Feeding Machine Learning with Knowledge Graphs for Explainable Object Detection*

Tanguy Pommellet[1] and Freddy Lécué[1,2]

[1] Thales, CortAIx, Canada
{firstname.lastname@ca.thalesgroup.com}
[2] Inria, France
freddy.lecue@inria.fr

Abstract. Machine Learning (ML), as one of the key driver of Artificial Intelligence, has demonstrated disruptive results in numerous industries. However one of the most fundamental problem of applying ML, and particularly Artificial Neural Network models, in critical systems is its inability to provide a rational of their decisions. For instance a ML system recognizes an object to be a warfare mine through comparison with its similar observations. No human-transposable rationale is given, mainly because common sense knowledge or reasoning is out-of-scope of ML systems. We developed an asset, combining ML and knowledge graphs to expose a human-like explanation when recognizing an object of any class in a knowledge graph of 4,233,000 resources.

1 Introduction and Related Work

The current hype of Artificial Intelligence (AI) mostly refers to the success of Machine Learning (ML) and its sub-domain of deep learning. However industries operating with critical systems are either highly regulated, or require high level of certification and robustness. Therefore, such industry constraints do limit the adoption of non deterministic and ML systems. Answers to the question of explainability will be intrinsically connected to the adoption of AI in industry at scale. Indeed explanation, which could be used for debugging intelligent systems or deciding to follow a recommendation in real-time, will increase acceptance and (business) user trust. Explainable AI (XAI) is now referring to the core backup for industry to apply AI in products at scale, particularly for industries operating with critical systems.

This is particular valid for object detection task in ML, as objet detection is usually performed from a large portfolio of Artificial Neural Networks (ANNs) architectures such as YOLO trained on large amount of labelled data. In such contexts explaining object detections is rather difficult due to the high complexity (i.e., number of layers, filters, convolutions phases) of the most accurate ANNs. Therefore explanations of an object detection task are limited to features

involved in the data and model e.g., saliency maps [1] or at best to examples [2], or prototypes [3]. They are the best state-of-the-art approaches but explanations are limited by data frames feeding the ANNs.

We present a system expanding and linking initial (training, validation and test) data (for a ML object detection task) with entities in knowledge graphs, in order (i) to encode context in data, (ii) to capture complex relations among objects, classes and properties and even (iii) to support inference and causation, rather than only correlation.

2 Explaining Objet Detection with Knowledge Graphs

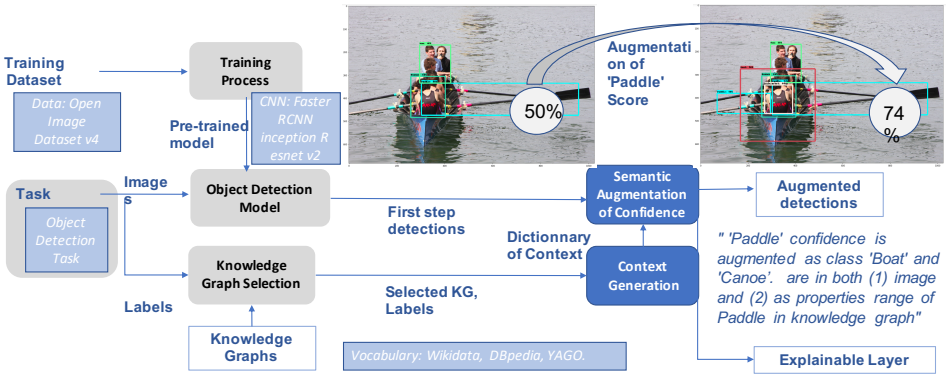

Fig. 1. Architecture: XAI for Object Detection using Knowledge Graphs. (color print).

2.1 Architecture

(ML) Training Process: We selected Faster RCNN (Region CNN, designed for object detection) Inception Resnet v2 [4]. Due to resources required to train a neural network on this dataset, we used pre-trained detection model. Among the pre-trained models on OID v4 available online, the faster RCNN with Inception Resnet v2 is the best tradeoff between detection performance and speed.
(ML) Object Detection Model: The configuration of the faster-RCNN is as following: The region proposal networks suggest 100 regions, with non maximum suppression IOU (Intersection-over-union) threshold at .7, and no NMS (Non-maximum suppression) hyper-parameters score threshold. The second stage of the RCNN infers detections for these 100 regions, with no additional NMS. These 100 predictions are the baseline of our work.
(ML) Object Detection Task: Our method is tested on a subset of the Open Image v4 Validation Dataset (described below). We designed our system to detect objects among the 600 categories of the Open Image challenge. All those categories are used to drive the search in Knowledge graph, and extract contextual information for each one of these categories to augment the baseline object detection approach.

Knowledge Graph Selection: Knowledge graphs are selected based on overage of labels of Open Image v4 Validation Dataset and an initial set of knowledge graphs: Wikidata, DBpedia, YAGO. Fuzzy match are used to optimize coverage, and the knowledge graph with the highest coverage is used for the further steps.

Context Generation: The task is to detect and localize objects out of a set C of 600 categories. For each category in $c \in C$ we determine a list of other categories C' that are 'close enough' (i.e., through identification of direct link among entities in the knowledge graph) to c so that if they are detected simultaneously, then we can confidently increase detections scores. The main difficulty is to identify a unique resource in the graph that will correspond to the given category. But often, several web resources can represent the same category, and categories can also have several homonym in the knowledge graph. Towards this challenge DBpedia is the most efficient graph to extract a unique resource associated to a category. Indeed, duplicates are quasi-null, disambiguate pages enabling to differentiate between homonyms, and redirection property enable to deal with synonyms issues. Moreover, it has a propriety that redirects every dbpedia resource toward YAGO and Wikidata. This is extremely interesting as it is hard to obtain a unique resource from a category name using wikidata itself, and it can be useful to combine several knowledge graph. The output of this process is a dictionary, where every key is a label of our detection task, and the value is the subset of the labels that are contextually linked to it.

Semantic Augmentation of Confidence: We obtain: 100 predictions with bounding boxes and a contextual dictionary extracted from the knowledge graph. The process is as following: (1) We define a first hyper-parameter (optimized during training) which is a score threshold s_t. Detections confidence under this threshold are not augmented, and cannot contribute to confidence augmentation of another detection. (2) For each prediction with an initial score higher than s_t, we derive a trustworthy indicator: it should indicate if the context (meaning the other detections on the image) is coherent with the detected category according to the dictionary extracted from the knowledge graph. We look at the list of linked label in the contextual dictionary. For each linked label, we look if it has also been detected in the image (with a confidence score higher than s_t). If positive then we add its confidence score to the trustworthy indicator. Then we compare the trustworthy indicator to a predefined threshold (new hyper-parameter). (3) If it is less to the threshold, the initial detection score is unchanged. The context does not bring more confidence about the detection. (4) If it is higher, we augment the initial score. To derive the score to add, we compute the same indicator than in the first step, but we do not take into account contribution whose did not reach the trustworthy threshold in the first step (we do not want bad predictions to infer in the increase of confidence).

Explainable Layer: During the augmentation process, we record the predictions that contribute to the increase in confidence of each detection. We obtain concepts that account for an intelligible and semantic explanation of prediction.

2.2 Demonstration

We illustrate our demonstration (Figure 2) on an image from the open image validation dataset. We feature detections with confidence score higher than .4 before and after the semantic augmentation of confidence.

Evaluation: We evaluate detection performance with the Open Images mean Average Precision @0.5 score used for the Open Image detection challenge[3]. The Average Precision score for a category is derived as the era under the Precision-recall curve for a specific IoU threshold (here 0.5). Precision measures how accurate is the predictions. i.e. the percentage of the predictions are correct and recall measures how good the identification of positive is. The mean Average Precision or mAP score is calculated by taking the mean AP over all classes.

Results: $mAP0.5 = 49.5\%$ with baseline, $mAP0.5 = 49.9\%$ with our approach. Our semantic augmentation confidence slightly improves the average detection performance of the model, while providing an interpretable layer due to the use of external information extracted from knowledge graphs.

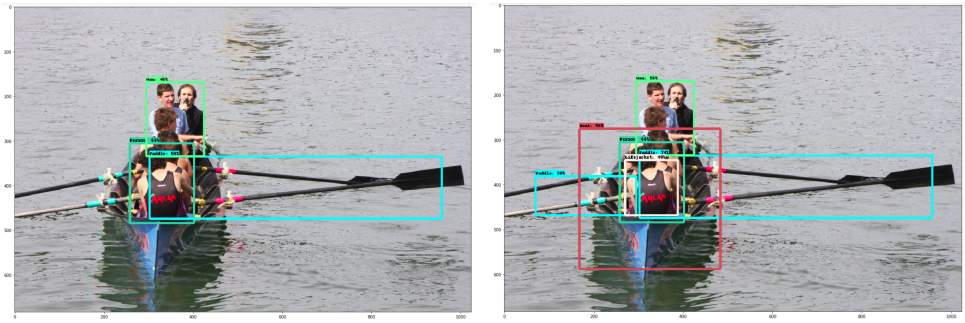

Fig. 2. Left image: results from baseline Faster RCNN: Paddle: 50% confidence, Person: 66%, Man: 46%. Right image: results from the semantic augmentation: **Paddle: 74%** confidence, Person: 66%, **Man: 56%**, **Boat: 58% with explanation: Person, Paddle, Water as part of the context and knowledge graph of concept Boat**.

References

1. Chang, C.H., Creager, E., Goldenberg, A., Duvenaud, D.: Interpreting neural network classifications with variational dropout saliency maps. In: Proc. NIPS. Volume 1. (2017) 6
2. Li, O., Liu, H., Chen, C., Rudin, C.: Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. (2018) 3530–3537
3. Kim, B., Koyejo, O., Khanna, R.: Examples are not enough, learn to criticize! criticism for interpretability. In: Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. (2016) 2280–2288
4. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. (2015) 1440–1448

[3] https://storage.googleapis.com/openimages/web/challenge.html