IE643: Deep Learning - Theory and Practice **July-Nov 2024**

Project Report: Creating Graphics Programs from Images

Team Name: Your team's name *Team Members: Roll numbers of team members*

Abstract

This project report contains the details on our project which aims to infer graphics software programs from hand-drawn images. We describe the deep learning tools involved in our project. In particular, we explain about the deep neural network architectures, the loss functions and activation functions used and the training algorithm used. We have tried a new deep learning architecture and superior results are presented using the proposed architecture.

1 Introduction

Provide a short introduction about the project with proper motivation and applications. Give a broad overview of the project without getting much into the details.

Explain the structure of the project report as below:

We provide a survey of existing literature in Section 3. Our proposal for the project is described in Section 4. We give details on experiments in Section 6. A description of future work is given in Section 8. We conclude with a short summary and pointers to forthcoming work in Section 9.

2 Project Workflow

Outline the entire workflow of the project, beginning with the problem statement and detailing how the team has advanced towards completing the task. This section should utilize concise bullet points, and a workflow diagram would be appreciated.

3 Literature Survey

This section should include all relevant work, papers, and approaches that the team has reviewed in order to achieve the task.

Example:

Our project draws inspiration from a closely related work by Ellis et al. [2]. In their paper [2], the authors describe a three-step process where an input hand-drawn image is first used to generate a program specification (spec) which is then used to generate a graphics software program which is later rendered to generate the hand-drawn image as a graphics object (like a bitmap image, jpeg image, etc.). This process is illustrated in Figure 1.

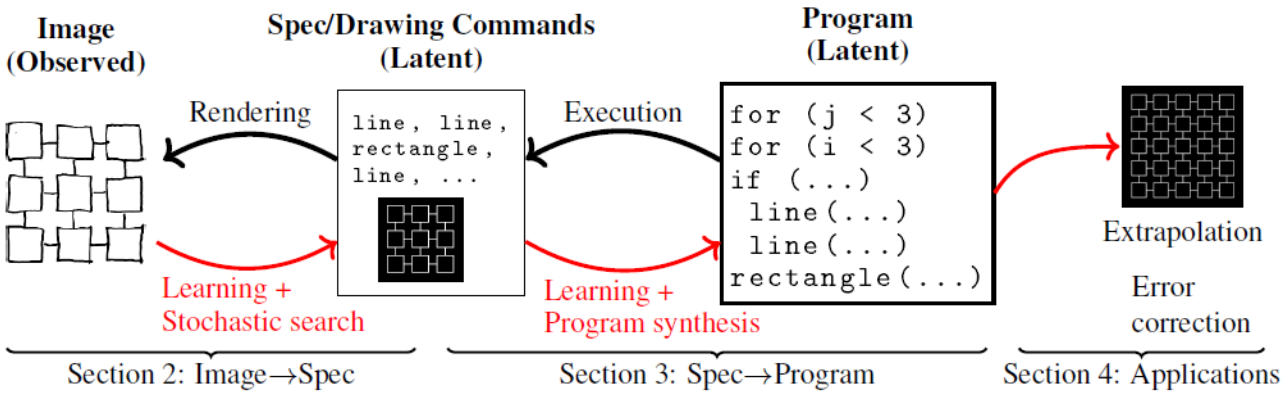

Figure 1: Three Step Process to generate program specs from image and then to convert specs to graphics programs and later render the programs to graphics objects

The neural network structure consists of two major components one of which is used to infer program specifications (**specs**) from the input image, and the other network to generate graphics programs to specs.

The neural network to infer spec S from an input image I uses ideas from Neurally guided procedural models [5] and attend-infer-repeat [3] models. The network is trained on sampled specs S and target images I from randomly generated scenes. The network is trained by maximizing the parametrized likelihood $P_\theta(S|I)$ of inferring a spec S given an image I, with the parameters θ being the parameters associated with the neural network. The network structure is illustrated in Figure 2. For further robustness to correct the network errors, the network outputs are further processed by a sequential Monte Carlo sampling scheme [1].

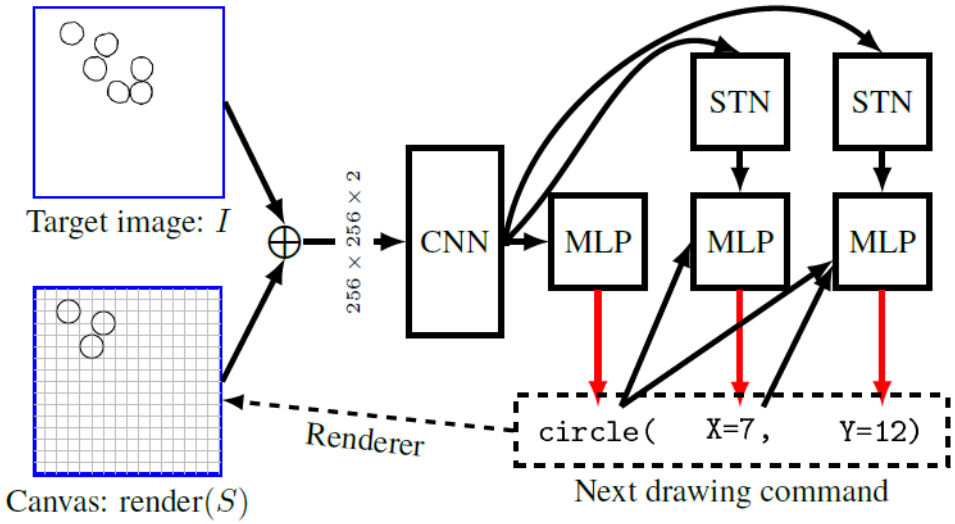

Figure 2: Neural architecture to infer specs from images.

Though the specs capture the contents of a scene, they do not contain information about repeated motifs

or symmetries. To capture these high-level features, graphics programs (typically software programs) are synthesized from specs. The grammar used in the synthesis of graphics program is restricted to a set of grammar rules given in Figure 3.

Program→	Statement; $\cdots$; Statement
Statement→	`circle`(Expression,Expression)
Statement→	`rectangle`(Expression,Expression,Expression,Expression)
Statement→	`line`(Expression,Expression,Expression,Expression,Boolean,Boolean)
Statement→	`for`$(0 \leq$ Var $<$ Expression$)$ `{ if` $($Var $> 0)$ `{` Program `}; ` Program `}`
Statement→	`reflect`(Axis) `{` Program `}`
Expression→	$\mathbb{Z}\times$Var+$\mathbb{Z}$
Axis→	`X` = $\mathbb{Z}$ | `Y` = $\mathbb{Z}$
$\mathbb{Z}\rightarrow$	an integer

Figure 3: Grammar rules used for graphics program.

The best graphics program that satisfies the grammar rules listed in Figure 3 is synthesized so that it best captures the spec. The closeness of a synthesized program p with the spec S is captured by an indicator function $\mathbb{I}(p\text{is consistent with } S)$. In addition the cost of a program p (denoted by $\text{cost}(p)$) is measured using the number of program statements in p. Thus the next network seeks to find the minimum cost program which is consistent with the program rules in Figure 3. Thus the following objective is used:

$$\text{PROG}(S) = \arg \max_{p \text{ obeys Grammar}} \mathbb{I}(p\text{is consistent with } S)\exp(-\text{cost}(p)). \quad (1)$$

To measure this consistency of a program p with a spec S, a Sketch tool [4] is used. Sketch tool transforms the consistency check condition to a constraint satisfaction problem and uses a SAT solver to find a min-cost program satisfying the spec.

A few experiments on a dataset are provided which indicate the usefulness of the approach.

4 Proposed Approach or Approaches

Give details on the proposed approach or approaches, which team has tried. Explain how it is different from existing work. Explain all the neural network structure to be used. Explain all other required details.

4.1 Work done before prep-presentation review

This section can be populated from your prep-presentation project report.

4.2 Work done after prep-presentation review

Give details on the work done after the prep-presentation project review. Explain how it is significant. Explain all the neural network structures used. Explain all other required details.

5 Data set Details

Explain all details on the data sets used for the project. Description of data size and attributes, nature and type of data (image/audio/text/video etc.), data pre-processing techniques used should be illustrated. Other relevant details on data procurement (the website from where the data is obtained), and how the data is to be used in experiments should be described.

6 Experiments

The team should include all experiments, even those with negative results, and address all experiments that were performed.

Give details on experiments performed. Training procedure and algorithm, the settings used for optimization algorithm and other relevant algorithmic details need to be included. Details on the hardware configuration should also be described.

7 Results

Add suitable plots and tables describing the results obtained during the project. Comparative results should be included if new ideas are tried. Description of the results and the inferences made using the results should be described.

8 Plan for Novelty Assessment

Explain the work to be done further for Novelty Assessment.

9 Conclusion

Conclude by giving a summary of your project, summarizing the problem, the methods used, and the significance of results obtained. Give some indications of future work to be done.

References

[1] Arnaud Doucet, Nando de Freitas, and Neil Gordon (Eds.). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag New York, 1 edition, 2001.

[2] Kevin Ellis, Daniel Ritchie, Armando Solar-Lezama, and Joshua B. Tenenbaum. Learning to infer graphics programs from hand-drawn images. In *Advances in Neural Information Processing Systems*, 2018.

[3] SM Eslami, N Heess, and T Weber. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems*, 2016.

[4] Armando Solar Lezama. *Program Synthesis By Sketching*. PhD thesis, EECS Department, University of California, Berkeley, 2008.

[5] Daniel Ritchie, Anna Thomas, Pat Hanrahan, and Noah Goodman. Neurally-guided procedural models: Amortized inference for procedural graphics programs using neural networks. In *Advances in Neural Information Processing Systems*, 2016.