Bachelor Thesis
in Information Systems and Management

Label Extraction from Image via 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 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.

Contents

Chapter 1

Introduction

1.1 Motivation

Optical Character Recognition is the concept of extracting typed, handwritten or printed text of an image. Techniques for this concept have improved a lot due to the advances in the field of deep learning [ZJG+20]. Deep Learning is a technology based on Artificial Neural Networks where data is processed in multiple layers to extract complex features and solve a given problem [SM19]. Deep Learning has only caught on in the recent years as the big computational cost has been met by the improvement in computer hardware [PRN+17]. Finding the right solution in the space of deep learning and applying these new capabilities to the use case of extracting information of equipment labels is the focus of this thesis.

1.2 Problem description

The central part of the bachelor thesis is finding the right apprach for the extraction of textual information of of images with equipment labels. This includes defining functional requirements such as detecting rotated text but also non functional requirements such as given computational power of mobile devices that are to use the solution. These requirements define properties that a solution must have in order to be classified as viable. Thus the discussion of techniques from end-to-end OCR to dividing the process into text detection and text recognition is centered around the requirements which are given by the problem. The research ranges from established solutions for similar problems to current research in the field.

Following aspects such as deploying and maintaining a solution in a production environment shall not be subject of this thesis.

1.3 Methodology

The goal of this work is to implement and train a Deep Learning model to read in labels from photos. The emerging artifact can be used to solve the problem detailed in 1.2. The expository instantiation is helpful to gain more understanding the artifact as it is common in design science. In particular this is justificatory knowledge on the design on the Deep Learning model and Machine Learning way of approaching problems. This is important in order to apply it and to optimize existing research to the specific problem.

The methodology is based on action research [JP21]. It constists of a cycle of five phases: Diagnosis, Planning, Intervention, Evaluation, Reflection. The first cycle will entail an exploratory data analysis which corresponds to the Diagnosis part. Here it is important to recognize main characteristics of the images and to find outliers and other potential problems [Cox17]. The research is then extended to existing practical solutions for similar practical problems as well as proposed architectures from academic research. Theoretical knowledge about the models as well as practical information about results for similar problems contribute to the discussion about which approach is the most promissing. Combining architectures is also a viable possibility to solve the given problem. This concludes the Planning phase and will lead to a model exaptation that evolves to be the artifact at the center of this thesis. The next step is implementing and training the chosen approach which. Evaluation for of the current model follows. Storing and analyzing results of training and cross validation as well as visualizing the training progress is an important part of this. In the Reflection stage it is decided whether a new cycle should be carried out.

From the second cycle on the first three phases change as there already is a model that is to be improved. This time the Diagnosis phase entails asking questions about the existing model: What worked? Why did it work/not work? What needs to change? Changes are planned and implemented accordingly. The Evaluation and Reflection phases are not changing in the second cycle thus closing the loop. The incremental adjustments to the model are made in order to improve the accuracy. This includes possibly adjusting the architecture, hyperparameter tuning and preprocessing approaches like image compression.

1.4 Expected results and outlook

The research into the theoretical foundation of Deep Learning and into possible approaches leads to a strong understanding of the underlying technology. This is helpful to produce a comparison of approaches that is based on theoretical as well as practical knowledge. The goal is to find out which approach work best for the chosen practical problem and why that is the case. Implementation and training of the most promissing one is yielding the artifact this work revolves around. The process of optimization not only improves the solution to the problem (see 1.2) but is also used to learn more about the implemented approach.

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

2.2 Deep Learning

1. ANN / MLP
   - Architecture → Input, Hidden, Output
   - Feedforward
   - Optimization → Backpropagation, SGD, ADAM, ...
2. Regularization
3. important architectures
   - CNN
   - RNN
   - Specific foundation architectures for relevant approaches

Deep Learning in Character Recognition Considering Pattern Invariance Constraints [OOK15] 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

Ohne Quelle Generative and Convolution for 'feature generation' → which one is best?

2.2.1 Generative Architectures

Deep Learning in Character Recognition Considering Pattern Invariance Constraints [OOK15] Stacked Denoizing Auto Encoder lowest error rate on translation Deep Belief Network lowest error rate on rotation, scale, low noise

2.2.1.1 Auto-Encoder

Auto-Encoder not always seen as generative! **Deep Learning in Character Recognition Considering Pattern Invariance Constraints** [OOK15] denoizing

- Generative
- Learn underlying features during training
- Single layer, feedfoward network
  - input and output neurons in equal amount
  - number of hidden units is smaller
  - endcode — input → hidden; decode — hidden → output
- Unsupervised (see source for details)
- auto encoders can be stacked on one another → more distributed and hierarchical representation → Stacked Auto Encoders

2.2.1.2 Deep belief network

Deep Learning in Character Recognition Considering Pattern Invariance Constraints [OOK15]

- Generative
- graphical and probabilistic, directed acyclic graph composed of stochastic variables
- combination of Sigmoid Belief Network (aka Bayesian network) and a Restricted Blotzman Machine

2.2.2 Convolutional Neural Network

Comparative analysis of deep learning image detection algorithms [SDA+21] 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 [SM19] 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 Opical Character Recognition

Deep Learning based OCR [ZJG+20] 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!!!

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

2.3.1 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

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 [SDA+21] YOLO-V3 outperforms SSD and Faster R-CNN

VGG-16 widely used feature generating architecture

Faster-RCNN

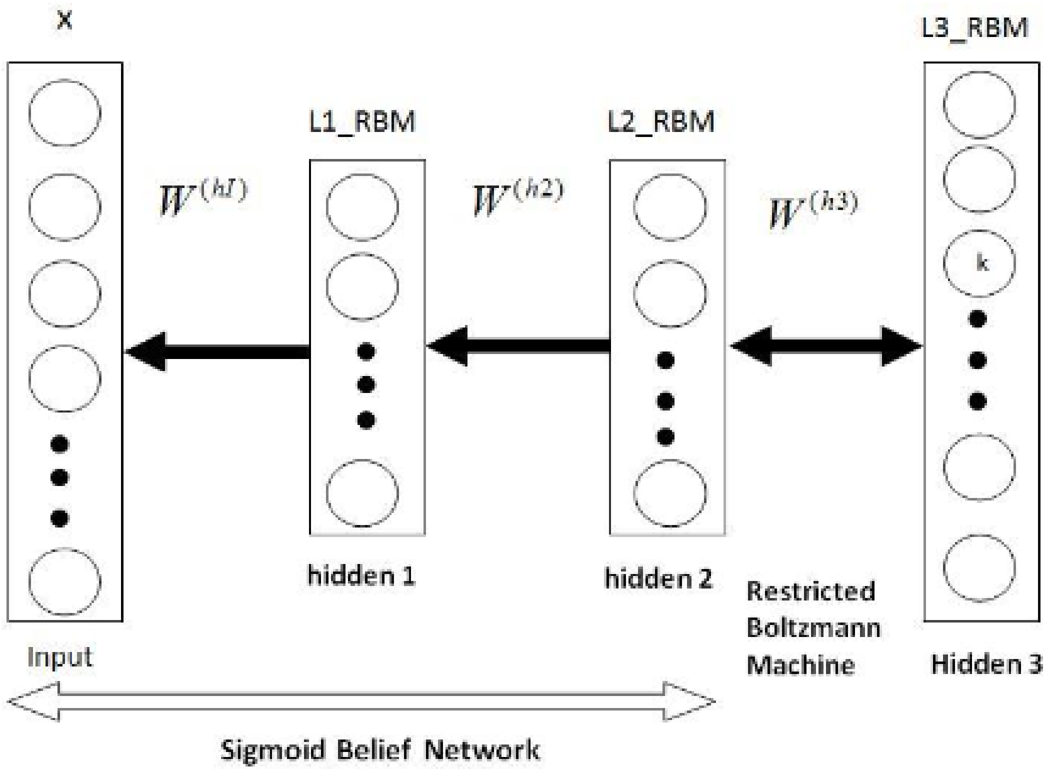
A deeper look at how Faster-RCNN works [Gos18] composed of 3 neural nets:

- 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 [OOK15] 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)

2.3.2 Character recognition

Recognize text based on position coordinates



2.3.3 End-To-End

Chapter 3

Exploratory Data Analysis

When determining whether automisation is an improvement four aspects have to be examined. These are time, costs, quality and flexibility. The aspects build a quadrangle that is based on the optimizing trade-off between the factors [DLRMR13].

Without software supporting the task of reading the name of the picture and typing it into the system, can take long seconds, whereas a trained Deep Learning model could complete the task in a mere instant. Therefor automisation via Deep Learning should improve the efficiency of the process when compared to manually reading and typing the information off the image.

Training costs for a Deep Learning model are very high due to the computing intensive backpropagation algorithm that tunes the network to the data. But the usage cost is low. For manual labor the opposite is the case as training a person to type in a label is done quickly and labor costs are high in comparison to the expenses for running the model.

Both Deep Learning models and human labor are not 100% accurate. It is human to make mistakes and because Deep Learning is trained only trained on a specific set of data it makes sense that not all predictions can be correct as there can always be outliers in the data. The question is whether the model can be as accurate or even better than its human counterpart. This is especially interesting when it is applied in the real world where it might have to do good in subpar situations. An example is bad image quality.

Flexibility is concerned with how well a process can adjust to changing requirements. A set of new equipment names that have to be included can pose a problem to a Deep Learning model because it is not trained for the new data. A human on the other hand should not have any problems in this regard.

The main concern for the solution's efficacy is whether it is accurate enough. Therefor this work focuses on this aspect in particular.

Chapter 4

System Design

4.1 Approach comparison

include Pipeline differences

4.1.1 Approach Research

GitHub implementation

Two models that can be used in conjunction

detection [Beo21b]

uses RetinaNet structure [LGG+18]

applies techniques from textboxes++ [LSB18]

character recognition [Beo21a]

needs cropped text area as input

uses CRNN [SBY15] → end-to-end learning, LSTM fir arbitrary length of input and output, no need to apply detection and cropping to each single character

Tesseract

Open Source OCR engine [Smi07]

- 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

Faster-RCNN

A deeper look at how Faster-RCNN works [Gos18] composed of 3 neural nets:

- Feature Network: pre-trained image classification netork $\rightarrow$ 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 $\rightarrow$ 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

current research

An Efficient and Accurate Scene Text Detector [ZYW+17]
SOFT: Softmax-free Transformer with Linear Complexity [LYZ+21]
Generative Pretraining from Pixels [CRC+]

- 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 [YYY$^+$21]

- Deductive approach to rotated object detection
- box is ‘translated’ to 2D-Gaussian $\rightarrow$ 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 [XDZZ21]

- 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

4.1.2 Comparison

4.2 Approach selection

Chapter 5

Implementation

5.1 Software and Tools

5.2 Preprocessing

5.3 Prototype

5.4 Optimizations

Chapter 6

Discussion

6.1 Results

6.2 Method reflection

6.3 Future and follow up research

Chapter 7

Conclusion

Appendix A

References

List of Figures

List of Tables

Bibliography

[Beo21a] Beom. *CRNN (CNN+RNN)*, September 2021. original-date: 2018-01-14T07:52:25Z.

[Beo21b] Beom. *Text Detector for OCR*, August 2021. original-date: 2019-03-12T05:11:06Z.

[Cox17] Victoria Cox. *Translating Statistics to Make Decisions: A Guide for the Non-Statistician*. Apress, Berkeley, CA, 2017.

[CRC+] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. *Generative Pretraining from Pixels*. page 12.

[DLRMR13] Marlon Dumas, Marcello La Rosa, Jan Mendling, and Hajo A. Reijers. *Fundamentals of Business Process Management*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[Gos18] Subrata Goswami. *A deeper look at how Faster-RCNN works*, July 2018.

[JP21] Paul Johannesson and Erik Perjons. *An Introduction to Design Science*. Springer International Publishing, Cham, 2021.

[LGG+18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. *Focal Loss for Dense Object Detection*. arXiv:1708.02002 [cs], February 2018. arXiv: 1708.02002.

[LSB18] 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. arXiv: 1801.02765.

[LYZ+21] 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. arXiv: 2110.11945.

[OOK15] 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.

[PRN+17] 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. ISSN: 2474-0705.

[SBY15] 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. arXiv: 1507.05717.

[SDA+21] 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.

[SM19] Ajay Shrestha and Ausif Mahmood. *Review of Deep Learning Algorithms and Architectures.* IEEE Access, 7:53040–53065, 2019. Conference Name: IEEE Access.

[Smi07] 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. ISSN: 2379-2140.

[XDZZ21] 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. arXiv: 2110.13740.

[YYY+21] 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. arXiv: 2106.01883.

[ZJG+20] 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.

[ZYW+17] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. *EAST: An Efficient and Accurate Scene Text Detector.* arXiv:1704.03155 [cs], July 2017. arXiv: 1704.03155.

Appendix B

Code