



Bachelor Thesis
in Information Systems and Management

Optical Character Recognition for Labels Using Deep Learning

Johannes Reichle
Matriculation No. 04797218

Supervisor Prof. Dr. Rainer Schmidt
Date of Submission XX.XX.2022

Declaration

I hereby certify that I have written this bachelor thesis
on my own and that I have not used any sources or aids
other than those indicated.

Munich, the XX.XX.2022

.....
Johannes Reichle

Abstract

Here abstract for Bachelor Thesis.

Keywords: Deep Learning, Optical Character Recognition, Scene Text Recognition, Literature Review

Contents

List of Figures	2
List of Tables	2
Abbreviations	3
1 Introduction	4
1.1 Motivation	4
1.2 Problem description	4
1.3 Methodology	5
1.4 Expected results	6
2 Theoretical Foundation	7
2.1 Machine Learning	7
2.2 Deep Learning	7
2.3 Optical Character Recognition	8
3 Problem analysis	9
3.1 Use Case	9
3.2 Literature on Defining Quality	10
3.3 Qualities	11
4 Current Research	14
4.1 Different pipeline Frameworks	14
4.2 Techniques/Modules for Improvement	15
5 Discussion	16
5.1 Analysis	16
5.2 Reflection	16
5.3 Outlook	16
6 Conclusion	17

CONTENTS

Bibliography	18
--------------	----

List of Figures

3.1 Examples for label images	10
---	----

List of Tables

3.1 Qualities specific to use case — exclusion criterias	9
3.2 Qualities identified through literature	12

Abbreviations

DL Deep Learning

DNN Deep Neural Network

ML Machine Learning

MLS Machine Learning System

NN Neural Network

OCR Optical Character Recognition

STR Scene Text Recognition

Chapter 1

Introduction

1.1 Motivation

Optical Character Recognition (OCR) is the concept of extracting typed, handwritten or printed text from an image Zhao et al. (2020). Techniques for this concept have improved a lot due to the advances in the field of Deep Learning (DL) (Zhao et al., 2020). When compared to traditional methods DL improves automation, effectiveness and generalization (Chen et al., 2021). DL is a technology based on Neural Networks (NNs) where data is processed in multiple layers to extract complex features to solve a given problem (Shrestha and Mahmood, 2019). DL has only caught on in the recent years as the big computational cost has been met by improvement in computer hardware as well as in automatic feature learning (Ponti et al., 2017; Chen et al., 2021). Applying these new capabilities and finding the right solution in the space of DL for the use case of extracting information of labels is the focus of this thesis. This is an interesting task as performance of OCR systems in complex scenes is still challenging (Zhao et al., 2020). Such scenes entail natural scenes captured by a camera. OCR in these conditions is also known as Scene Text Recognition (STR) (Chen et al., 2021; Baek et al., 2019). Factors such as complex backgrounds, noise, perspective and variability in fonts, colors and sizes, of scene texts complicate the process (Hu et al., 2020b; Chen et al., 2021; Baek et al., 2019).

1.2 Problem description

Technicians in the field work with different equipment. It is useful to digitize the labels of such equipment, to keep an overview over the inventory (Abramowicz and Corchuelo, 2019). The goal of this thesis is to find a solution which

simplifies the digitization of equipment labels. The research question guiding the process is most crucial: Which state of the art DL approaches for STR are viable for the use case of extracting textual label data from images. The definition of the viability of an approach has to be determined for this. What qualities such as detecting alpha-numeric strings or suitability despite inadequate image conditions must a solution have (Ghosh et al., 2017; Hu et al., 2020b)?

It is difficult to assess how well a DL approach performs before it has been implemented and tested on the specific problem or dataset (Arpteg et al., 2018). Therefore, multiple promising approaches that can be implemented and experimented with must be identified and compared. The research and discussion of techniques from end-to-end STR to dividing the process into text detection and text recognition is centered Chen et al. (2021) around the requirements which are given by the problem.

The article Ashmore et al. (2021) defines four phases of the Machine Learning (ML) lifecycle, namely, Data Management, Model Learning, Model Verification and Model Deployment. Only the substage Model Selection from Model Learning will only be looked at in the scope of this thesis. Other aspects such as data analysis, implementation, training, deployment and maintenance of a solution in a production environment shall not be performed.

1.3 Methodology

The methodology of this thesis can be labeled as a literature review (Snyder, 2019; Torraco, 2005). The goal is to provide an overview over current DL pipelines and models that can help in choosing which to implement and test to solve the specific problem defined in Section 1.2 and more detailed in Chapter 3.

The research question guiding the process is most crucial: Which state of the art DL approaches for STR are viable for the use case of extracting textual label data from images. In order to improve the validity for the subsequent analysis, the problem is dissected further. This includes analysing the specific use case as well as researching which qualities have been identified as generally critical for STR models. The qualities are taken from literature which covers ML in general to literature which covers STR.

The literature is identified through searching in reputable journals. All research after 2017 which pertains to STR is regarded as relevant. OCR solutions may not hold validity in practice, as the image qualities can vary in the defined problem (Chen et al., 2021). An important criteria is that the paper contributes to the ML model. This extends to the whole pipeline from preprocessing an image to the final result of the model. Conclusive to the

distinction in Section 1.2, contributions to other stages in the ML lifecycle are not examined. Therefore, keywords for the search include: Deep Learning, Scene Text Recognition, Pipeline, Preprocessing, End-to-end, Text Recognition, Text Detection, Text Segmentation.

The identified literature is synthesized into an overview over the most common approaches for STR. This includes listing important factors for DL such as the number of parameters, or which type of layers are used to achieve success. The overview will be organized into the categories for the ML pipeline, such as End-to-End solutions as in Xing et al. (2019) or a split into Text Detection and Text Recognition as in Yang et al. (2021); Chen and Li (2018).

In the analysis possibly viable approaches are compared with the qualities defined in Chapter 3. The approaches are analysed in detail in regards to commonalities as well as differences and the possible effect on the feasibility. The analysis thus shows which approaches are worthwhile to apply the whole ML lifecycle to.

1.4 Expected results

In addition to a deeper understanding of the problem and its detailed definition, the literature review lays the foundation for finding the right approach for the extraction of textual information from images with equipment labels through literature review. In the subsequent analysis different approaches are highlighted for their theoretical fit as a solution.

In the following, the structure of this thesis is listed and each chapter's expected result is detailed along with its benefits for the overall objective of producing an overview of state of the art STR relevant for the problem described in Section 1.2. comprehension of the following chapters is gathered. This includes general principles of DL and by extension ML but also of OCR. In Chapter 3 the problem from Section 1.2 is addressed in more detail. The result shall be a firm understanding of qualities that a solution must possess. These requirements are the point of focus for the further examination of STR techniques. After laying the foundation, in Chapter 4 current research in regards to the identified requirements is examined. The resulting overview can be viewed as a basis for a decision when it comes implementing a practical solution. Therefore, it enables the discussion in Chapter 5. Here not only the results and the availability of a solution but also the methodology of this work are assessed critically. The conclusion is a summary of the results compared to the expected results detailed in this chapter as well as an outlook for further research into the topic.

Chapter 2

Theoretical Foundation

Work with Goodfellow et al. (2016) as much as possible

2.1 Machine Learning

1. Loss Function / Error Metrics: CTC-loss, Cross entropy loss and maximum likelihood estimation (Goodfellow et al. (2016))
2. Supervised — Unsupervised / Categorization
3. Optimization techniques: Stochastic-Batch Gradient Descent, GD Momentum, Adam
4. Bias-Variance tradeoff / Overfitting — Underfitting

2.2 Deep Learning

1. ANN / MLP
 - Architecture → Input, Hidden, Output
 - Feedforward
 - Optimization → Backpropagation, SGD, ADAM, ...
2. Regularization: L0,L1,L2, Dropout, Dropconnect
3. important architectures
 - CNN
 - RNN

- Specific foundation architectures for relevant approaches
4. transfer learning: reuse parameters from pretrained models

2.3 Optical Character Recognition

Chapter 3

Problem analysis

This chapter entails an analysis of the problem which is the research question's foundation. It is crucial, as the quality of requirements ultimately determines the quality of the overview and subsequent analysis.

Requirements for a software system that involves ML and thus DL differs from the traditional approach. The data-driven software components are not entirely defined by the programmer but are influenced by data. The system acts with dependency on the test data (Siebert et al., 2021). This poses a challenge in determining requirements and measuring quality of results (Nakamichi et al., 2020). Instead of categorizing functional and non-functional requirements, like for traditional software projects (Zowghi et al., 2014), qualities that a Machine Learning System (MLS) must possess are defined.

3.1 Use Case

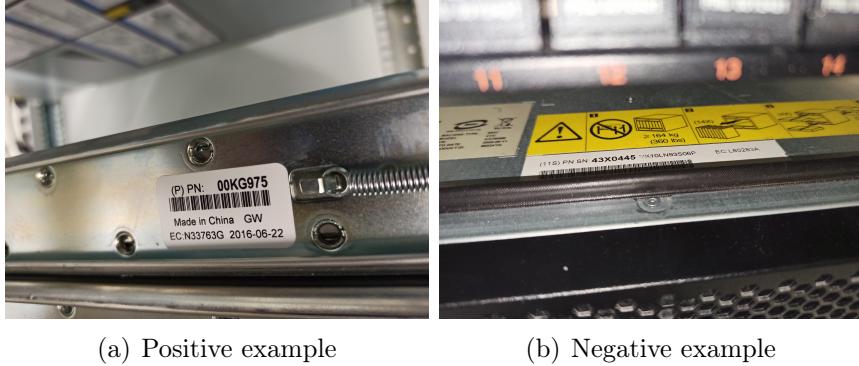
The problem can be depicted by a use case. This use case sets the foundation for determining requirements for an approach because qualities derive from the intended purpose of use (Siebert et al., 2021). Table 3.1 gives an overview over the relevant properties that can be derived from the use case. For this

Alphanumeric recognition	Recognize alphanumeric strings such as serial numbers
Semantics retention	Retain semantics given implicitly by space, structure and rotation of text in labels
Offline Capabilities	Perform extraction process offline

Table 3.1: Qualities specific to use case — exclusion criterias

thesis, the basic use case is as follows: A technician takes a photo of a device

label with his smart phone. For this the technician is situated in locations like a cable shaft. Due to this, there's no internet availability. The process from taking the image to storing the extracted text safely must work offline. The resulting image contains printed textual information which must be extracted by an application on the smart phone. Space and structure of this information can vary from label to label (see figure 3.1). The text, spacing and structure carries semantic information which can be important for later processing in the scope of a business process (Chen et al., 2021). The goal is to extract the text and preserve semantics that are implicitly provided through structure and space. This means text and the respective coordinates, height, width and a possible rotation angle must be output as the result (Yang et al., 2021). Those values can then be transformed into other formats such as JSON or HTML as needed. In addition to this, the labels can contain arbitrary alphanumeric



(a) Positive example

(b) Negative example

Figure 3.1: Examples for label images

strings such as serial numbers (see figure 3.1). This results in the requirement that the DL model has to be able to recognize sequences that are not part of a predefined lexicon (Ghosh et al., 2017). The qualities for the MLS that can be derived directly from the use case (see table 3.3) can be regarded as excluding criterias, because an approach that does not possess the qualities in question, cannot be regarded as viable for the use case.

3.2 Literature on Defining Quality

In the article Ashmore et al. (2021) the qualities are identified and assigned to different challenges in regards to working with MLS: Development Challenges, Production Challenges, Organizational Challenges. Because the only the Model Selection substage of the lifecycle is performed, the challenges and their qualities are not relevant for this thesis, as they concern the operational aspect of MLSs.

In Nakamichi et al. (2020); Siebert et al. (2021) systematic approaches for identification and documentation of qualities are detailed. In MLSs various entities interact to in order to produce the desired functionality. The paper Nakamichi et al. (2020) suggests that in order to adequately evaluate the qualities, it is essential to not only consider the model but the entire MLS. These entities are data, model, environment, system/infrastructure (Nakamichi et al., 2020; Siebert et al., 2021). The article Siebert et al. (2021) differentiates between system and infrastrucure. The infrastrucure represents given hardware and available libraries, whereas the system depicts the software that surrounds the model in the runtime environment. The data view pertains to the quality of development and runtime data (Siebert et al., 2021). The model consists of subcomponents organized in directed acyclc graph building a pipeline. This directed acyclic graph depicts everything from processing the images to the extracted information (Siebert et al., 2021). The environment entity covers the external aspects to the MLS which may interact with it (Siebert et al., 2021). In the scope of this work the environment entails mostly the conditions in which images are taken. For this thesis the entities data and system cannot be regarded as given. The entities environment and infrastrucure are only losely defined through the use case. That is why the systematic approaches cannot be performed in the scope of this thesis. For example Siebert et al. (2021) proposes to follow the systematic CRISP-DM approach of identifying qualities. It cannot be performed due to the lack of data and the other entities. Instead many qualities that are highlighted by research that fit the problem are taken into account along with two critical qualities (alphanumeric recognition, semantic retention) that are directly derived from the use case. When it comes to documenting the identified qualities, both Nakamichi et al. (2020) and Siebert et al. (2021) define a meta model for qualities that combines qualities with measurement methods and values and assignes them to an entity of the MLS. The implementation and testing phase are not performed in the scope of this thesis and the difficulty in assessing the performance ahead of those phases, prevents the evaluation of measurements. Additionally, experimental results from literature can only be compared as long as factors such as hardware, platform, source code, configuration and dataset are uniform (Arpteg et al., 2018). Comparing models through results of difrent papers is troublesome, because different papers might use different evaluation and testing environments (Baek et al., 2019). This applies to studies that present an overview such as Chen et al. (2021); Long et al. (2021). These studies can only be regarded as guiding values because the performance for a specific dataset cannot be predicted without testing on it Arpteg et al. (2018). That's why targets for measurements are not defined, as evaluation would only deliver a false sense of certainty.

3.3 Qualities

In addition to the qualities that arise directly from the use case, literature reveals a number of common qualities in regards to MLS (see table 3.3), some of which can be regarded as relevant and other do not hold any relevance for the specific use case. The qualities are taken from literature which covers ML in general to literature which covers STR. Only qualities that concern the model will be looked at because the model is the focus of this thesis. The qualities may however be influenced by other entities.

Relevant	Irrelevant
Appropriateness	Fairness
Performance	Interpretability
Robustness	Reusability
Performance efficiency	

Table 3.2: Qualities identified through literature

The appropriateness quality refers to the ability to perform the type of task that is required by the use case (Siebert et al., 2021; Nakamichi et al., 2020). For this thesis this applies to STR models. Additionally, the properties which are derived from the use case (see Table 3.1), can be bundled together under this quality.

‘An ML model is performant if it operates as expected according to a measure (or set of measures) that captures relevant characteristics of the model output’ (Ashmore et al., 2021). For the performance quality, a measure is chosen depending on the type of task to be solved (Siebert et al., 2021). The F-Score is an example for a metric that is used to compare different models Chen et al. (2021); Long et al. (2021). Performance is usually measured with a test dataset that is independent from training and validating a model in order to approximate the generalization performance Goodfellow et al. (2016); Nakamichi et al. (2020).

The robustness of a model concerns environmental uncertainty Ashmore et al. (2021). Due to the uncontrolled environment of STR in the practical aspect of taking the images on-site beneficial image properties can not be guaranteed (Chen et al., 2021). Robust text extraction can be influenced by factors such as complex backgrounds, text form (text rotation, font variability, arrangement), image noise (lighting conditions, blur, interference and low resolution) and access (perspective, shape of text) (Oyedotun et al., 2015; Ghosh et al., 2017; Chen et al., 2021). Therefore, these properties have to be accounted for when determining the viability for an approach. Some of these factors do not change the expected prediction (noise), others do (text form) Hu

et al. (2020a). An example for bad image quality in regards to OCR can be seen in figure 3.1(b).

Performance efficiency addresses time and resource utilization when the model is in use. This does not involve the training phase but the execution or prediction (Siebert et al., 2021). The efficiency refers to low latency needs and to minimizing resource needs such as memory usage or power consumption (Nakamichi et al., 2020; Siebert et al., 2021; Sourvanos and Tsatiris, 2018). This quality is especially important for usage on mobile devices in conjunction with Deep Neural Network (DNN) (Sourvanos and Tsatiris, 2018; Niu et al., 2019).

The first quality often found in research that is not relevant for the use case is fairness. A fair model is free from discrimination bias. For ML this can be a big problem, since discrimination can not only be influenced through explicit programming in terms of the model but also through implicit knowledge from the data (Vogelsang and Borg, 2019). For the use case however no relevance is attached. The model can either recognize the text or it fails the task.

The interpretability of a model helps to justify the output (Ashmore et al., 2021). The interpretability is twofold: explain what the model has learned, explain how a model given the input comes to the output (Vogelsang and Borg, 2019). This can be challenging for two reasons. ML models used can be complex in terms of size and structure (Ashmore et al., 2021). Modular processing pipelines are continuously replaced with end-to-end models Arpteg et al. (2018).

Another quality for a ML model refers to how well a model intended for one task can be reused for another related task. This can be beneficial because transfer learning can speed up the training, thus reducing training cost (Ashmore et al., 2021). Reusability is not relevant in the scope of this work as it targets the training phase of the ML lifecycle.

Chapter 4

Current Research

What is wrong with scene text recognition model comparison

4.1 Different pipeline Frameworks

Baek et al. (2019) Four stages derived from existing STR-Models

- Transformation: normalize input image → Spatial Transformer Network
- Feature extraction: map input image to representation that focuses on relevant attributes, while suppressing irrelevant features
- Sequence Modeling: capture contextual information within sequence of characters
- Prediction: estimate output character sequence

Text Recognition in the Wild: A Survey (Chen et al., 2021)

- pipeline
 - text detection: text localization & text verification
 - (Text Segmentation)
 - Text Recognition
- various stages of OCR:
 - text localization: localize text components, group into candidate text regions with as little background as possible, DNN
 - text verification: verify text candidate regions as text or non-text, filter false-positives, CNN

- text detection: determine whether text is present using localization and verification procedures, basis for end-to-end, can be regression or segmentation based
 - text segmentation: most challenging, includes text line (splitting a region of multiple text lines into subregion of single text lines) and character segmentation (separating text instance into single characters, typically used in earlier approaches)
 - text recognition: translates cropped text instance image into target string sequence, basis for end-to-end, DL encoder-decoder frameworks
 - end-to-end-system: given scene text image → convert all text regions into target string sequences, includes detection, recognition and postprocessing, can be seen as independent subproblems but also joint by sharing information
- text enhancement: recover degraded text, improve text resolution, remove distortions, remove background → reduce difficulty of recognition

Challenges in input preprocessing for mobile OCR applications Sourvanos and Tsatiris (2018)

- Acquisition: obtaining image — digitization, binarization, compression
- Preprocessing: enhancing image quality — noise removal, skew removal, thinning, morphological operations
- Segmentation: separating structural elements — implicit and explicit segmentation
- Feature extraction: generating salient features — geometrical, statistical
- Classification: categorizing individual characters to their respective classes — clustering, neural networks, bayesian models, etc.
- Post-processing: improving and filtering — contextual approaches, multiple classifiers, dictionary based approaches

4.2 Techniques/Modules for Improvement

model pruning: Niu et al. (2019) integer inference: Ignatov et al. (2019)

Chapter 5

Discussion

5.1 Analysis

try to find top 3 – 5

Welche Ergebnisse werden konkret Erwartet -> Taxonomie machen und dann fragen

5.2 Reflection

Challenges DL(Arpteg et al., 2018) Note that actual experiments with models have to be done Problem: different papers have different components → Hardware, Platform, Source Code, Configuration → studies can't really be compared

‘A major challenge in developing DL systems is the difficulties in estimating the results before a system has been trained and tested.’ (Arpteg et al., 2018)

Threats to validity!

5.3 Outlook

What to do next: next steps Data Collection, Data Cleaning, Data Labeling, Model Training, Model Evaluation, Model Deployment, Model Monitoring Watanabe et al. (2019)

Chapter 6

Conclusion

Bibliography

Witold Abramowicz and Rafael Corchuelo, editors. *Business Information Systems: 22nd International Conference, BIS 2019, Seville, Spain, June 26–28, 2019, Proceedings, Part II*, volume 354 of *Lecture Notes in Business Information Processing*. Springer International Publishing, Cham, 2019. ISBN 978-3-030-20481-5 978-3-030-20482-2. doi: 10.1007/978-3-030-20482-2. URL <http://link.springer.com/10.1007/978-3-030-20482-2>.

Anders Arpteg, Björn Brinne, Luka Crnkovic-Friis, and Jan Bosch. Software Engineering Challenges of Deep Learning. In *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 50–59, August 2018. doi: 10.1109/SEAA.2018.00018.

Rob Ashmore, Radu Calinescu, and Colin Paterson. Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges. *ACM Computing Surveys*, 54(5):1–39, June 2021. ISSN 0360-0300, 1557-7341. doi: 10.1145/3453444. URL <https://dl.acm.org/doi/10.1145/3453444>.

Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4714–4722, Seoul, Korea (South), October 2019. IEEE. ISBN 978-1-72814-803-8. doi: 10.1109/ICCV.2019.00481. URL <https://ieeexplore.ieee.org/document/9010273/>.

Lei Chen and Shaobin Li. Improvement Research and Application of Text Recognition Algorithm Based on CRNN. In *Proceedings of the 2018 International Conference on Signal Processing and Machine Learning - SPML '18*, pages 166–170, Shanghai, China, 2018. ACM Press. ISBN 978-1-4503-6605-2. doi: 10.1145/3297067.3297073. URL <http://dl.acm.org/citation.cfm?doid=3297067.3297073>.

Xiaoxue Chen, Lianwen Jin, Yuanzhi Zhu, Canjie Luo, and Tianwei Wang. Text Recognition in the Wild: A Survey. *ACM Computing Surveys*, 54(2):

BIBLIOGRAPHY

- 1–35, April 2021. ISSN 0360-0300, 1557-7341. doi: 10.1145/3440756. URL <https://dl.acm.org/doi/10.1145/3440756>.
- Suman K. Ghosh, Ernest Valveny, and Andrew D. Bagdanov. Visual Attention Models for Scene Text Recognition. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 943–948, November 2017. doi: 10.1109/ICDAR.2017.158. ISSN: 2379-2140.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts, 2016. ISBN 978-0-262-03561-3.
- Boyue Caroline Hu, Rick Salay, Krzysztof Czarnecki, Mona Rahimi, Gehan Se-lim, and Marsha Chechik. Towards Requirements Specification for Machine-learned Perception Based on Human Performance. In *2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pages 48–51, September 2020a. doi: 10.1109/AIRE51212.2020.00014.
- Wenyang Hu, Xiaocong Cai, Jun Hou, Shuai Yi, and Zhiping Lin. GTC: Guided Training of CTC Towards Efficient and Accurate Scene Text Recognition. *arXiv:2002.01276 [cs, eess]*, February 2020b. URL <http://arxiv.org/abs/2002.01276>. arXiv: 2002.01276.
- Andrey Ignatov, Radu Timofte, Andrei Kulik, Seungsoo Yang, Ke Wang, Felix Baum, Max Wu, Lirong Xu, and Luc Van Gool. AI Benchmark: All About Deep Learning on Smartphones in 2019. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3617–3635, October 2019. doi: 10.1109/ICCVW.2019.00447. ISSN: 2473-9944.
- Shangbang Long, Xin He, and Cong Yao. Scene Text Detection and Recognition: The Deep Learning Era. *International Journal of Computer Vision*, 129(1):161–184, January 2021. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-020-01369-0. URL <https://link.springer.com/10.1007/s11263-020-01369-0>.
- Koji Nakamichi, Kyoko Ohashi, Isao Namba, Rieko Yamamoto, Mikio Aoyama, Lisa Joeckel, Julien Siebert, and Jens Heidrich. Requirements-Driven Method to Determine Quality Characteristics and Measurements for Machine Learning Software and Its Evaluation. In *2020 IEEE 28th International Requirements Engineering Conference (RE)*, pages 260–270, August 2020. doi: 10.1109/RE48521.2020.00036. ISSN: 2332-6441.

BIBLIOGRAPHY

- Wei Niu, Xiaolong Ma, Yanzhi Wang, and Bin Ren. 26ms Inference Time for ResNet-50: Towards Real-Time Execution of all DNNs on Smartphone. *arXiv:1905.00571 [cs, stat]*, May 2019. URL <http://arxiv.org/abs/1905.00571>. arXiv: 1905.00571.
- Oyebade K. Oyedotun, Ebenezer O. Olaniyi, and Adnan Khashman. Deep Learning in Character Recognition Considering Pattern Invariance Constraints. *International Journal of Intelligent Systems and Applications*, 7(7): 1–10, June 2015. ISSN 2074904X, 20749058. doi: 10.5815/ijisa.2015.07.01. URL <http://www.mecs-press.org/ijisa/ijisa-v7-n7/v7n7-1.html>.
- Moacir Antonelli Ponti, Leonardo Sampaio Ferraz Ribeiro, Tiago Santana Nazare, Tu Bui, and John Collomosse. Everything You Wanted to Know about Deep Learning for Computer Vision but Were Afraid to Ask. In *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, pages 17–41, October 2017. doi: 10.1109/SIBGRAPI-T.2017.12. ISSN: 2474-0705.
- Ajay Shrestha and Ausif Mahmood. Review of Deep Learning Algorithms and Architectures. *IEEE Access*, 7:53040–53065, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2912200. Conference Name: IEEE Access.
- Julien Siebert, Lisa Joeckel, Jens Heidrich, Adam Trendowicz, Koji Nakamichi, Kyoko Ohashi, Isao Namba, Rieko Yamamoto, and Mikio Aoyama. Construction of a quality model for machine learning systems. *Software Quality Journal*, June 2021. ISSN 0963-9314, 1573-1367. doi: 10.1007/s11219-021-09557-y. URL <https://link.springer.com/10.1007/s11219-021-09557-y>.
- Hannah Snyder. Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104:333–339, November 2019. ISSN 0148-2963. doi: 10.1016/j.jbusres.2019.07.039. URL <http://www.sciencedirect.com/science/article/pii/S0148296319304564>.
- Nikolaos Sourvanos and Georgios Tsatiris. Challenges in Input Preprocessing for Mobile OCR Applications: A Realistic Testing Scenario. In *2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–5, July 2018. doi: 10.1109/IISA.2018.8633688.
- Richard J. Torraco. Writing Integrative Literature Reviews: Guidelines and Examples. *Human Resource Development Review*, 4(3):356–367, September 2005. ISSN 1534-4843. doi: 10.1177/1534484305278283. URL <https://doi.org/10.1177/1534484305278283>. Publisher: SAGE Publications.

BIBLIOGRAPHY

- Andreas Vogelsang and Markus Borg. Requirements Engineering for Machine Learning: Perspectives from Data Scientists. In *2019 IEEE 27th International Requirements Engineering Conference Workshops (REW)*, pages 245–251, September 2019. doi: 10.1109/REW.2019.00050.
- Yasuhiro Watanabe, Hironori Washizaki, Kazunori Sakamoto, Daisuke Saito, Kiyoshi Honda, Naohiko Tsuda, Yoshiaki Fukazawa, and Nobukazu Yoshioka. Preliminary Systematic Literature Review of Machine Learning System Development Process. *arXiv:1910.05528 [cs]*, October 2019. URL <http://arxiv.org/abs/1910.05528>. arXiv: 1910.05528.
- Linjie Xing, Zhi Tian, Weilin Huang, and Matthew Scott. Convolutional Character Networks. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9125–9135, Seoul, Korea (South), October 2019. IEEE. ISBN 978-1-72814-803-8. doi: 10.1109/ICCV.2019.00922. URL <https://ieeexplore.ieee.org/document/9010699/>.
- Xue Yang, Xiaojiang Yang, Jirui Yang, Qi Ming, Wentao Wang, Qi Tian, and Junchi Yan. Learning High-Precision Bounding Box for Rotated Object Detection via Kullback-Leibler Divergence. *arXiv:2106.01883 [cs]*, October 2021. URL <http://arxiv.org/abs/2106.01883>. arXiv: 2106.01883.
- Zhenyao Zhao, Min Jiang, Shihui Guo, Zhenzhong Wang, Fei Chao, and Kay Chen Tan. Improving Deep Learning based Optical Character Recognition via Neural Architecture Search. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–7, July 2020. doi: 10.1109/CEC48606.2020.9185798.
- Didar Zowghi, Zhi Jin, Simone Diniz Junqueira Barbosa, Phoebe Chen, Alfredo Cuzzocrea, Xiaoyong Du, Joaquim Filipe, Orhun Kara, Igor Kotenko, Krishna M. Sivalingam, Dominik Ślęzak, Takashi Washio, and Xiaokang Yang, editors. *Requirements Engineering*, volume 432 of *Communications in Computer and Information Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-662-43609-7 978-3-662-43610-3. doi: 10.1007/978-3-662-43610-3. URL <http://link.springer.com/10.1007/978-3-662-43610-3>.