comput. Ind. 43 (1) (2000) 31–41 doi:https://doi.org/10.1016/S0166-3615(00)00056-7.
[34] A. Sluga, M. Jermol, D. Zupanic, D. Mladenic, Machine learning approach to machinability analysis, Comput. Ind. 37 (3) (1998) 185–196 doi:https://doi.org/10.1016/S0166-3615(98)00098-0.
[35] M.J. Shaw, P.L. Tu, P. De, Applying machine learning to model management in decision support systems, Decis. Support Syst. 4 (3) (1988) 285–305, http://dx.doi.org/10.1016/0167-9236(88)90017-6.
[36] L. Deng, D. Yu, Deep learning: methods and applications, Found. Trends Signal Process. 7 (3-4) (2014) 197–387, http://dx.doi.org/10.1561/2000000039.
[37] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: P.L. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3–6, 2012, Lake Tahoe, Nevada, United States, 2012, pp. 1106–1114 http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.
[38] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, 2009 IEEE Conference on Computer Vision and Pattern Recognition (2009) 248–255, http://dx.doi.org/10.1109/CVPR.2009.5206848.
[39] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, F. Mujica, A. Coates, A.Y. Ng, An empirical evaluation of deep learning on highway driving, CoRR (2015), http://arxiv.org/abs/1504.01716.
[40] D. Levi, N. Garnett, E. Fetaya, Stixelnet: A deep convolutional network for obstacle detection and road segmentation, in: X. Xie, M.W. Jones, G.K.L. Tam (Eds.), Proceedings of the British Machine Vision Conference (BMVC), BMVA Press, 2015, , http://dx.doi.org/10.5244/C.29.109 ISBN:1-901725-53-7; 109.1–109.12.
[41] A. Lavin, S. Gray, Fast algorithms for convolutional neural networks, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 4013–4021, http://dx.doi.org/10.1109/CVPR.2016.435.
[42] M. Bojarski, D.D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L.D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, K. Zieba, End to end learning for self-driving cars, CoRR (2016), http://arxiv.org/abs/1604.07316.
[43] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M.S. Lew, Deep learning for visual understanding: a review, Neurocomputing 187 (2016) 27–48, http://dx.doi.org/10.1016/j.neucom.2015.09.116.
[44] Y. Liu, L. Nie, L. Han, L. Zhang, D.S. Rosenblum, Action2activity: recognizing complex activities from sensor data, CoRR (2016), http://arxiv.org/abs/1611.01872.
[45] G. Adomavicius, A. Tuzhilin, Context-Aware Recommender Systems, Springer US, Boston, MA, 2015, pp. 191–226, http://dx.doi.org/10.1007/978-1-4899-7637-6_6 ISBN:978-1-4899-7637-6.
[46] N. Ye, Y. Zhang, C.M. Borror, Robustness of the Markov-chain model for cyber-attack detection, IEEE Trans. Reliab. 53 (1) (2004) 116–123, http://dx.doi.org/10.1109/TR.2004.823851.
[47] T. Schlegl, P. Seeböck, S.M. Waldstein, U. Schmidt-Erfurth, G. Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, CoRR (2017), http://arxiv.org/abs/1703.05921.
[48] D. Carrera, G. Boracchi, A. Foi, B. Wohlberg, Detecting anomalous structures by convolutional sparse models, 2015 International Joint Conference on Neural Networks (IJCNN) (2015) 1–8, http://dx.doi.org/10.1109/IJCNN.2015.7280790.
[49] K. Katevas, I. Leontiadis, M. Pielot, J. Serrà, Practical processing of mobile sensor data for continual deep learning predictions, CoRR (2017), http://arxiv.org/abs/1705.06224.
[50] S.Z. Yong, L. Gao, N. Ozay, Weak Adaptive Submodularity and Group-based Active Diagnosis with Applications to State Estimation with Persistent Sensor Faults, (2017) http://arxiv.org/abs/1701.06731.arXiv:arXiv:1701.06731.
[51] X. Wang, V. Ly, R. Guo, C. Kambhamettu, 2d-3d face recognition via restricted boltzmann machines. 2014 International Conference on Computer Vision Theory and Applications (VISAPP), vol. 2 (2014) 574–580.
[52] Y. Zhu, R. Mottaghi, E. Kolve, J.J. Lim, A. Gupta, L. Fei-Fei, A. Farhadi, Target-driven visual navigation in indoor scenes using deep reinforcement learning, CoRR (2016), http://arxiv.org/abs/1609.05143.
[53] Y. Wang, W. Deng, Generative model with coordinate metric learning for object recognition based on 3d models, CoRR (2017), http://arxiv.org/abs/1705.08590.
[54] D. Pathak, R. Girshick, P. Dollár, T. Darrell, B. Hariharan, Learning Features by Watching Objects Move, (2016) arXiv preprint arXiv:161206370.
[55] S. Ren, K. He, R.B. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, CoRR (2015), http://arxiv.org/abs/1506.01497.
[56] P. Wohlhart, V. Lepetit, Learning descriptors for object recognition and 3d pose estimation, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 3109–3118, http://dx.doi.org/10.1109/CVPR.2015.7298930.
[57] A. Toshev, C. Szegedy, Deeppose: human pose estimation via deep neural networks. 2014 IEEE Conference on Computer Vision and Pattern Recognition (2014) 1653–1660, http://dx.doi.org/10.1109/CVPR.2014.214.
[58] J. Xie, L. Xu, E. Chen, Image denoising and inpainting with deep neural networks, in: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 341–349 http://papers.nips.cc/paper/4686-image-denoising-and-inpainting-with-deep-neural-networks.pdf.
[59] Y. Tang, R. Salakhutdinov, G. Hinton, Robust boltzmann machines for recognition and denoising, 2012 IEEE Conference on Computer Vision and Pattern Recognition (2012) 2264–2271, http://dx.doi.org/10.1109/CVPR.2012.6247936.
[60] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.A. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (2010) 3371–3408 http://dl.acm.org/citation.cfm?id=1756006.1953039.
[61] J.M. Hellerstein, Quantitative Data Cleaning for Large Databases, (2008) http://db.cs.berkeley.edu/jmh/papers/cleaning-unece.pdf.
[62] A. Zimek, E. Schubert, H.P. Kriegel, A survey on unsupervised outlier detection in high-dimensional numerical data, Stat. Anal. Data Min. 5 (2012) 363–387, http://dx.doi.org/10.1002/sam.11161.
[63] Z. Ferdousi, A. Maeda, Unsupervised outlier detection in time series data, 22nd International Conference on Data Engineering Workshops (ICDEW'06) (2006), http://dx.doi.org/10.1109/ICDEW.2006.157 x121-x121.
[64] C. Shang, A. Palmer, J. Sun, K.S. Chen, J. Lu, J. Bi, Vigan: Missing View Imputation with Generative Adversarial Networks, (2017) https://arxiv.org/abs/1708.06724.arXiv:arXiv:1708.06724.
[65] J. Wu, C. Zhang, T. Xue, W.T. Freeman, J.B. Tenenbaum, Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling, CoRR (2016), http://arxiv.org/abs/1610.07584.
[66] A. Graves, Generating sequences with recurrent neural networks, CoRR (2013), http://arxiv.org/abs/1308.0850.
[67] R. Nallapati, B. Xiang, B. Zhou, Sequence-to-sequence rnns for text summarization, CoRR (2016), http://arxiv.org/abs/1602.06023.
[68] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, CoRR (2014), http://arxiv.org/abs/1409.0473.
[69] K. Cho, van Merrienboer, B. b Çaglar Gülçehre, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, CoRR (2014), http://arxiv.org/abs/1406.1078.
[70] A. Bordes, X. Glorot, J. Weston, Y. Bengio, Joint learning of words and meaning representations for open-text semantic parsing, Proceedings of 15th International Conference on Artificial Intelligence and Statistics (2012), http://www.jmlr.org/

proceedings/papers/v22/bordes12/bordes12.pdf.
[71] D. Delen, H. Zaim, C. Kuzey, S. Zaim, A comparative analysis of machine learning systems for measuring the impact of knowledge management practices, Decis. Support Syst. 54 (2) (2013) 1150–1160, http://dx.doi.org/10.1016/j.dss.2012.10.040.
[72] S.G. Vadlamudi, T. Chakraborti, Y. Zhang, S. Kambhampati, Proactive decision support using automated planning, CoRR (2016), http://arxiv.org/abs/1606.07841.
[73] Y. Yang, P.A. Fasching, M. Wallwiener, T.N. Fehm, S.Y. Brucker, V. Tresp, Predictive clinical decision support system with RNN encoding and tensor decoding, CoRR (2016), http://arxiv.org/abs/1612.00611.
[74] A. Mansoul, B. Atmani, S. Benbelkacem, A hybrid decision support system: application on healthcare, CoRR (2013), http://arxiv.org/abs/1311.4086.
[75] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, Y. Wu, Learning fine-grained image similarity with deep ranking, CoRR (2014), http://arxiv.org/abs/1404.4661.
[76] P.S. Huang, X. He, J. Gao, L. Deng, A. Acero, L. Heck, Learning deep structured semantic models for web search using clickthrough data, Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management. CIKM '13, ACM, New York, NY, USA, 2013, pp. 2333–2338, http://dx.doi.org/10.1145/2505515.2505665 ISBN:978-1-4503-2263-8.
[77] M. Richardson, A. Prakash, E. Brill, Beyond pagerank: Machine learning for static ranking, Proceedings of the 15th International Conference on World Wide Web. WWW '06, ACM, New York, NY, USA, 2006, pp. 707–715, http://dx.doi.org/10.1145/1135777.1135881 ISBN:1-59593-323-9.
[78] R. Chalapathy, A.K. Menon, S. Chawla, Robust, deep and inductive anomaly detection, CoRR (2017), http://arxiv.org/abs/1704.06743.
[79] D.T. Shipmon, J.M. Gurevitch, P.M. Piselli, S.T. Edwards, Time Series Anomaly Detection; Detection of Anomalous Drops with Limited Features and Sparse Examples in Noisy Highly Periodic Data, (2017) https://arxiv.org/abs/1708.03665.arXiv:arXiv:1708.03665.
[80] D. Hsu, Anomaly Detection on Graph Time Series, (2017) https://arxiv.org/abs/1708.02975.arXiv:arXiv:1708.02975.
[81] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, LSTM-based encoder-decoder for multi-sensor anomaly detection, CoRR (2016), http://arxiv.org/abs/1607.00148.
[82] F.M. Bianchi, E. Maiorino, M.C. Kampffmeyer, A. Rizzi, R. Jenssen, An overview and comparative analysis of recurrent neural networks for short term load forecasting, CoRR (2017), http://arxiv.org/abs/1705.04378.
[83] Y. Zhu, D. Gordon, E. Kolve, D. Fox, L. Fei-Fei, A. Gupta, R. Mottaghi, A. Farhadi, Visual semantic planning using deep successor representations, CoRR (2017), http://arxiv.org/abs/1705.08080.
[84] E. Groshev, A. Tamar, S. Srivastava, P. Abbeel, Learning Generalized Reactive Policies Using Deep Neural Networks, (2017) https://arxiv.org/abs/1708.07280.arXiv:arXiv:1708.07280.
[85] F. Arman, J.K. Aggarwal, Model-based object recognition in dense-range images – a review, ACM Comput. Surv. 25 (1) (1993) 5–43, http://dx.doi.org/10.1145/151254.151255.
[86] J.W.H. Tangelder, R.C. Veltkamp, A survey of content based 3d shape retrieval methods, Proceedings Shape Modeling Applications 2004 (2004) 145–156, http://dx.doi.org/10.1109/SMI.2004.1314502.
[87] D. Zhang, G. Lu, Review of shape representation and description techniques, Pattern Recogn. 37 (1) (2004) 1–19, http://dx.doi.org/10.1016/j.patcog.2003.07.008.
[88] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, K. Ramani, Three-dimensional shape searching: state-of-the-art review and future trends, Comput.-Aided Des. 37 (5) (2005) 509–530, http://dx.doi.org/10.1016/j.cad.2004.07.002.
[89] D. Bespalov, W.C. Regli, A. Shokoufandeh, Local feature extraction and matching partial objects, Comput.-Aided Des. 38 (9) (2006) 1020–1037, http://dx.doi.org/10.1016/j.cad.2006.07.005.
[90] D.E. Clark, J.R. Corney, F. Mill, H.J. Rea, A. Sherlock, N.K. Taylor, Benchmarking shape signatures against human perceptions of geometric similarity, Comput.-Aided Des. 38 (9) (2006) 1038–1051, http://dx.doi.org/10.1016/j.cad.2006.05.003.
[91] W. Gao, S. Gao, Y. Liu, J. Bai, B. Hu, Multiresolutional similarity assessment and retrieval of solid models based on DBMS, Comput.-Aided Des. 38 (9) (2006) 985–1001, http://dx.doi.org/10.1016/j.cad.2006.06.004.
[92] T. Hong, K. Lee, S. Kim, Similarity comparison of mechanical parts to reuse existing designs, Comput.-Aided Des. 38 (9) (2006) 973–984, http://dx.doi.org/10.1016/j.cad.2006.05.004.
[93] M.R. Munafò, B.A. Nosek, D.V.M. Bishop, K.S. Button, C.D. Chambers, N. Percie du Sert, U. Simonsohn, E.J. Wagenmakers, J.J. Ware, J.P.A. Ioannidis, A manifesto for reproducible science, Nat. Hum. Behav. 1 (2017), http://dx.doi.org/10.1038/s41562-016-0021 0021+.
[94] A.X. Chang, T.A. Funkhouser, L.J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, F. Yu, Shapenet: an information-rich 3d model repository, CoRR (2015), http://arxiv.org/abs/1512.03012.
[95] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338, http://dx.doi.org/10.1007/s11263-009-0275-4.
[96] Y. Xiang, R. Mottaghi, S. Savarese, Beyond pascal: a benchmark for 3d object detection in the wild, IEEE Winter Conference on Applications of Computer Vision (2014) 75–82, http://dx.doi.org/10.1109/WACV.2014.6836101.
[97] Y. LeCun, C. Cortes, MNIST Handwritten Digit Database, (2010) http://yann.lecun.com/exdb/mnist/.
[98] B. Falcidieno, M. Spagnuolo, P. Alliez, E. Quak, C. Houstis, E. Vavalis, Towards the semantics of digital shapes: the aim@shape approach, Knowledge-Based Media Analysis for Self-Adaptive and Agile Multi-Media, Proceedings of the European Workshop for the Integration of Knowledge, Semantics and Digital Media Technology, EWIMT 2004, November 25–26, 2004, QMUL, London, UK, 2004 ISBN:0-902-23810-8; http://dblp.uni-trier.de/db/conf/ewimt/ewimt2004.html#FalcidienoSAQVH04.
[99] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, S. Vijayanarasimhan, Youtube-8m: a large-scale video classification benchmark, CoRR (2016), http://arxiv.org/abs/1609.08675.
[100] S. Jayanti, Y. Kalyanaraman, N. Iyer, K. Ramani, Developing an engineering shape benchmark for CAD models, Comput.-Aided Des. 38 (9) (2006) 939–953, http://dx.doi.org/10.1016/j.cad.2006.06.007.
[101] Y. Guo, J. Zhang, M. Lu, J. Wan, Y. Ma, Benchmark datasets for 3d computer vision, 2014 9th IEEE Conference on Industrial Electronics and Applications (2014) 1846–1851, http://dx.doi.org/10.1109/ICIEA.2014.6931468.
[102] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324, http://dx.doi.org/10.1109/5.726791.
[103] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S.E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, CoRR (2014), http://arxiv.org/abs/1409.4842.
[104] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, CoRR (2015), http://arxiv.org/abs/1512.03385.
[105] D. Erhan, Y. Bengio, A. Courville, P.A. Manzagol, P. Vincent, S. Bengio, Why does unsupervised pre-training help deep learning? J. Mach. Learn. Res. 11 (2010) 625–660 http://dl.acm.org/citation.cfm?id=1756006.1756025.
[106] Q.V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, A.Y. Ng, On optimization methods for deep learning, Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, 2011, pp. 265–272 June 28–July 2, 2011.
[107] P. Refaeilzadeh, L. Tang, H. Liu, Cross-Validation, Springer US, Boston, MA, 2011, pp. 532–536, http://dx.doi.org/10.1007/978-0-387-39940-9_565 ISBN:978-0-387-39940-9.
[108] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359, http://dx.doi.org/10.1109/TKDE.2009.191.
[109] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks? in: Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014, pp. 3320–3328 http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf.
[110] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. CVPR '14 (2014) 1717–1724, http://dx.doi.org/10.1109/CVPR.2014.222 ISBN:978-1-4799-5118-5.
[111] K. Perlin, F. Neyret, Flow noise, 28th International Conference on Computer Graphics and Interactive Techniques; vol. Technical Sketches and Applications, SIGGRAPH, 2001, p. 187 https://hal.inria.fr/inria-00537499.
[112] G. Bradski, Open source computer vision library, Doctor Dobbs J. 25 (11) (2000) 120–126.
[113] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow Large-scale Machine Learning on Heterogeneous Systems, (2015) https://www.tensorflow.org/about/bib.
[114] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, Proceedings of the 22nd ACM International Conference on Multimedia. MM '14, ACM, New York, NY, USA, 2014, pp. 675–678, http://dx.doi.org/10.1145/2647868.2654889 ISBN:978-1-4503-3063-3.
[115] D. Maturana, S. Scherer, Voxnet: a 3d convolutional neural network for real-time object recognition, 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2015) 922–928, http://dx.doi.org/10.1109/IROS.2015.7353481.

Jonathan Dekhtiar is currently a PhD candidate at UTC (Sorbonne Universités, Université de Technologie de Compiègne) in the Roberval research group (UMR CNRS 7337). His research interests are the possible applications of self-learning systems with a strong focus on deep learning techniques. One of his major focus is manufacturing defects detection using convolutional neural networks and unsupervised learning. https://www.jonathandekhtiar.eu, https://fr.linkedin.com/in/jonathandekhtiar.

Alexandre Durupt is an assistant professor at UTC (Sorbonne Universités, Université de Technologie de Compiègne) in the SIPP (Product/Process Systems Integration) team of the Roberval research group (UMR CNRS 7337). Since 2013, he is director (scientific coordinator) of the DIMEXP (DIgital Mock-up for multi-EXPertises integration – dimexp.utc.fr) joint research group between UTC and DeltaCAD (software company, www.deltacad.fr). Previously, he was assistant professor at UTT (Université de Technologie de Troyes) where he received his PhD degree in 2010. His research topic is the heterogeneous data integration for Smart DMU.

Matthieu Bricogne is an assistant professor at the Mechanical Engineering Department/Roberval laboratory (FRE - 2012) in Université de Technologie de Compiègne (UTC). He obtained his MSc in Mechanical Engineering and a Mechanical Engineering Degree at UTC in 2004. He has worked for 5 years for Dassault Systèmes, an industrially-leading company specializing in 3D and Product Lifecycle Management (PLM) software. He obtained his PhD degree in Mechanical Engineering from the Ecole de Technologie Supérieure in Montréal (ETS) and UTC in 2015. His main research topics are collaborative design, mechatronics system design and Knowledge Based Engineering within a PLM environment. Since 2013, he is vice director of the ANR LabCom DIMEXP (DIgital MockUp multi-EXPertises) joint laboratory UTC/DeltaCAD, whose main objective is to develop methods and models to improve the integration of multidisciplinary expertises and heterogeneous data in advanced digital environments.

Professor Benoît Eynard is currently the Director of the Institute for Mechatronics at the Université de Technologie de Compiègne - UTC. In 2007, he has joined UTC as a Full Professor for leading the Department of Mechanical Systems Engineering until 2012. He is also member of the UMR CNRS/UTC 7337 Roberval. Previously, he was Assistant Professor at the Université de Technologie de Troyes where he managed the MSc program on Information Technology for Mechanical Engineering. Since 2013, he is general chairman of the academic group of the French Association of Mechanical Engineering on Factories of the Future: integrated design and advanced manufacturing also known as S-mart (previously AIP-PRIMECA) network. He is also member of the IFIP working group 5.1. dealing with "Global Product development for the whole life-cycle" and of the Design Society where he currently leads the special interest group on "Design Methods for Cyber-Physical Systems". In 1999, he obtained his PhD on Computer-Integrated Manufacturing from the University of Bordeaux. Now, he is a recognised researcher in Product Lifecycle Management, collaborative design, systems engineering, mechatronic design, digital factory, manufacturing process management, eco-design and sustainable manufacturing. He has published over 70 papers of international journals and 200 international conferences. He also has been guest editor for 15 journal special issues and academic books.

Harvey Rowson Experienced Software Project Manager. Specialised in Agile Processes & DevOps, Distributed Architectures, Artificial Intelligence and PLM. Initial training in Mechanical Engineering with strong emphasis on Software Tools. Bilingual French/English. Strong experience in industrial high end Information Systems, in particular Web and Client/Server Architectures (JEE and.NET), Enterprise Resource Planning (ERP, Timesheets, Planning, Reporting), Product Lifecycle Management (PLM, PDM) and custom software dedicated to manufactured product and process design. Involved in managing internal software projects (internal ERP deployment and management of associated projects), in managing strategic projects for customers (managing projects for large industrial organisations, ranging from commercial phases to final delivery) and in coordinating or participating in collaborative research and development projects (state funded, academic partners, etc.). Particular experience on subjects like knowledge management and capitalisation, artificial intelligence/machine learning/expert systems/CSP/optimisation, pre/post processing of simulation software, advanced geometric algorithms, Nurbs/Bezier/Iges, 3D/VR/AR visualisation, Reverse Engineering of manufactured products, advanced product configurators, collaborative workflows, decision support, reporting and heterogeneous data integration.

Dimitris Kiritsis is Faculty Member at the Institute of Mechanical Engineering of the School of Engineering of EPFL, Switzerland, where he is leading a research group on ICT for Sustainable Manufacturing. His research interests are Closed Loop Lifecycle Management, IoT, Semantic Technologies and Data Analytics for Engineering Applications. He served also as Guest Professor at the IMS Center of the University of Cincinnati, and Invited Professor at the University of Technology of Compiègne, the University of Technology of Belfort-Montbéliard and at ParisTech ENSAM Paris. Prof. Kiritsis is actively involved in EU research programs in the area of Factories of the Future and Enabling ICT for Sustainable Manufacturing. He has more than 220 publications. Since September 2013 Dimitris is Chair of IFIP WG5.7 – Advanced Production Management Systems and member of the Advisory Group of the European Council on Leadership on Enabling Industrial Technologies – AG LEIT-NMBP. He is also founding fellow member of the International Society for Engineering Asset Management (ISEAM), of various international scientific communities in his area of interests including EFFRA and among the initiators of the IOF (Industrial Ontologies Foundry).