Business Process Sketch Recognition

Bernhard Schäfer[1,2]

[1] Intelligent Robotic Process Automation, SAP SE, Walldorf, Germany
bernhard.schaefer@sap.com
[2] Data and Web Science Group, University of Mannheim, Mannheim, Germany
bernhard@informatik.uni-mannheim.de

Abstract. In early stages of a BPM project, simple process diagrams are often sketched on paper or whiteboard. Transferring a process sketch into existing modeling systems is a tedious manual process. Yet, there is little existing research on how to support professionals in seamlessly transferring their sketches into formal models. We address this gap with a technique that automatically converts a sketched process into a structured model. To recognize the symbols and structure of handwritten flowcharts, we have developed *Arrow R-CNN*. Arrow R-CNN is the first deep learning detector for flowchart structure recognition. It outperforms existing systems on a public flowchart dataset by a wide margin. We plan to incorporate the missing components for end-to-end flowchart recognition and then adapt our technique to other business process notations. We also consider integrating knowledge from a process repository to correct recognition errors. Similarly, a repository could be used to match a sketch or fragment thereof to models in the repository.

1 Introduction

For an initial sketch of a business process, pen and paper can be a suitable approach. During process discovery workshops or other interactive modeling scenarios, simple process diagrams are sketched on whiteboards and brown paper [11, p.85]. The sketching process can be supported by haptic tools such as sticky notes or magnetic BPMN elements. During or after the workshop, a process analyst manually constructs a BPMN model from the sketched process model [11, p.174]. While constructing a high-quality BPMN model requires strong expertise in process modeling, transferring a sketch into an interchangeable model format could be partially automated. The need for such an automation tool has been expressed by customers of a leading BPM software provider [28]. Although some prototypes have been proposed to this end, these prototypes only recognize basic symbols, and do not consider connections, handwriting, and more complex elements such as swimlanes [20, 28]. Our work addresses this gap with a technique for end-to-end recognition of sketched business processes from photos.

The paper is organized as follows. Section 2 briefly surveys related work in business process diagram recognition. Section 3 describes our developed handwritten flowchart recognition system. Section 4 outlines our planned extensions towards business process sketch recognition and matching, including identified problems that threaten those ambitions.

2 Related Work

There are numerous works on gesture-based diagram editors that operate on touch-input devices such as interactive whiteboards [6, 7, 10, 14, 17, 19, 21, 24]. Recognition of sticky notes, handwritten on a tablet or smartphone, has been addressed within the remote collaboration tool Tele-Board [12, 13]. On these touch-enabled devices, the gestures and handwriting are recorded as a temporal sequence of strokes. The recognition based on strokes is commonly referred to as online recognition. In contrast, our work focuses on offline recognition, where the input is a raster image and thus less structured. Offline recognition is considered more challenging than its online counterpart [3]. To our knowledge, there is little work on offline business process diagram recognition. Zapp et al. [28] present a prototype to recognize EPC diagrams from images. To validate their recognizer, the authors collect 108 private images of sketched EPCs. These sketches do not contain handwritten text, and their system does not recognize arrow connections. Due to their varying form and shape, text phrases and arrows are considered the greatest challenge in flowchart recognition [4]. In the context of Tangible BPM, Lübbe et al. developed a prototype that recognizes basic BPMN symbols from a sketch photo [20]. In a second step, the user manually annotates connections, handwriting, and more complex elements such as swimlanes.

In the area of handwritten flowchart recognition, a lot of research took place after the publication of an online dataset in 2011 [1]. The 419 flowcharts in the dataset describe algorithms, such as the neural network training procedure shown in Fig. 1. Even though algorithms conceptually differ from business processes, there are similarities from a recognition standpoint. Following the dataset release, various systems for online recognition were proposed [2, 4, 5, 18, 25, 26]. More recently, the flowchart dataset has also been used for offline recognition by plotting the smoothed strokes onto a white image. Similar to [28], Bresler et al. [3] use a stroke reconstruction preprocessing step and then continue with their online recognizer proposed in [4]. Julca-Aguilar [15] trains an object detector for flowchart symbol recognition. While achieving promising results for symbol recognition, their system is not capable to recognize the structure of a flowchart.

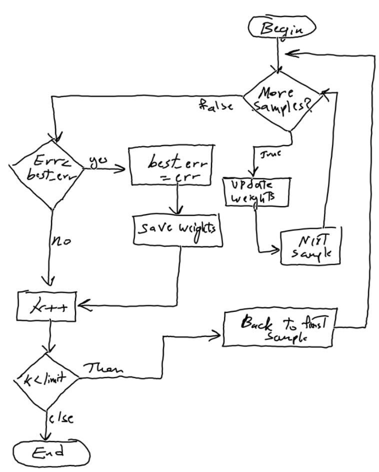

Fig. 1. Awal et al. [1] flowchart sample: neural network training procedure algorithm

3 State of the Project

As discussed in Section 2, there are no public business process sketch datasets. The closest to our knowledge is the public flowchart dataset by Awal et al. [1].

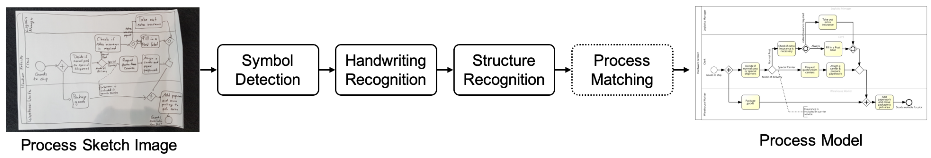

Fig. 2. Business process sketch recognition pipeline

While the flowcharts in this dataset describe algorithms instead of business processes, they are related from a recognition perspective. Thus, we have placed our initial efforts into developing an accurate recognizer for this type of flowcharts. We have proposed *Arrow R-CNN*[3], an offline handwritten flowchart recognizer that achieves state in the art on mentioned dataset. Arrow R-CNN extends the Faster R-CNN object detection system by Ren et al. [23], which uses a convolutional neural network (CNN) for detecting objects in an image. Our system recognizes flowchart elements, their arrow interconnections and performs text detection. As an example, our system is able to recognize the symbols and structure of the flowchart in Fig. 1 without a single error. Arrow R-CNN outperforms existing offline systems by a wide margin (97.9% vs. 84.2% symbol recognition rate). It also achieves state of the art in online recognition, without having knowledge of the temporal order of strokes or other explicit stroke information.

4 Planned

Our process sketch recognition system is composed of the stages shown in Fig. 2:

1. *Symbol Detection*: recognize visual elements (e.g. nodes, arrows, text phrases)
2. *Handwriting Recognition*: identify the written text of each detected phrase
3. *Structure Recognition*: detect relationships and form business process graph
4. *Process Matching*: optionally match graph against processes in repository

The remainder of the section discusses the open points regarding those steps and our intended work to go from flowchart to business process recognition.

4.1 Handwriting Recognition

For end-to-end flowchart recognition, two components are still missing: *handwriting recognition* and *text to symbol mapping*. We do not consider handwritten text recognition a focus topic of this dissertation and plan to use an off-the-shelf recognizer if possible. Regarding text to symbol mapping, in the flowchart dataset text phrases are commonly located within a node or close to an arrow. We plan to exploit this observation with a rule-based assignment strategy. If this approach is insufficient for other notations, we would try to learn this assignment with a neural relationship proposal network inspired by [27].

[3] Submitted to the International Workshop on Graphics Recognition (GREC 2019)

4.2 Natural Image Recognition

The flowcharts in [1] have a white background and do not contain image noise. Our system is based on Faster R-CNN, an object detector which can recognize objects in all sorts of image backgrounds. We believe that our system can be trained to recognize process diagrams from a wide span of images, such as the one in Fig. 2. We plan to verify this hypothesis by collecting our own dataset. To this end, we want to photograph handwritten process sketches on whiteboards or paper from different angles and distances. To simulate process discovery workshops, we might also use sticky notes for representing activity and control nodes.

We consider the size of this dataset the key challenge threating this ambition. In general, training deep learning models requires a lot of data. While the required volume can be reduced through data augmentation and transfer learning, it is difficult to estimate the magnitude of required images upfront.

4.3 BPMN Recognition

Since BPMN is a widely-used standard for process modeling, it is a natural choice for our work besides less formal notations such as flowcharts or simple linear sequences. From a recognition perspective, BPMN is far more complex than flowcharts due to the larger number of elements and the hierarchical structure. A first step towards BPMN sketch recognition could be the recognition of computer-generated BPMN diagrams. To that end, we would curate a dataset of BPMN models and for each create diagrams using different layouts and modeling software. This diagram recognizer could be used to generate a BPMN XML format from a BPMN diagram embedded in an image or document.

Overall, the goal is to maximize the flexibility of users w.r.t the process notation and the sketching surface and tools they can use. To handle multiple notations, a *notation image classifier* could be used, which predicts the predominant notation in an image. Recognition would then be handled by a set of notation-specific models. Alternatively, an inherent multi-notation recognizer could generate a notation-independent *business process graph* [9]. A second step would convert this graph into the desired notation.

4.4 Process Sketch to Repository Matching

Given a process model or fragment thereof, the problem of retrieving similar models from a process repository has been studied extensively [8, 9, 16, 22]. Regarding process sketches, a process matcher could identify fragments of the sketch that correspond to a process from a repository. During whiteboard modeling, this could be used to notify professionals when a similar model already exists. Leveraging the process repository can also improve the overall recognition accuracy. For example, handwriting recognition errors could be alleviated by comparing recognized texts with commonly used activity labels. Similarly, activity label and type co-occurrence statistics could resolve cases when the neural flowchart recognizer is uncertain about the symbol type. Albeit interesting, we are not sure yet to what extent this matching problem fits into the scope of this dissertation.

References

1. Awal, A.M., Feng, G., Mouchère, H., Viard-Gaudin, C.: First Experiments on a new Online Handwritten Flowchart Database. In: Document Recognition and Retrieval XVIII. p. 78740A (Jan 2011)
2. Bresler, M., Prùa, D., Hlavác, V.: Modeling Flowchart Structure Recognition as a Max-Sum Problem. In: 2013 12th International Conference on Document Analysis and Recognition. pp. 1215–1219 (Aug 2013)
3. Bresler, M., Průša, D., Hlaváč, V.: Recognizing Off-Line Flowcharts by Reconstructing Strokes and Using On-Line Recognition Techniques. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 48–53 (Oct 2016)
4. Bresler, M., Průša, D., Hlaváč, V.: Online recognition of sketched arrow-connected diagrams. International Journal on Document Analysis and Recognition (IJDAR) **19**(3), 253–267 (Sep 2016)
5. Carton, C., Lemaitre, A., Coüasnon, B.: Fusion of Statistical and Structural Information for Flowchart Recognition. In: 2013 12th International Conference on Document Analysis and Recognition. pp. 1210–1214 (Aug 2013)
6. Chen, Q., Grundy, J., Hosking, J.: SUMLOW: Early design-stage sketching of UML diagrams on an E-whiteboard. Software: Practice and Experience **38**(9), 961–994 (2008)
7. Damm, C.H., Hansen, K.M., Thomsen, M.: Tool Support for Cooperative Object-oriented Design: Gesture Based Modelling on an Electronic Whiteboard. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 518–525. CHI '00 (2000)
8. Dijkman, R., Dumas, M., García-Bañuelos, L.: Graph Matching Algorithms for Business Process Model Similarity Search. In: Business Process Management. pp. 48–63. Lecture Notes in Computer Science (2009)
9. Dijkman, R., Dumas, M., van Dongen, B., Käärik, R., Mendling, J.: Similarity of business process models: Metrics and evaluation. Information Systems **36**(2), 498–516 (Apr 2011)
10. Donaldson, A., Williamson, A.: Pen-based Input of UML Activity Diagrams for Business Process Modelling. In: Proc HCI 2005 Workshop on Improving and Assessing Pen-Based Input Techniques, Edinburgh. vol. 3, p. 26 (2005)
11. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.: Fundamentals of Business Process Management. 2 edn. (2018)
12. Gericke, L., Wenzel, M., Gumienny, R., Willems, C., Meinel, C.: Handwriting recognition for a digital whiteboard collaboration platform. In: 2012 International Conference on Collaboration Technologies and Systems (CTS). pp. 226–233 (May 2012)
13. Gumienny, R., Gericke, L., Quasthoff, M., Willems, C., Meinel, C.: Tele-Board: Enabling efficient collaboration in digital design spaces. In: Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD). pp. 47–54 (Jun 2011)
14. Hammond, T., Davis, R.: Tahuti: A Geometrical Sketch Recognition System for UML Class Diagrams. In: ACM SIGGRAPH 2006 Courses. SIGGRAPH '06 (2006)
15. Julca-Aguilar, F.D., Hirata, N.S.T.: Symbol Detection in Online Handwritten Graphics Using Faster R-CNN. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). pp. 151–156 (Apr 2018)
16. Klinkmüller, C., Weber, I., Mendling, J., Leopold, H., Ludwig, A.: Increasing Recall of Process Model Matching by Improved Activity Label Matching. In: Business Process Management. pp. 211–218. Lecture Notes in Computer Science (2013)

17. Kolb, J., Rudner, B., Reichert, M.: Towards Gesture-Based Process Modeling on Multi-touch Devices. In: Advanced Information Systems Engineering Workshops. pp. 280–293. Lecture Notes in Business Information Processing (2012)
18. Lemaitre, A., Mouchère, H., Camillerapp, J., Coüasnon, B.: Interest of Syntactic Knowledge for On-Line Flowchart Recognition. In: Graphics Recognition. New Trends and Challenges. pp. 89–98. Lecture Notes in Computer Science (2013)
19. Lentzsch, C., Nolte, A.: From sketching to modeling – Supporting end-users to elicit processes. In: Multikonferenz Wirtschaftsinformatik (MKWI). p. 12 (Jan 2016)
20. Lübbe, A.: Tangible Business Process Modeling. PhD Thesis, Universität Potsdam (Nov 2011)
21. Mangano, N., Sukaviriya, N.: Inkus: A Freehand Method of Creating Business Process Models. In: Proceedings of the Seventh Sketch-Based Interfaces and Modeling Symposium. pp. 143–150. SBIM '10 (2010)
22. Polyvyanyy, A., Ouyang, C., Barros, A., van der Aalst, W.M.P.: Process querying: Enabling business intelligence through query-based process analytics. Decision Support Systems **100**, 41–56 (Aug 2017)
23. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: Advances in Neural Information Processing Systems 28. pp. 91–99 (2015)
24. Schenk, C., Schimmler, S., Minas, M.: Investigating uni-stroke gesture input for diagram editors on large wall-mounted touchscreens. In: 2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC). pp. 279–283 (Oct 2017)
25. Wang, C., Mouchère, H., Lemaitre, A., Viard-Gaudin, C.: Online flowchart understanding by combining max-margin Markov random field with grammatical analysis. International Journal on Document Analysis and Recognition (IJDAR) **20**(2), 123–136 (Jun 2017)
26. Wu, J., Wang, C., Zhang, L., Rui, Y.: Offline Sketch Parsing via Shapeness Estimation. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (Jun 2015)
27. Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D.: Graph R-CNN for Scene Graph Generation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 670–685 (2018)
28. Zapp, M., Fettke, P., Loos, P.: Towards a Software Prototype Supporting Automatic Recognition of Sketched Business Process Models. Wirtschaftsinformatik 2017 Proceedings (Jan 2017)