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

List of Tables

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. 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., 2021b). 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., 2021b). 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., 2021b). 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., 2021b).

1.2 Problem description

Technicians in the field work with different equipement. It is useful to digitize the lables of such equipement, 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 promissing approaches that can be implemented and experimented with have to be identified and analyzed. The reasearch and discussion of techniques from end-to-end STR to dividing the process into text detection and text recognition is centered Chen et al. (2021b) 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 Learing, 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 in order 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 disected 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., 2021b). 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. Concludent to 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 in order 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

2.1 Machine Learning

1. Loss Function / Error Metrics
2. Supervised — Unsupervised / Categorization
3. Optimization techniques: Stochastic-Batch Gradient Descent, GD Momentum, Adam
4. Bias-Variance tradeoff / Overfitting — Underfitting

'The prediction error of a model has three components: irreducible error, which cannot be elim- inated regardless of the algorithm or training methods employed; bias error, due to simplifying assumptions intended to make learning the model easier; and variance error, an estimate of how much the model output would vary if different data were used in the training process. The aim of training is to minimise the bias and variance errors, and therefore the objective functions reflect these errors. The objective functions may also contain simplifying assumptions to aid optimiza- tion, and these assumptions must not be present when assessing model performance [59].'(Ashmore et al., 2021) See (Ashmore et al., 2021) for measures, ROC curve and cost curve Difference in Robusteness vs Performance see (Ashmore et al., 2021) (pretty much bias-variance tradeoff) Robusteness: training set does not include all possible ranges of values -> ability to generalize

See (Seshia et al., 2018) for mathematical notation for ML

Define generalization

The model consists of subcomponents organized in directed acyclc graph building a pipeline (Siebert et al., 2021). This directed acyclic graph depicts everything from processing the images to the extracted information (Siebert et al., 2021).

2.2 Deep Learning

‘ One of the main differences from traditional ma- chine learning (ML) methods is that DL automatically learns how to represent data using multiple layers of abstraction [5], [6]. In traditional ML, a significant amount of work has to be spent on “feature engineering” to build this representation manually, but this process can now be automated to a higher degree. Having an automated and data-driven method for learning how to represent data improves both the performance of the model and reduces requirements for manual feature engineering work [7], [8].’ (Arpteg et al., 2018)

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

    For reusability: see (Ashmore et al., 2021): ‘Convolutional neural networks (CNN) are particularly suited for partial model transfer [59] since the convolutional layers encode features in the input space, whilst the fully connected layers encode reasoning based on those features.’

Deep Learning in Character Recognition Considering Pattern Invariance Constraints (Oyedotun et al., 2015) Deep Learning: neural network architecture of more than a single hidden layer as opposed to shallow networks Features of deep networks: distributed representation of knowledge at each hidden layer, distinct features are extracted by units or neurons in each hidden layer several units can be active concurrently Each layer extracts moredefined/advanced features → hierarchical representation of features

Common problems with training deep learning

- saturating units

- vanishing gradients
- over-fitting & underfitting

Classification of deep learning architectures

- Generative Architectures:
  not deterministic of class patterns that input belong to → sample joint statistical distribution of data
  unsupervised learning: greedy layer-wise pre-training
  Use auto encoders (generative) when a lot unlabelled but not a lot labelled data → generatively train network and then fine tune with labelled
- Discriminative Architectures:
  required to be deterministic of correlation of input data to the classes of patterns therein
  supervised learning
- Hybrid
  combination of discriminative and generative
  generally pre-trained and discriminately fine-tuned for deterministic purposes

Transfer Learning

Factors in Finetuning Deep Model for Object Detection with Long-tail Distribution (Ouyang et al., 2016) finetuning: appraoch dat initializes model parameters for target task from parameters pretrained on another related task

'Transfer learning [183] allows for a model learned in one domain to be exploited in a second domain, as long as the do- mains are similar enough so that features learned in the source domain are applicable to the target domain. Where this is the case, all or part of a model may be transferred to reduce the training cost.' Ashmore et al. (2021)

'Convolutional neural networks (CNN) are particularly suited for partial model transfer [59] since the convolutional layers encode features in the input space, whilst the fully connected layers encode reasoning based on those features.' Ashmore et al. (2021)

Convolutional Neural Network

Comparative analysis of deep learning image detection algorithms (Srivastava et al., 2021) These layers apply filters to extract patterns from images.

The filter moves over the image to generates the output. Different filters recognize different patterns. Initial layers have filters to recognize simple patterns. They become more complex through the layers over time as follows:

Review of Deep Learning Algorithms and Architectures (Shrestha and Mahmood, 2019) Def Neural Network:

- Machine Learning technique that consists of processing units organized in input, hidden and output layers
- the nodes or units in each layer are connected to nodes in adjacent layers
- each connection has weight value
- inputs are multiplied by weight and summed up at each unit
- the sum is used with an activation function (e.g. ReLU, Sigmoid, Tanh, SoftPlus)

2.3 Optical Character Recognition

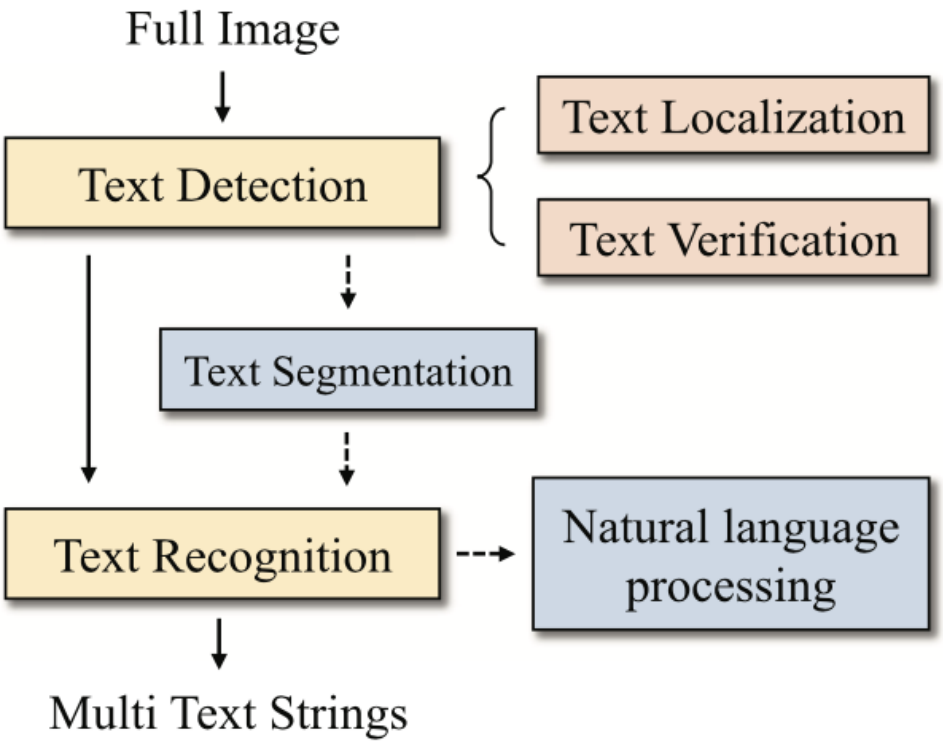
Deep Learning based OCR (Zhao et al., 2020) What is OCR: process of converting images of typed, handwritten or printed text into machine-encoded one includes two sub frameworks: text detection and text recognition (based on position coordinates) **End-To-End also possible** Process can include image processing!!!

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

- 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 detectoin, recognition and postprocessing, can be seen as indipendent subproblems but also joint by sharing information

- text enhancement: recover degraded text, improve text resolution, remove distortions, remove background → reduce difficulty of recognition



OCR-phases Srivastava et al. (2021)

Phase	Description	Approaches
Acquisition	Obtaining the 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: corners, edges, etc. Statistical: moments, etc.
Classification	Categorizing individual characters to their respective classes	Clustering (e.g. K-NN), neural networks, bayesian models, etc.
Post-processing	Improving and filtering results	Contextual approaches, multiple classifiers, dictionary based approaches

see Sourvanos and Tsatiris (2018) for mobile issues for OCR

no source grid: divides image into parts → each part has own bounding boxes bounding boxes: regressor for box, each bounding box is assigned an anchor box (respective to grid cell) anchor boxes: default 'shape' for bounding box

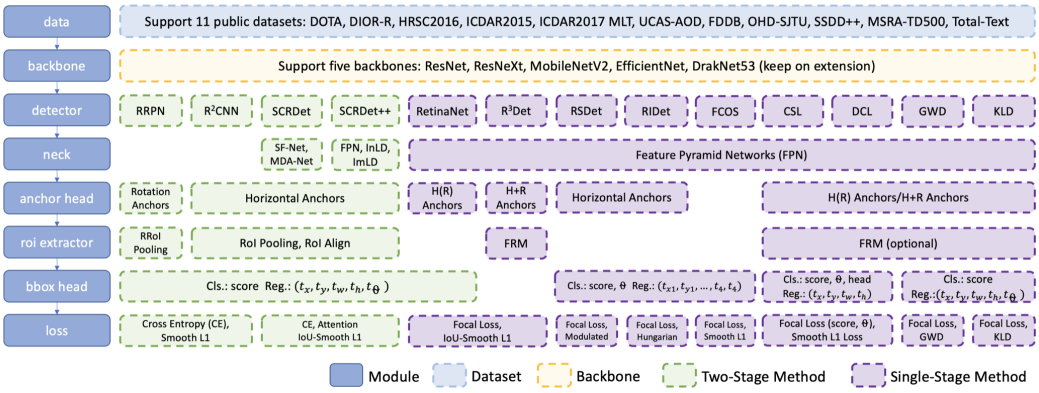
bounding boxes different stages of convolution / 2-d size → different object size to detect

Text detection

subfield of object detection (e.g. YOLOv4 can be used for text)

Detect position coordinates containing text in input image Text detection more challenging

From Github (noa)



Two object detection methods — CNN-based

- Region-based

views detection problem as classification problem
CNN to extract deep features of proposals by selective search → Use SVM to classify with features
e.g. R-CNN

- single 'look' extract feature maps on entire image
  directly regress bounding boxes on feature maps
  e.g. YOLO — You Only Look Once, SSD — Single Shot Detection

Non CNN-based: DETR

Comparison Object Detection basic algos

Comparative analysis of deep learning image detection algorithms (Srivastava et al., 2021) YOLO-V3 outperforms SSD and Faster R-CNN
VGG-16 widely used feature generating architecture

Faster-RCNN

A deeper look at how Faster-RCNN works (Goswami, 2018) composed of 3 neural nets:

- Feature Network: pre-trained image classification netork → generate good features
- Region Proposal Network:
  - NN with 3 conv layers
  - one layer splits up network to: classification and bounding box regression
  - bounding box regression → bounding boxes are region of interes (ROI) that might contain an object
- Detection Network: take input from previous nets, generate final class and bounding box, 4 fully connected, 2 stacked common layers shared by classification and bounding box regression layer

Deep Learning in Character Recognition Considering Pattern Invariance Constraints (Oyedotun et al., 2015) Neural networks can learn features of task on which they are designed and trained Neural networks better than other approaches (e.g. template matching, syntactic analysis) → NNs can learn and adapt to moderate variations (e.g. translation, rotation, scaling, noisy patterns)

Text Recognition

Recognize text based on position coordinates

character based or word based

Visual attention models for scene text recognition (Ghosh et al., 2017) Divided into word detection (generate bounding boxes) and word recognition word recognition can be divided into dictionary-based methods and unconstrained methods

End-To-End

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.

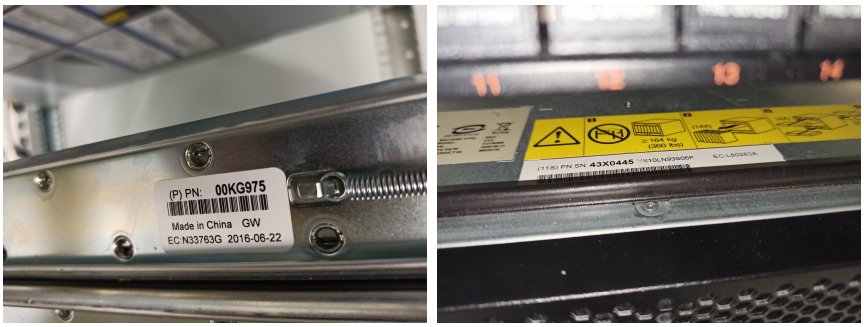
Requirements for a software system that involves ML and thus DL differs from the traditional approach. The data-driven software components are not entirely defined by the programmer but are influenced by data. The system acts with dependency on the test data (Siebert et al., 2021). This poses a challenge 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.

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/infrastrucure (Nakamichi et al., 2020; Siebert et al., 2021). The article Siebert et al. (2021) differenciates 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. 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). This applies to studies that create an overview such as Chen et al. (2021b); 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.

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). For this thesis, the basic use case is as follows: A technician takes a photo of a device label with his smart phone. The resulting image contains printed textual information which must be extraced 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,



(a) Positive example (b) Negative example

Figure 3.1: Examples for label images

spacing and structure carries semantic information which can be important for later processing in the scope of a business process (Chen et al., 2021b). The goal is to extract the text and preserve semantics from 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. The labels can contain arbitrary alphanumeric 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) 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.

Alphanumeric recognition	Recognize alphanumeric strings such as serial numbers
Semantics retention	Retain semantics given implicitly be space, strucure and rotation of text in labels

Table 3.1: Qualities specific to use case — exclusion criterias

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), 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	Not relevant
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. For this thesis this applies to STR models.

'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). The 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. (2021b); Long et al. (2021). Performance is usually measured with a test dataset that is independed 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., 2021b). 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., 2021b). 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 is 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 continusously 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 relavant in the scope of this work as it targets the training phase of the ML lifecycle.

Chapter 4

Current Research

no transformers → self-attention mechanism is too computationally expensive???

model-pruning → remove connections for better performance

'The great advances that have been made in fields such as computer vision and speech recognition, have been accom- plished by replacing a modular processing pipeline with large neural networks that are trained end-to-end [37]. In essence, transparency is traded for accuracy. This is an unavoidable reality.'(Arpteg et al., 2018)

include Pipeline differences

Two models that can be used in conjunction **detection** (Beom, 2021b)
uses RetinaNet structure (Lin et al., 2018)
applies techniques from textboxes++ (Liao et al., 2018)

character recognition (Beom, 2021a)
needs cropped text area as input
uses CRNN (Shi et al., 2015) → end-to-end learning, LSTM fir arbitrary length of input and output, no need to apply detection and cropping to each single character

Open Source OCR engine (Smith, 2007)

- uses Deep Learning (found c++ code for layers in repo)
- Processing in step-by-step pipeline, some unusual stages
  1. Line and Word finding
  1.1. Line finding
  1.2. Baseline Fitting
  1.3. Fixed Pitch Detection and Chopping
  1.4. Proportional Word Finding
  2. Word Recognition
  2.1 Chopping Joined Characters

2.2 Accociating Broken Characters
3. Static Character Classifier
3.1 Features
3.2 Classification
3.3 Training Data
4. Linguistic Analysis
5. Adaptive Classifier

Performs poorly with unstructured text with significant noise
An Efficient and Accurate Scene Text Detector (Zhou et al., 2017)
SOFT: Softmax-free Transformer with Linear Complexity (Lu et al., 2021)
Generative Pretraining from Pixels (Chen et al., 2021a)

- unsupervised representation learning (approach transfered from NLP)
- training of sequence Transformer to auto-regressively predict pixels without incorporating knowledge of 2D input structure
- Active part: GPT-2 scale model learns image representations and performs extremely well even when compared to supervised models

Learning High-Precision Bounding Box for Rotated Object Detection via Kullback-Leibler Divergence (Yang et al., 2021)

- Deductive approach to rotated object detection
- box is 'translated' to 2D-Gaussian → KLD with prediction and true gaussian as Loss
- LIMIT: cannot be directly applied to quadrilateral detection

DP-SSL: Towards Robust Semi-supervised Learning with A Few Labeled Samples (Xu et al., 2021)

- Semi-supervised learning:
  - provides way to leverage unlabeled data by pseudo labels
  - performs poorly and unstable when size of labeled data is very small (low quality of pseudo labels)
- Data programming:
  - paradigm for the programmatic creation of training sets
  - existing methods rely on human experts to provide initial labeling functions (LF)

- DP-SSL
  - multiple-choice learning (MCL) based approach to automatically generate labeling functions
  - scheme to generate probabilistic labels for unlabeled data

which aspects to compare? quantitative, qualitative

Chapter 5

Discussion

5.1 Analysis

try to find top $3-5$

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

yangxue0827/RotationDetection: This is a tensorflow-based rotation detection benchmark, also called AlphaRotate. URL https://github.com/yangxue0827/RotationDetection.

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.

Beom. CRNN (CNN+RNN), September 2021a. URL https://github.com/qjadud1994/CRNN-Keras. original-date: 2018-01-14T07:52:25Z.

Beom. Text Detector for OCR, August 2021b. URL https://github.com/qjadud1994/Text_Detector. original-date: 2019-03-12T05:11:06Z.

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.

Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative Pretraining from Pixels. page 12, 2021a.

Xiaoxue Chen, Lianwen Jin, Yuanzhi Zhu, Canjie Luo, and Tianwei Wang. Text Recognition in the Wild: A Survey. *ACM Computing Surveys*, 54(2): 1–35, April 2021b. 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.

Subrata Goswami. A deeper look at how Faster-RCNN works, July 2018. URL `https://whatdhack.medium.com/a-deeper-look-at-how-faster-rcnn-works-84081284e1cd`.

Boyue Caroline Hu, Rick Salay, Krzysztof Czarnecki, Mona Rahimi, Gehan Selim, 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.

Minghui Liao, Baoguang Shi, and Xiang Bai. TextBoxes++: A Single-Shot Oriented Scene Text Detector. *IEEE Transactions on Image Processing*, 27(8):3676–3690, August 2018. ISSN 1057-7149, 1941-0042. doi: 10.1109/TIP.2018.2825107. URL `http://arxiv.org/abs/1801.02765`. arXiv: 1801.02765.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. *arXiv:1708.02002 [cs]*, February 2018. URL `http://arxiv.org/abs/1708.02002`. arXiv: 1708.02002.

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.

Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang. SOFT: Softmax-free Transformer with Linear Complexity. *arXiv:2110.11945 [cs]*, October 2021. URL http://arxiv.org/abs/2110.11945. arXiv: 2110.11945.

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.

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.

Wanli Ouyang, Xiaogang Wang, Cong Zhang, and Xiaokang Yang. Factors in Finetuning Deep Model for Object Detection With Long-Tail Distribution. pages 864–873, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Ouyang_Factors_in_Finetuning_CVPR_2016_paper.html.

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.

Sanjit A. Seshia, Ankush Desai, Tommaso Dreossi, Daniel J. Fremont, Shromona Ghosh, Edward Kim, Sumukh Shivakumar, Marcell Vazquez-Chanlatte, and Xiangyu Yue. Formal Specification for Deep Neural Networks. In Shuvendu K. Lahiri and Chao Wang, editors, *Automated Technology for Verification and Analysis*, Lecture Notes in Computer Science, pages 20–34, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01090-4. doi: 10.1007/978-3-030-01090-4_2.

Baoguang Shi, Xiang Bai, and Cong Yao. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. *arXiv:1507.05717 [cs]*, July 2015. URL `http://arxiv.org/abs/1507.05717`. arXiv: 1507.05717.

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`.

R. Smith. An Overview of the Tesseract OCR Engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633, September 2007. doi: 10.1109/ICDAR.2007.4376991. ISSN: 2379-2140.

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.

Shrey Srivastava, Amit Vishvas Divekar, Chandu Anilkumar, Ishika Naik, Ved Kulkarni, and V. Pattabiraman. Comparative analysis of deep learning image detection algorithms. *Journal of Big Data*, 8(1):66, December 2021. ISSN 2196-1115. doi: 10.1186/

s40537-021-00434-w. URL https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00434-w.

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.

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/.

Yi Xu, Jiandong Ding, Lu Zhang, and Shuigeng Zhou. DP-SSL: Towards Robust Semi-supervised Learning with A Few Labeled Samples. *arXiv:2110.13740 [cs]*, October 2021. URL http://arxiv.org/abs/2110.13740. arXiv: 2110.13740.

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.

Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. EAST: An Efficient and Accurate Scene Text Detec-

tor. *arXiv:1704.03155 [cs]*, July 2017. URL http://arxiv.org/abs/1704.03155. arXiv: 1704.03155.

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.