Design Studies and Intelligence Engineering
L.C. Jain et al. (Eds.)

doi:10.3233/FAIA220700

Interpretable Neural Symbol Learning Methods to Fuse Deep Learning Representation and Knowledge Graph: Zhejiang Cuisine Recipe Intangible Cultural Heritage Use Case

Zhongliang Yang[a,b,1] , Xingli Jia[b],Xinyu Zhang[b], and Jialu Tang[b]
[a] *Design Intelligence Innovation Center, China Academy of Art, Hangzhou 310024, China*
yzl@dhu.edu.cn
[b] *College of Mechanical Engineering, Donghua University, Shanghai 201620, China*

Abstract. Deep learning (DL) is difficult to provide explanations verified by non-technical audiences such as end-users or domain experts. This paper uses symbolic knowledge in the form of an expert knowledge graph, and proposes an interpretable neural-symbol learning (RF-YOLOv5) method, designed to learn symbols and deep representations. Finally, the deep learning representation and knowledge map are integrated in the learning process, so as a good basis for interpretability. Among them, the RF-YOLOv5 method involves specific two aspects of interpretation, respectively in reasoning and training time (1) YOLOv5-EXPLANet: experts alignment explained part of the auxiliary network architecture, combined convolutional neural network, using symbol representation, and (2) interpretable artificial intelligence training process, correct and guide the DL process and such symbol representation form of knowledge graph. The camera is placed above the refrigerator to detect the variety of ingredients, and then used in the RF-YOLOv5 method recommended by Zhejiang cuisine recipes, and demonstrates that using our method can improve interpretability while improving interpretability.

Keywords. Interpretable artificial intelligence, Deep Learning, Neural Symbol Learning, Knowledge graph, Object detection and classification

1. Introduction

Currently, Deep Learning (DL) has constructed many advanced models for solving the problem [1-5]. But these models are complex, opaque and difficult to debug, and often require large amounts of oversimplified annotated data to train, excluding a significant portion of the centuries-long knowledge of domain experts. At the same time, DL generates corresponding outputs by using corresponding shortcuts, making it very picky

[1] Corresponding Author

and difficult to output correctly. Conversely, most classical symbolic AI methods are interpretable, but their performance is neither comparable nor scalable.

To make black-box deep learning methods more interpretable, the topic of explainable artificial intelligence (XAI) has emerged. Given an audience, an XAI system produces details or justifications that make its function clear or easy to understand [6,7].

In line with the principles of responsible human-centered AI, having a specific and broad audience contributes to the inclusivity and accessibility of AI models. Furthermore, as advocated by [8,9], when deploying human-centered AI systems, broadening the inclusion of different minority groups and audiences can improve the effectiveness of AI models.

Therefore, fusing DL with domain expert knowledge becomes a key challenge, aligning deep learning with symbolic representations to bring interpretability [10]. To this end, this paper proposes an explainable neural-symbol (RF-YOLOv5) learning method, which is realized by exploring expert knowledge in the form of knowledge graph. The RF-YOLOv5 approach aims to make neuromyotonic models interpretable while providing more general interpretations for end users and domain experts. RF-YOLOv5 aims to improve the performance and interpretability of DL, especially a convolutional neural network (CNN) classification model. The RF-YOLOv5 method consists of three main components:

1. Neural processing component: for learning neural representations. In our example, YOLOv5-EXPLANet is used as a combined deep architecture to classify detected objects.

2. Symbol processing component: used to process symbolic representation. In this example, the knowledge graph is used to build the explicit knowledge of domain experts.

3. Neural Symbol Alignment Component: Used to guide the alignment of model outputs with symbolic interpretations.

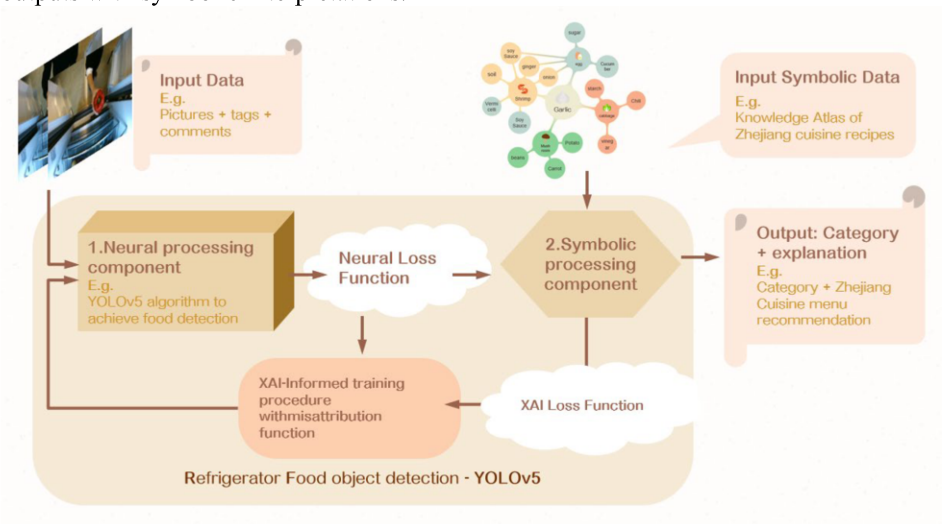

Figure 1. RF-YOLOv5 to interpretable neural symbolic learning.

This article illustrates the use of the RF-YOLOv5 method through a guided use case for refrigerator ingredient detection and Zhejiang cuisine recipe recommendation. The method link component is shown in Figure 1. Combination of modules can realize a general template architecture, which makes the fusion of representations of different properties possible. The RF-YOLOv5 approach can be adapted to the use case and allows the model to be trained in a continuous learning setting.

2. Image detection and results of YOLOv5-EXPLANet refrigerator

2.1. Refrigerator food material image detection data set

The team used raspberry Pi cameras to shoot and select 3,005 images of 15 categories on the ordinary single door refrigerator and double door refrigerator respectively. Data coverage content includes different light conditions, occlusion degree, packaging, background, aggregation scale of fruits and vegetables and FMCG. Examples of the refrigerator ingredients are shown in Figure 2, and the type and quantity statistics are shown in Table 1.

Table 1. Types and quantities of refrigerator food material experimental samples

Ingredients	Name	Count	Factor rate (%)
Fruit	banana	180	5.61
	pitaya	292	9.10
Vegetables	corn	241	7.51
	purple cabbage /	161	5.02
	freshly cut purple cabbage	158	4.92
	cauliflower	169	5.27
	tomato	163	5.08
	pumpkin	165	5.14
	eggplant	191	5.95
	carrot	316	9.85
	cap fungus	223	6.95
	baby cabbage	261	8.13
	cucumber	209	6.51
	celery	163	5.08
Eggs	eggs	317	9.88

Figure 2. Sample set of refrigerator food material images.

2.2. Experimental model construction for target detection

YOLOv5 algorithm selects 20% of 3005 samples, a total of 601 plots as test set, (training set + validation set): test set = 8:2; training set: validation set = 9:1. Platform hardware configuration used for the algorithm training: Intel (R) Core (TM) i5-7500CPU, The GTX1060 8G Memory GPU, Limited by the computer performance, Batch size, unified adopt 4,2; After the training session, training in the, MINOVERLAP = 0.8, Confidence = 0.001, Verification of the results for the 601 test sets under a threshold index of nmu_iou =0.5, It was evaluated from the aspects of precision Precision, recall rate Recall, F1 value, AP value and mAP value. The algorithm and environmental condition information used in the YOLOv5 target detection algorithm experiment is shown in Table 2.

Table 2. Types and quantities of refrigerator food material experimental samples

Algorithm	Backbone network	Input size	Epoch	Batch size	Learning rate
YOLOv5	CSPDarknet-53	640*640	0-50	4	1e-3
			50-100	2	1e-4

2.3. Analysis of experimental results

At present, the model convergence effect is good, and the identification accuracy reaches 93.61%, as shown in Figure 3. The image recognition effect is shown in Figure 4.

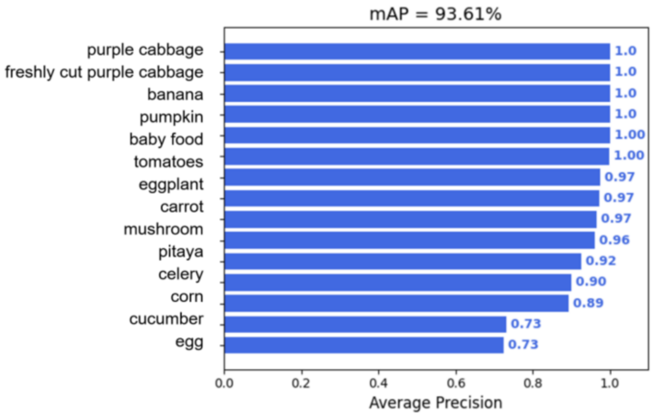

Figure 3. YOLOv5 model Map values.

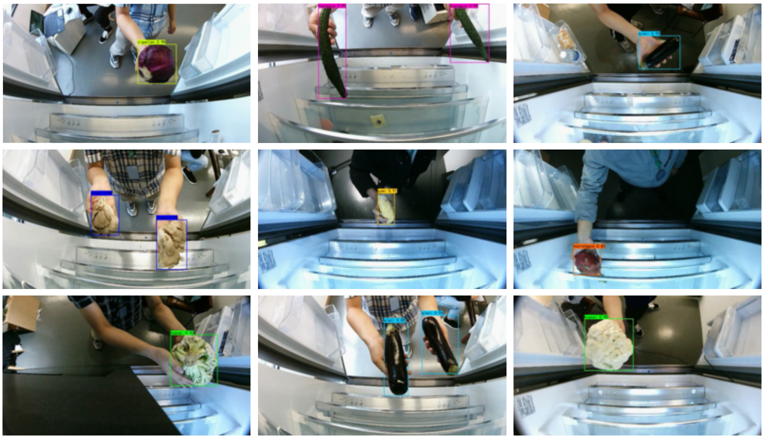

Figure 4. Image recognition effect.

3. Visualization method and system of Zhejiang cuisine recipe knowledge map

3.1. Zhejiang cuisine recipe knowledge atlas dataset

The team uses XPath to crawl Zhejiang cuisine recipe data from fully available web pages presented in a semi-structured form, as shown in Figure 5. Zhejiang cuisine menu data set contains "dishes", "specialty", "raw materials" ("is divided into" main "," accessories ",", "the)", "characteristic" and "production steps" five entity categories, "belong to", "main", "accessories", "ingredients"," production steps "," taste "," time-consuming "," process "and" difficulty " and other nine common relationship types. There are two data set versions, Mini lightweight version for 10 categories, 50 dishes; Pro expanded version for 362 categories, more than 8,000 dishes. In order to quickly build the specific function of the knowledge map, the team currently selects the Mini lightweight version as the data set for this experiment. Among them, the "food categories" include 10 categories, with 50 dishes related to each other, including various ingredients, characteristics and production steps.

Figure 5. Web page with semi-structured recipe data.

3.2. *Zhejiang cuisine menu entity level visualization*

The team uses the tree structure to store recipes and attribute data and visualize the hierarchical tree of the entity, as shown in Figure 6. Then, for the tree structure storage data, in triples format: "dishes" -belong to "boutique" specialty "," specialty "-main-" main material "," specialty "-" specialty "-ingredients-" ingredients "," specialty "-accessories- -" accessories "," boutique specialty "-production steps-" steps " list said all data. The data used for visualization is divided into graph structure data composed of triples vizdata.json and data entities_items.json composed of entity properties. Dataset entities, relationships, and triples number statistics are shown in Table 3.

Meal types:
_ Boutique specials
_ raw material
_ main material
_ auxiliary material
_ ingredients
_ characteristics
_ taste
_ time
_ process
_ difficulty
_ Preparation procedure

Figure 6. Entity-layer-level visualization.

Table 3. Zhejiang cuisine recipe data set data statistics

	Number of entities	Relationship types	Number of triples
Dish categories	10	1	50
Fine specialty dishes	50	2	224
raw material (Main material, auxiliary materials, and ingredients)	174	1	305
characteristic (Taste, workmanship, time-consuming, and difficulty)	50	4	200
production procedure	50	1	50
total	334	9	829

Triplet graph structure data vizdata.json stores dictionary data, "links" keys correspond to triples of all headers-relationship-tail entities, and "nodes" sets properties such as the type, name, and size of the node.

"Boiled Fish":
{
"Main Ingredients": [
"Grass carp: 1 ",
"Yellow Bean sprouts: Right amount"
].
"Accessories": [
"Dried chilies: in moderation ",
"Sichuan pepper: Moderate amount ",
"Ginger:1 piece"
].
"Feature": [
"Taste: Spicy",
"Craft: Cook ",
"Time: one hour",
"Difficulty: Normal"
].
"Making Steps": [
"1: Prepare the ingredients." .
"2: Clean and slice the fish... ",
...]
},...

3.3. *Visualization of Zhejiang cuisine recipe knowledge map*

D3 is a data-based document manipulation javascript library. D3 can combine data with HTML, SVG, and CSS to create interactive data charts. The team built a knowledge graph visualization system with D3 knowledge graph force-oriented graph and Neo4j respectively. D3 has better display and flexibility in visualization, so D3 is chosen for the visualization of knowledge graph.

For the relationship graph data obtained above, vizdata.json and entity attribute data entities_itmes.json are stored in the local project, because D3 visualization only supports reading json data from web services. Due to the word limit, this article presents several main modules of D3 visualization.

First, you need to set the visualization style. Then, you need to read the relationship graph data from the json file: use the links data in vizdata.json to drive the line width of the edge between the two nodes: add all the nodes, and set the nodes according to different types for each node Color: By clicking on the dot and text, the node switches between different modes: There is a switch for different types of entities, which determines whether a type of entity node is displayed: When the mouse hovers over an entity node, the attribute information of the entity is displayed. It can be displayed. If it is the entity of the boutique specialty dishes, the information such as the pictures of the dishes can be displayed: set the search function, and display all the nodes matching the keywords according to the keywords in the search box: the nodes of the same type of entities use the same color Indicates that when the mouse is over a node, it displays other entities associated with it and the relationship name between them; it has the same type of entity display switch, node display mode conversion, and supports search function; each dish is displayed in the information bar of the dish The corresponding finished product images are aligned with entities_aglin.py. After data cleaning and analysis, redundant information in food raw materials is eliminated. The system function display is shown in Figure 7.

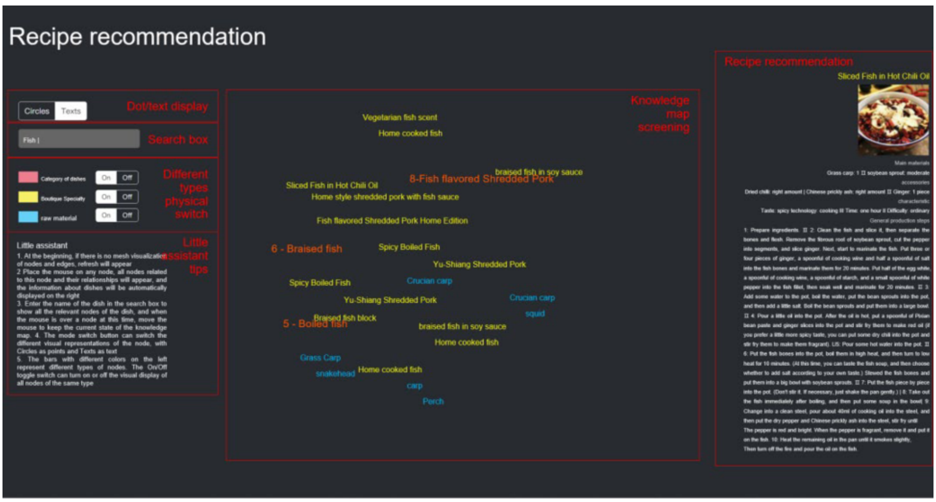

Figure 7. System function display (dot / text display, search, different types of entity switches, small assistant tips, recipe recommendation).

3.4. Target detection and knowledge graph alignment components

The acquisition camera above the refrigerator realizes the label output of the food material for the target detection through YOLOv5-EXPLANet, and then inputs it into the search box of the knowledge map to realize the menu recommendation of the corresponding food material.

4. Results and discussion

To achieve the challenges of fusion and alignment of deep learning representations with domain expert knowledge, we propose RF-YOLOv5 methods to integrate deep learning and symbolic representations, integrating neural symbols relatively together to form an interpretable feedback mechanism. We demonstrate the complete path of RF-YOLOv5 using the refrigerator food material dataset collected by the refrigerator raspberry Pi camera and the Zhejiang cuisine recipe dataset collected by the network, combining target detection and knowledge graph to form a good recipe recommendation function. The RF-YOLOv5 method constructed thus can continue to be improved and applied to more other fields.

Acknowledgments. This study was partly supported by the Zhejiang Provincial Key Laboratory of Integration of Healthy Smart Kitchen System, China (No. 2014E10014) and the National Social Science Fund of Art, China (No. 20ZD09,18ZD19).

References

[1] Natalia Díaz-Rodríguez,Alberto Lamas, et al.EXplainable Neural-Symbolic Learning (X-NeSyL) methodology to fuse deep learning representations with expert knowledge graphs: The MonuMAI cultural heritage use casell.Information Fusion 79 (2022) 58–83.

[2] Yann LeCun, Yoshua Bengio, Geoffrey Hinton, Deep learning, Nature 521 (7553)(2015) 436–444.

[3] Geoffrey E. Hinton, Ruslan R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[4] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, Yoshua Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: Proceedings of the International Conference on Machine Learning, PMLR, 2015, pp. 2048–2057.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones,Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, Attention is all you need,in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017.

[6] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, Li Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Rattern Recognition, 2014, pp. 1725–1732.

[7] Matthew D. Zeiler, Rob Fergus, Visualizing and understanding convolutional networks, in: European Conference on Computer Vision, Springer, 2014, pp.818–833.

[8] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.

[9] Chris Olah, Alexander Mordvintsev, Ludwig Schubert, Feature visualization, Distill 2 (11) (2017) e7, http://dx.doi.org/10.23915/distill.00007, https://distill.pub/2017/feature-visualization.

[10] Feng X Y, Mei W, Hu D S. Aerial Target Detection Based on Improved Faster R- CNN[J]. Acta Optica Sinica, 2018, 38(6):0615004.