A Constraint-Based Visual Dataset Generator

Thomas Eiter[1[0000−0001−6003−6345]], Nelson Higuera[1[0000−0003−3172−723X]], Katsumi Inoue[2[0000−0002−2717−9122]], and Sota Moriyama[2[0009−0004−2327−7720]]

[1] Vienna University of Technology (TU Wien),
Favoritenstrasse 9–11, Vienna, 1040, Austria
{thomas.eiter,nelson.ruiz}@tuwien.ac.at
[2] National Institute of Informatics (NII),
2 Chome-1-2 Hitotsubashi, Chiyoda City, Tokyo 101-0003, Japan
{inoue,sotam}@nii.ac.jp

Abstract. A current challenge in nerusoymbolic computation is to find benchmark tools that can easily produce data based on logical formulas. We address this by creating a constraint-based visual dataset generator. We use Answer Set Programming to define grid-like structures that contain geometric objects in the cells. By defining a constraint and its converse, we produce models that are then visually represented and can be readily used to train a neural network for binary classification or object detection and recognition.

Keywords: Neurosymbolic · Answer Set Programming · Computer Vision.

1 Introduction

Neurosymbolic computation [2] aims to bridge the gap between neural networks and logical reasoning systems. A key challenge in this field is the development of efficient tools that can generate data derived from logical formulas, which are crucial for testing neurosymbolic systems [5]. In response to this need, we present a novel constraint-based visual dataset generator. Using Answer Set Programming (ASP), we define structures containing objects where constraints govern the placement of the latter. By manipulating these constraints, we can generate datasets that either satisfy or violate specific conditions, enabling their use to benchmark neurosymbolic systems.

2 Constraint-Based Image Generator

We consider grid like structures that contain symbols. The symbols in the grids must respect certain logical rules. These can produce a variety of known puzzles such as sudoku or latin squares when the symbols are numbers. We use geometric objects for our dataset with attributes form, color and size.

```
{obj(X,Y,F,C,S) : form(F), color(C), size(S)} = 1 :- width(X), heigth(Y).
```

Consider now a possible constraint such that no two adjacent objects have the same form:

```
:- obj(X, Y, S, _, _), obj(X1, Y, S, _, _), X1=X+1.
:- obj(X, Y, S, _, _), obj(X, Y1, S, _, _), Y1=Y+1.
```

Its easy to introduce the converse of this constraint:

```
adj_same_shape :- obj(X, Y, S, _, _), obj(X1, Y, S, _, _), X1=X+1.
adj_same_shape :- obj(X, Y, S, _, _), obj(X, Y1, S, _, _), Y1=Y+1.
:- not adj_same_shape.
```

In this way we can create models that comply or break a constraint. We identify three types of constraints that are interesting for this puzzles, namely, spatial, equality and counting constraints.

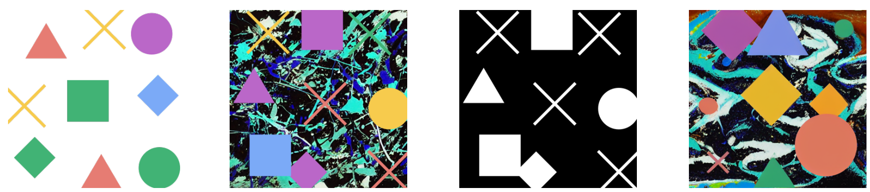

Fig. 1: Different stages of the generation of the images.

We then visualize the models by creating images of them. To increase the challenge for neural networks, we use the following:

1. Move the objects randomly inside their grids.
2. Use Stable Diffusion [1] to add backgrounds based on Jackson Pollock .
3. Use Stable Diffusion on the original image along a mask with the objects to generate small perturbations in the objects.

The output of our system are the images and labels for the objects, their bounding boxes, and whether the image that contains them complies or break the constraint.

We generated 1000 images that follow the example constraint and a 1000 that break it. We split the dataset into 1600 images for training and 400 for validation. The data was tested on YOLOv8n [4], where in 100 epochs it reaches an accuracy plateu of ∼80%.

3 Conclusion and Future Work

We successfully developed a challenging constraint-based visual dataset generator based on Answer Set Programming.

For future work, we aim to produce a baseline using YOLOv8 by modifying its loss function to include a term which verifies if the objects detected comply with the constraint or not [3].

References

1. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Rombach, R.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024. OpenReview.net (2024), https://openreview.net/forum?id=FPnUhsQJ5B
2. d'Avila Garcez, A.S., Gori, M., Lamb, L.C., Serafini, L., Spranger, M., Tran, S.N.: Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. FLAP **6**(4), 611–632 (2019), https://collegepublications.co.uk/ifcolog/?00033
3. Giunchiglia, E., Stoian, M.C., Khan, S., Cuzzolin, F., Lukasiewicz, T.: ROAD-R: the autonomous driving dataset with logical requirements. Mach. Learn. **112**(9), 3261–3291 (2023). https://doi.org/10.1007/S10994-023-06322-Z, https://doi.org/10.1007/s10994-023-06322-z
4. Jocher, G., Qiu, J., Chaurasia, A.: Ultralytics YOLO (Jan 2023), https://github.com/ultralytics/ultralytics
5. Singh, G.: Benchmarking symbolic and neuro-symbolic description logic reasoners. In: d'Amato, C., Pan, J.Z. (eds.) Proceedings of the Doctoral Consortium at ISWC 2023 co-located with 22nd International Semantic Web Conference (ISWC 2023), Athens, Greece, November 7, 2023. CEUR Workshop Proceedings, vol. 3678. CEUR-WS.org (2023), https://ceur-ws.org/Vol-3678/paper11.pdf