# Meta-learning for Adaptive CNN Architecture Selection

Yesmina JAAFRA*, Jean Luc LAURENT** and Aline DERUYVER*
*ICube Laboratory, Université de Strasbourg, équipe CSTB, France
**Segula Technologies, Trappes, France

## Background

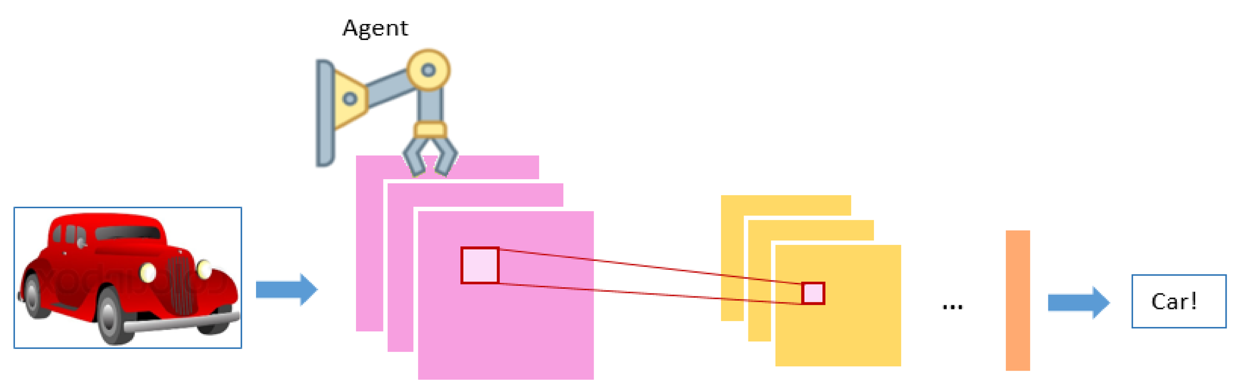
- The models issued from deep learning research have been implemented with a large success in a wide range of signal applications. Convolutional neural networks (CNN) have shown outstanding potentials in computer vision and more specifically in classification tasks [1].
- Although implementing the CNNs becomes easier due to available libraries like Tensorflow or PyTorch, selecting the appropriate architecture requires a significant human intervention in the search process.
- In this work, we aim at automating the task of CNN architecture design with meta-learning. We construct a generalizable approach that allow a continuous adaptation for specific learning tasks.

## Problem Statement

- CNN architecture consists of several types of layers including convolution, pooling, and fully connected.
- The network expert has to make multiple choices while designing a CNN such as the number and ordering of layers, the hyperparameters for each type of layer.
- Generate the model descriptions and related hyperparameters requires a trial and error manual search process mainly directed by intuition and experience.
- The number of available choices makes the selection space of CNN architectures extremely wide and impossible for an exhaustive manual exploration.

**Need for automation**

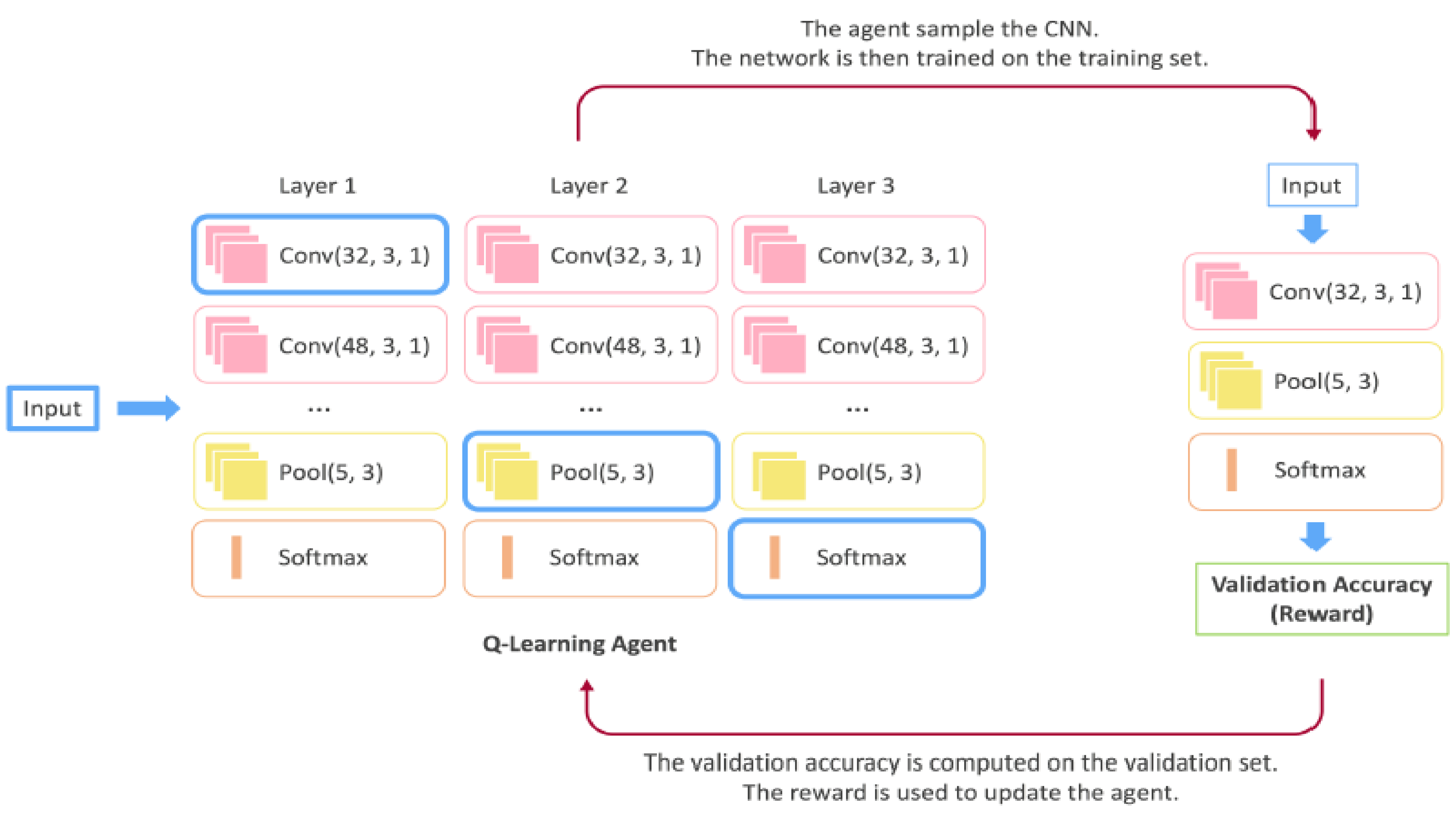
- The recent approaches for automating CNN architecture design are generally based on reinforcement learning (RL) techniques, e.g. Q-learning [2] and Recurrent neural networks (RNN) [3].




## Reinforcement learning for architecture design

**Exploration phase**: the agent identifies the states set of possible CNN layers and hyperparameters. It randomly operates actions, i.e. transition to next layer. The resulting CNN architecture is trained on test data in order to assess the associated reward (classification accuracy in our case).

**Exploitation phase**: on the basis of reinforcement learning policy, the agent selects the next actions maximizing the expected rewards [4].



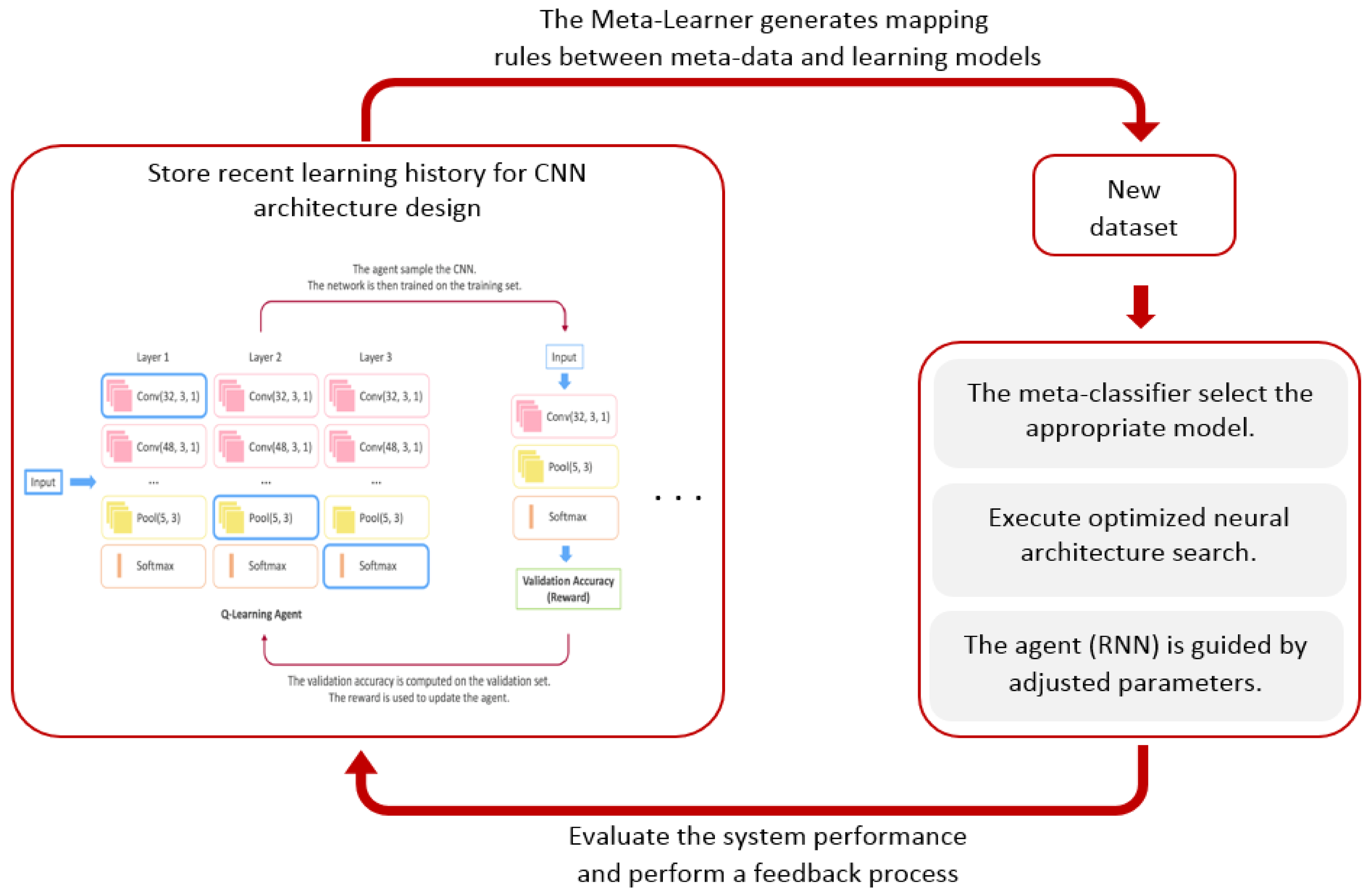

**Limits**

- The great number of possible layers associated with several types of features and hyperparameters induce a heavy computational cost for network generation.
- A network selected to solve a specific problem is not generalizable, i.e. it's hard to transfer to different cases [5].

## Meta-learning for continuous adaptation CNN architecture design

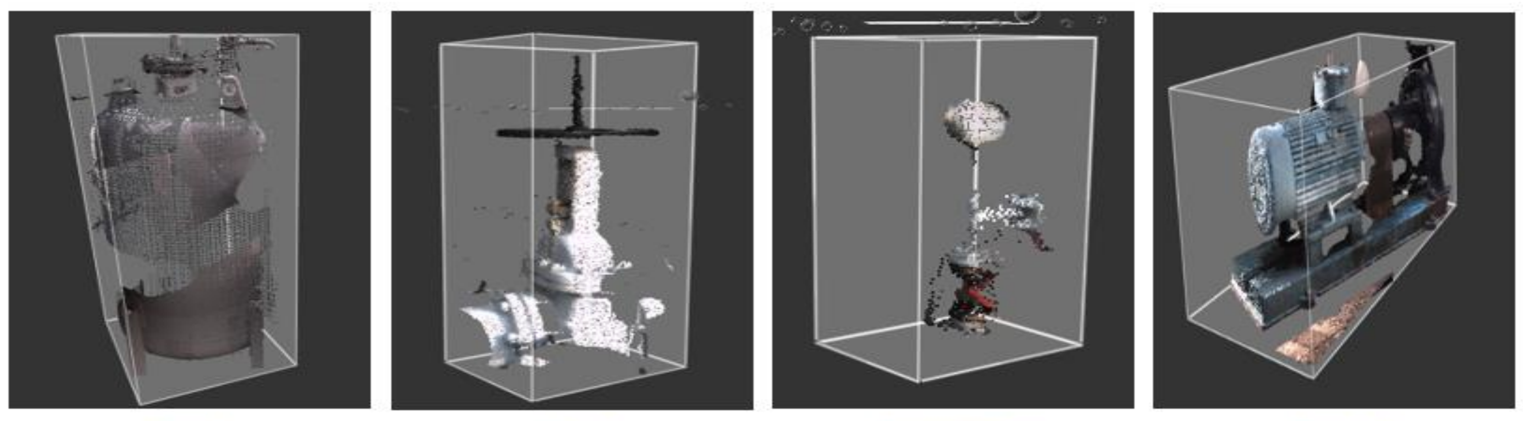
"How to profit from the **repetitive use** of a predictive model over **similar tasks**. The successful application of models in real-world scenarios requires **continuous adaptation** to new needs. Rather than starting afresh on new tasks, one would expect the learning mechanism itself **to relearn**, taking into account **previous experience**. This area of research, also known as **learning to learn**." [6]

- **First step**, we will use a model based on reinforcement learning to build optimized architectures for existing datasets.
- **Second step**: Perform meta-data analysis to allocate one of the identified models in base level to process a new dataset. The meta-learner is a shallow algorithm (e.g. SVM, k-NN) that will induce the prediction rules from meta-knowledge database.
- **Third step**: The generated meta-classifier will allow the transfer of knowledge through the initialization of RNN parameters and CNN hyperparameters for a quickly adaptation of the model to new tasks.

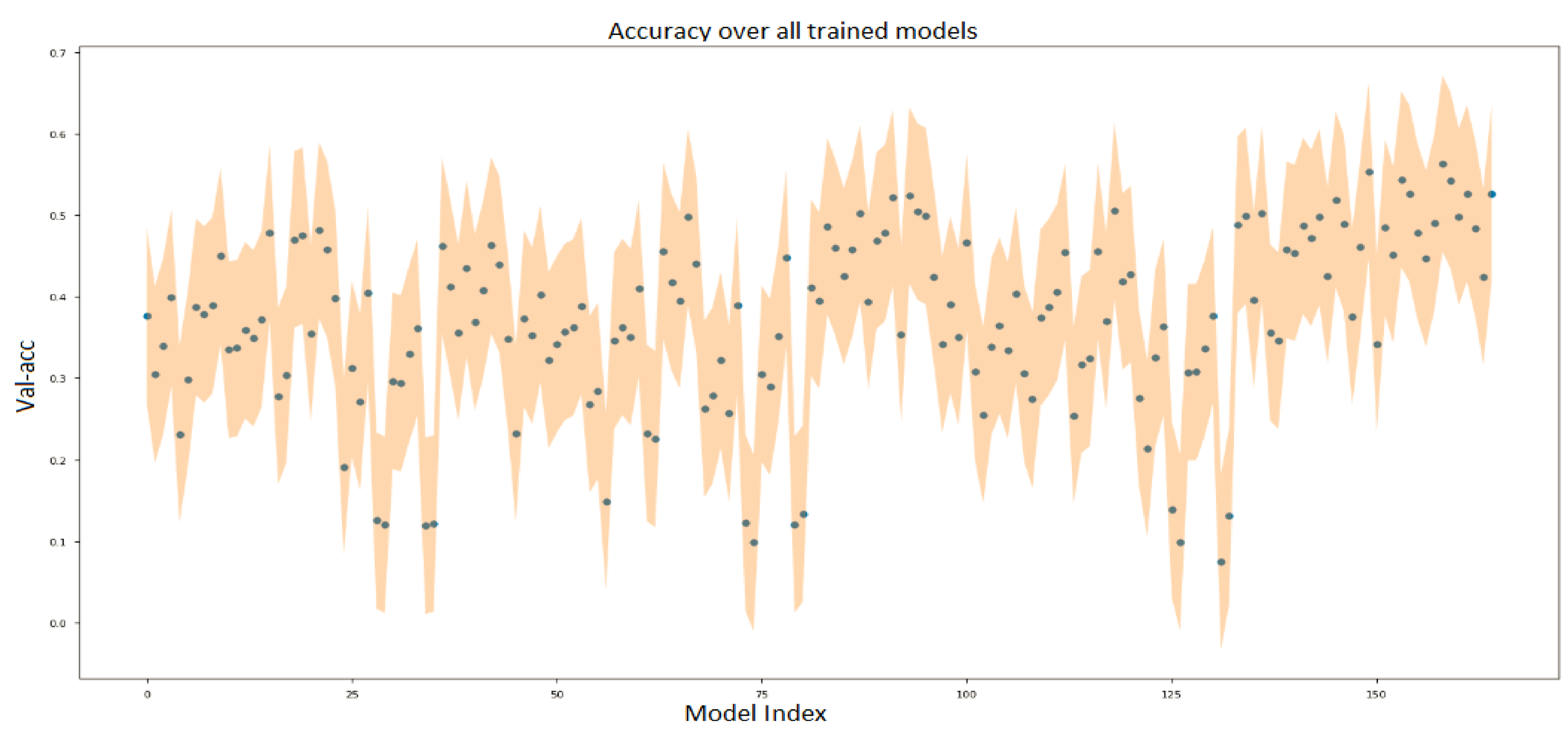



## Experiment and conclusion

- The experiment is conducted as part of Segula project for installations inventory and monitoring in refineries, platforms and petrochemical sectors.
- The Dataset is issued from 3D point clouds projected in 2D images. It is made up of 12000 images of size 200 * 200 pixels, 3 bands (RGB).
- 7 classes are considered : barometer, tank, fire, extinguisher, p-tank, pump, p-valve, pipe, valve.


Tank Valve Barometer Pump

- As base level learning (first step of our approach), we implemented a block-wise network generation strategy built on reinforcement learning. This CNN architecture design is inspired from progressive neural architecture search model [7].
- At this initial level of experiment, the best designed architectures generate a classification accuracy of 60%. The results are expected to improve with further implementation optimizations such as computing capacity and training consistency (volume of data and number of epochs).




- We note from the figure above that accuracy rate evolves with the increase of the number of trained models which confirms that the RL agent is improving its architecture selecting strategy with experience.

## Next Step

- Identify and test meta-features to generate meta-level knowledge: statistical measures, landmarking measures, model-based features and complexity measures [8].
- Perform second learning level and apply mapping rules between meta-data and learning models.
- Evaluate the deep learning strategy for our classification task by comparing it to shallow learning and assessing the implementation cost/performance ratio.

## References

[1] Y. LeCun, Y. Bengio and G. Hinton, «Deep learning» Nature, vol. 521, n°17553, pp. 436--444, 2015.
[2] B. Baker, O. Gupta, N. Naik and R. Raskar, "Designing Neural Network Architectures Using Reinforcement Learning," in ICLR, USA, 2017.
[3] B. Zoph and Q. V. Le, "Neural Architecture Search With Reinforcement Learning," in ICLR Google Brain, 2017.
[4] R. S. Sutton et A. G. Barto, Reinforcement Learning: An Introduction, A Bradford Book, London, 2017
[5] Z. Zhong, J. Yan and W. Wu, "Practical Block-wise Neural Network Architecture Generation," *CoRR,* Aug 2017.
[6] P. Brazdil, C. Giraud-Carrier, R. Vilalta et C. Soares, Meta-Learning - Applications to Data Mining, USA: Springer, 2010.
[7] C. Liu, B. Zoph, J. Shlens, W. Hua , L. Li and L. Fei, A. Yuille, J. Huang and K. Murphy, «Progressive Neural Architecture Search », in CoRR, 2017.
[8] A. Lorena, A. Maciel, P. de Miranda, I. Costa and R. Prudencio, « Data complexity meta-features for regression problems », in Machine Learning, USA, 2018.