A Loosely-coupled Neural-symbolic approach to Compliance of Electric Panels

Vito Barbara[1,2], Dimitri Buelli[3], Massimo Guarascio[4], Stefano Ierace[5], Salvatore Iiritano[6], Giovanni Laboccetta[2], Nicola Leone[1], Giuseppe Manco[4], Valerio Pesenti[5], Alessandro Quarta[1,2,7], Francesco Ricca[1] and Ettore Ritacco[4]

[1]*University of Calabria - Via Bucci - 87036 - Rende (CS), Italy*
[2]*DLVSystem s.r.l. - Viale della Resistenza 19/C - 87036 - Rende (CS), Italy*
[3]*Elettrocablaggi s.r.l. - Via Gregorini, 41 - 24065 - Lovere (BG), Italy*
[4]*ICAR-CNR - Via Bucci, 8/9C - 87036 - Rende (CS), Italy*
[5]*Intellimech - Via Stezzano, 87 – 24126 - Bergamo (BG), Italy*
[6]*Revelis s.r.l. - V. J. F. Kennedy, 126 - 87036 - Rende (CS), Italy*
[7]*Sapienza, Università di Roma - Piazzale Aldo Moro 5 - 00185 - Roma (RM), Italy*

Abstract
This paper presents an ongoing work on project MAP4ID "Multipurpose Analytics Platform 4 Industrial Data", where one of the objectives is to propose suitable combinations of machine learning and Answer Set Programming (ASP) to cope with industrial problems. In particular, we focus on a specific use case of the project, where we combine deep learning and ASP to solve a problem of compliance to blueprints of electric panels. The use case data was provided by Elettrocablaggi srl, a SME leader in the market. Our proposed solution couples an object-recognition layer, implemented resorting to deep neural networks, that identifies components in an image of an electric panel, and sends this information to a a logic program, that checks the compliance of the panel in the picture with the blueprint of the circuit.

Keywords
Answer Set Programming, Neural-symbolic AI, Compliance

1. Introduction

With the rise of new technologies for Cyber-Physical Systems and Big Data Analytics, the industry moved a step forward to a new era in the field of manufacturing. This complex transformation, including the integration of emerging paradigms and solutions (e.g., *Machine and Deep Learning*, *Human-Computer Interaction*, *Cloud Computing* and *Industrial Internet Of Things* (IIoT) and *Blockchain*), is referred as *Industry 4.0*. In this evolving scenario, Quality

CILC 2022: 37th Italian Conference on Computational Logic, June 29 – July 1, 2022, Bologna, Italy
barbara.vito@unical.it (V. Barbara); dimitri.buelli@elettrocablaggi.it (D. Buelli); massimo.guarascio@icar.cnr.it (M. Guarascio); stefano.ierace@intellimech.it (S. Ierace); salvatore.iiritano@revelis.eu (S. Iiritano); laboccetta@dlvsystem.it (G. Laboccetta); nicola.leone@unical.it (N. Leone); giuseppe.manco@icar.cnr.it (G. Manco); valerio.pesenti@intellimech.it (V. Pesenti); alessandro.quarta@uniroma1.it (A. Quarta); ricca@mat.unical.it (F. Ricca); ettore.ritacco@icar.cnr.it (E. Ritacco)
0000-0001-7244-1778 (V. Barbara); 0000-0001-7711-9833 (M. Guarascio); 0000-0002-9742-1252 (N. Leone); 0000-0001-9672-3833 (G. Manco); 0000-0001-6319-2466 (A. Quarta); 0000-0001-8218-3178 (F. Ricca); 0000-0003-3978-9291 (E. Ritacco)

Control (QC) is greatly benefiting from the adoption of advanced Artificial Intelligence (AI) solutions, indeed AI techniques and tools can allow to speed-up or automatize processes of assessment about the integrity, the working capability and the durability of the products. In particular, automating the compliance verification process for products represents an important problem for all the companies operating in the manufacturing field since it is an indispensable but expensive and time consuming task in the supply chain. Recently, defect detection for electrical control panels (ECPs) is gaining growing interest as these tools are used in different scenarios. Indeed, to date ECPs are employed to control a wide variety of components exploited in industry e.g., they permits to control mechanical equipment, electrical devices, etc. In this work, we consider the problem to assist the human operator in verifying the *compliance* of blueprints with the control panel instances so to timely detect possible errors such as missing components, wrong connections and placements, etc. To the best of our knowledge the problem addressed in this paper has been scarcely investigated in the literature. However, some recent works studied tasks relevant for Industy 4.0 within the Predictive Maintenance field. For example, in [1] the authors introduce a framework that integrates Industrial Internet of Things (IIoT) devices, neural networks, and sound analysis for detecting anomalies in the supply chain. [2] proposes a holistic solution for quality inspection based on merging Machine Learning techniques and Edge Cloud Computing technology. A Deep Learning based approach for monitoring the process of sealing and closure of matrix-shaped thermoforming food packages is proposed in [3]. [4] defines a deep neural network (DNN) soft sensor enabling a fast quality control for the Printing Industry. Finally, in [5] the authors describe a deep learning based framework to detect/recognize machines for smart factories. In this paper we devise a novel approach integrating Machine Vision (MV) and Answer Set Programming (ASP) [6] to support the QC for electrical control panels. ASP is a well-established paradigm for declarative programming and non-monotonic reasoning developed in the area of Knowledge Representation and Reasoning [7, 6, 8]. ASP has been employed to develop many academic and industrial applications of AI [9, 10]. ASP is based on logic programming and non-monotonic reasoning, and it allows for flexible declarative modeling of search problems, by means of logic programs (collection of rules), whose intended models (answer sets) encode solutions [11]. In our case, we propose solution composed of two main phases: *(i)* first, we defined a Machine Learning flow based on a neural architecture to address the problem of recognizing the components (*Object Detection*) from the pictures of the panels, *(ii)* then, we realized an Answer Set Programming-based system used to compare the scheme reconstructed from the picture with its original blueprints, to discover possible mismatches/errors. The development of this work was inspired by Elettrocablaggi srl, a SME leader in the market of electric panels. This is one of the use cases of the MAP4ID “Multipurpose Analytics Platform 4 Industrial Data” project that aims at proposing suitable combinations of machine learning and ASP to cope with industrial problems. This paper, after presenting an overview of the architecture of our system for the compliance of electric panels, focuses on the logic-programming-based module of our approach.

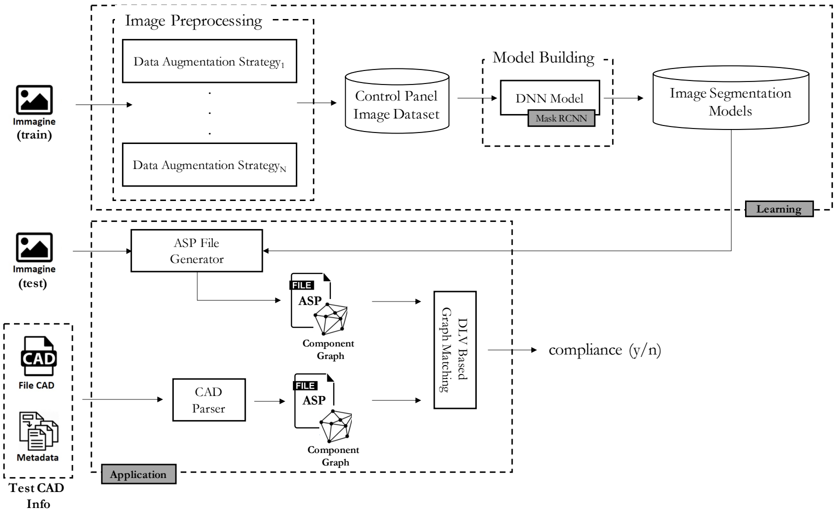

Figure 1: Framework for Automatic Compliance Verification.

2. Framework Overview

In this section we describe the solution approach devised to tackle the main problem i.e., automating the compliance verification process of the control panels. To this aim, we defined the framework shown in Figure 1 that includes two main macro-modules respectively named, *Component Detection* and *Quality Assessment*. The former block is devoted to recognizing the electrical components assembled in the cabinet. Basically, it includes a number of machine learning methods to train a model able to identify the components composing the panel from its picture. The *Model Building* module in Figure 1 allows for training the deep architecture used to perform the component detection. Basically, we used the Convolutional neural architecture proposed in [12], named *Mask R-CNN*, whose objective is to detect and highlight relevant items within an image.

The backbone of the Mask R-CNN used in our framework is a ResNet (Residual Network) [13], whose advantage is the generation of *skip connections* and *residual blocks*, whose usage allows for handling the well-known degradation problem (i.e., neural networks performing worse at the increasing depth), and ensures a good trade-off between convergence rapidity and expressivity/accuracy.

The Quality assessment module exploits ASP to tackle the task of compliance checking. It automatically compares the control panel scheme built starting from the neural network output and the corresponding blueprint to highlight any anomaly. The ASP-based module will be described in details in the following.

3. ASP-based compliance checking

In this section we describe the logic-based component of our architecture for compliance checking. The specification (logic program) described in the following can be fed to an ASP system to actually compute the solutions to the modeled program [14]. In the following we focus on the core parts of our solution and simplify some technical aspects that do not impact on the comprehension of the working principle of our solution. This is done with the aim of making the presentation more accessible and to meet space requirements. Hereafter, we assume the reader to be familiar with ASP, for more details refer to [6, 7, 8].

3.1. Input specification

In ASP the input specification is made by a set of "facts", that is assertions that model true sentences. Thus, the labelled blueprint of the circuit (we informally refer to it as cad) and output of the deep learning algorithm used to recognize the components and the output of the Mask-RCNN (we informally refer to it as net) is converted in a set of ASP facts of the following form:

```
object(LABEL, ID, X_TOP_L, Y_TOP_L, X_BOT_R, Y_BOT_R, MEMBERSHIP).
```

These facts provide information about the components like their label, id, and the top-left and the bottom-right coordinates and the membership. In particular, the membership is valued with "cad" if the object modeled is part of the blueprint of the panel, and "net" if it is recognized by the neural network in the actual picture we are comparing the blueprint.

Moreover, we also compute a graph of topological relations among objects, providing information on relative position and distance among objects. The relative position and the distance among components are actually calculated by our ASP program but for simplicity, we assume here they are given in input as facts of the form:

```
between(ID, START_ID, END_ID, DIR, MEMBERSHIP).
manhattan(ID1, ID2, DIST, MEM1, MEM2).
```

The predicate between denotes the neighbours for the component ID along the direction DIR; while the predicate manhattan specifies the manhattan distance between the two components ID1 and ID2, where the terms MEM1 and MEM2 stand for their membership.

3.2. ASP program

We now present ASP program (see Program 1) that encodes in a uniform way (w.r.t. the input instance provided as set of facts) the compliance problem.

First, the graph is preprocessed (lines 2-3), by calculating useful information about the relative positions of the objects. Next, according to the "guess-and-check" programming methodology a disjunctive rule guesses the mapping between "cad" components of the blueprint and "net" components predicted by the neural network (see lined 6-7). The disjunctive rule can be read as follows: 'Given a cad component and a net component of the same type, the two can be mapped, or not". The candidate solutions are filtered out by the constraints in lines 9-13, ensuring that the same element of the cad is not mapped twice, and the same element of the net is not mapped twice. The optimal mapping is obtained by weak constraints in lines 15-35. In detail, the program first minimizes the cad elements without a mapping (lines 15-16), then (also in order of

Algorithm 1 ASP program modeling compliance

```
1:  % Calculate auxiliary information
2:        previous(ID, Start_ID, D, M):- between(ID, Start_ID,_ , D, M).
3:        after(ID, End_ID, D, M):- between(ID, _, End_ID, D, M).
4:  % Guess mapping between cad components and net components
5:        simpObject(C1,ID1,M) :- object(C1,ID1,_,_,_,_,M).
6:        mapped(ID1,ID2) ‖ noMapped(ID1,ID2)
7:                    :- simpObject(C1,ID1,”cad”),simpObject(C1,ID2,”net”).
8:  % No element from the cad is mapped twice
9:        :- mapped(Cad_ID,Net_ID1), mapped(Cad_ID,Net_ID2),
10:           Net_ID1!=Net_ID2.
11: % No element from the net is mapped twice
12:       :- mapped(Cad_ID1,Net_ID), mapped(Cad_ID2,Net_ID),
13:           Cad_ID1!=Cad_ID2.
14: % Minimize the cad elements without a mapping
15:       atLeastOne(Cad_ID) :- mapped(Cad_ID,_).
16:       :~ simpObject(C1,ID1,”cad”), not atLeastOne(ID1). [1@3,C1,ID1]
17: % Optimize mapping by relative position
18:       :~ mapped(Cad_ID1, Net_ID1), mapped(Cad_ID2,Net_ID2),
19:           previous(Cad_ID1,Cad_ID2,DIR,”cad”),
20:           not previous(Net_ID1, Net_ID2, DIR,”net”).
21:           [1@2,Cad_ID1, Net_ID1,Cad_ID2,Net_ID2,DIR]
22:       :~ mapped(Cad_ID1,Net_ID1), mapped(Cad_ID2,Net_ID2),
23:           after(Cad_ID1, Cad_ID2,DIR,”cad”),
24:           not after(Net_ID1,Net_ID2,DIR,”net”).
25:           [1@2,Cad_ID1,Net_ID1,Cad_ID2,Net_ID2,DIR]
26:       :~ mapped(Cad_ID1, Net_ID1),
27:           previous(Cad_ID1, Cad_ID2, DIR,”cad”),
28:           absent(_,Cad_ID2). [1@2,Cad_ID1,Net_ID1,Cad_ID2,DIR]
29:       :~ mapped(Cad_ID1, Net_ID1),
30:           after(Cad_ID1, Cad_ID2, DIR,”cad”),
31:           absent(_,Cad_ID2). [1@2,Cad_ID1,Net_ID1,Cad_ID2,DIR]
32: % Optimize mapping by distance
33:       :~ mapped(Cad_ID, Net_ID),
34:           manhattan(Cad_ID, Net_ID, Dis,”cad”,”net”).
35:           [Dis@1,Cad_ID,Net_ID,Dis]
36: % Identify absent and in excess components
37:       mappedCad(ID1):- mapped(ID1,_).
38:       mappedNet(ID1):- mapped(_,ID1).
39:       absent(C1,ID1):- simpObject(C1,ID1,”cad”), not mappedCad(ID1).
40:       excess(C1,ID1):- simpObject(C1,ID1,”net”), not mappedNet(ID1).
```

priority) the weak constrains in lines 18-31 ensure that “If a cad component ID1 is mapped to a net component ID2, ID1 neighbors should be mapped to ID2 neighbors”. The mapping is further optimized considering distance (lines 33-35) between cad components and net components. The distance is optimal when the elements are in the same position in “net” and “cad”. Finally, the program identifies components that are absent or in excess w.r.t. the blueprint by rules in lines 37-40. The actual code used in our system implements others features, such as suggestions on where to place the absent elements in the right position inside the panel, and suggestions on where the misplaced components are expected to be moved. These are also obtained by logic rules that are omitted here to simplify the presentation and focus on the core of the solution.

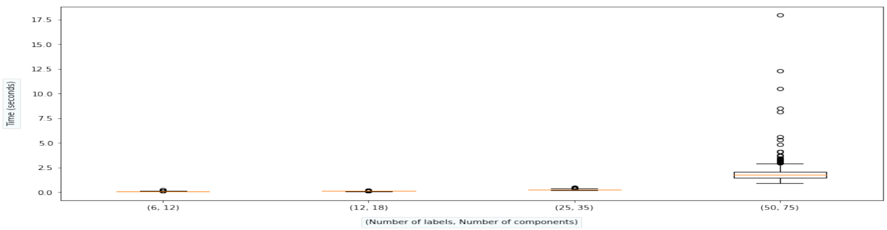

Figure 2: Timing boxplot

3.3. Preliminary results

The component detection module has shown a promising capability in automatically identifying the elements in the electric panels. In an experimentation on a dataset of about 10 thousand pictures (split in training and test sets, with a proportion of 90-10), the Mask R-CNN network was able to detect the 83.1% of all the components. On the other hand, the ASP-based component, that correctly implements our specification, always provides the expected answer, thus accuracy is only determined by the performance of the machine learning component. It might be interesting to know whether the ASP based component is efficient enough. To this end, we conducted an experiment to measure the execution time of the ASP-based component. Usually real panels are made of few components (the larger usually contains less than 25 components). We generated instances of compliance testing in a range of 6 to 50 labels (types of components), and of 12 to 75 components, and averaged over 500 samples the execution time need by DLV2 [15] to solve the instances. The results are reported in Figure 2. It is easy to see that our system can provide optimal answers in a short time, in the order of milliseconds for instances sized as real-world ones, and performance is acceptable (avg. 1.93s, max about 18s) also for instances of 75 components.

4. Conclusion

Our experience confirms that the loose combination of neural networks and ASP can result in an effective solution for checking the compliance of electric panels. The two modules are loosely coupled but complement each-other. Indeed, provided that the ML component knows how to recognize all the components, one can just provide a new logic specification of the blueprint to check compliance, with no need for retraining. As far as future work, we plan to further improve the neural network module to increase its performance, possibly trying to exploit ASP, implement a thorough validation analysis on data provided by Elettrocablaggi, and to develop an online panel compliance system featuring a user interface based on augmented reality glasses.

Acknowledgments

Supported by MISE under project MAP4ID - Multipurpose Analytics Platform 4 Industrial Data" Proj. N. 482 F/190138/01-03/X44.

References

[1] P. Tanuska, L. Spendla, M. Kebisek, R. Duris, M. Stremy, Smart anomaly detection and prediction for assembly process maintenance in compliance with industry 4.0, Sensors 21 (2021) 2376.

[2] J. Schmitt, J. Bönig, T. Borggräfe, G. Beitinger, J. Deuse, Predictive model-based quality inspection using machine learning and edge cloud computing, Advanced Engineering Informatics 45 (2020) 101101. doi:https://doi.org/10.1016/j.aei.2020.101101.

[3] N. Banus Paradell, I. Boada, P. Xiberta, P. Toldra, N. Bustins, Deep learning for the quality control of thermoforming food packages, Scientific Reports 11 (2021) 21887. doi:10.1038/s41598-021-01254-x.

[4] J. Villalba-Diez, D. Schmidt, R. Gevers, J. Ordieres-Meré, M. Buchwitz, W. Wellbrock, Deep learning for industrial computer vision quality control in the printing industry 4.0, Sensors 19 (2019) 3987. doi:10.3390/s19183987.

[5] H. Subakti, J.-R. Jiang, Indoor augmented reality using deep learning for industry 4.0 smart factories, in: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), volume 02, 2018, pp. 63–68. doi:10.1109/COMPSAC.2018.10204.

[6] G. Brewka, T. Eiter, M. Truszczynski, Answer set programming at a glance, Commun. ACM 54 (2011) 92–103. doi:10.1145/2043174.2043195.

[7] C. Baral, Knowledge Representation, Reasoning and Declarative Problem Solving, Cambridge University Press, 2003. URL: https://doi.org/10.1017/CBO9780511543357. doi:10.1017/CBO9780511543357.

[8] M. Gelfond, V. Lifschitz, Classical negation in logic programs and disjunctive databases, New Gener. Comput. 9 (1991) 365–386. doi:10.1007/BF03037169.

[9] E. Erdem, M. Gelfond, N. Leone, Applications of answer set programming, AI Magazine 37 (2016) 53–68. doi:10.1609/aimag.v37i3.2678.

[10] F. Calimeri, M. Gebser, M. Maratea, F. Ricca, Design and results of the fifth answer set programming competition, Artif. Intell. 231 (2016) 151–181. doi:10.1016/j.artint.2015.09.008.

[11] V. Lifschitz, Answer set planning, in: D. D. Schreye (Ed.), Logic Programming: The 1999 International Conference, Las Cruces, New Mexico, USA, November 29 - December 4, 1999, MIT Press, 1999, pp. 23–37.

[12] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980–2988. doi:10.1109/ICCV.2017.322.

[13] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[14] Y. Lierler, M. Maratea, F. Ricca, Systems, engineering environments, and competitions, AI Magazine 37 (2016) 45–52. doi:10.1609/aimag.v37i3.2675.

[15] M. Alviano, F. Calimeri, C. Dodaro, D. Fuscà, N. Leone, S. Perri, F. Ricca, P. Veltri, J. Zangari, The ASP system DLV2, in: M. Balduccini, T. Janhunen (Eds.), Logic Programming and Nonmonotonic Reasoning, LPNMR 2017, volume 10377 of *LNCS*, Springer, 2017, pp. 215–221.