Process Automation and Quality Assurance in Computer Vision for Real World Applications

Inaugural-Dissertation

zur

Erlangung des Doktorgrades der
Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Kirill Bogomasov

aus Charkiw

Düsseldorf, Januar 2023

aus dem Institut für Informatik
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Berichterstatter:

1. Prof. Dr. Stefan Conrad

2. Prof. Dr. Timo Dickscheid

Tag der mündlichen Prüfung: 13.07.2023

Ich versichere an Eides Statt, dass die Dissertation von mir selbstständig und ohne unzulässige fremde Hilfe unter Beachtung der *Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf* erstellt worden ist.

Die hier vorgelegte Dissertation habe ich eigenständig und ohne unerlaubte Hilfe angefertigt. Die Dissertation wurde in der vorgelegten oder in ähnlicher Form noch bei keiner anderen Institution eingereicht. Ich habe bisher keine erfolglosen Promotionsversuche unternommen.

Düsseldorf, den 10.01.2023 Kirill Bogomasov

Dedicated to my family

Acknowledgements

First and most of all, I would like to thank my supervisor and first reviewer, Prof. Dr. Stefan Conrad, for giving me the opportunity to become part of his team. In all the years that I worked on my dissertation, he gave me a lot of space for the implementation of my own ideas in research and teaching. During the entire time, he was always open to my questions and provided valuable feedback. Moreover, he accompanied me to meetings and supported with his competent opinion in negotiations and discussions. I also would like to acknowledge Prof. Dr. Gunnar Klau for being interested in my research and providing his competent feedback when I needed it the most. Furthermore, I would like to thank Prof. Dr. Timo Dickscheid for reviewing this work. I greatly appreciate his expertise in the field of Computer Vision.

I would also like to thank the people without whom this work would not have been possible. I thank Dr. Christian Rubbert and Dr. Lars Schimmöller for their support with data and informative discussions during the research.

I want to express my gratitude to Malte Eckhardt, who also supported me with data as well as the great feedback as part of the cooperation with PENNY Markt GmbH and REWE Systems GmbH. Additionally, I would like to express my gratitude to his team, that showed great interest in the implementation of the common project.

This work would not have been possible without technical and administrative support. My special thanks go to our system administrator Guido Königstein and our secretary Sabine Freese.

Of course, I would also like to thank my research colleagues for constructive discussions and a pleasant, friendly atmosphere, including Dr. Alexander Askinadze, Dr. Christian Bock, Mahn Khoi Doung, Thomas Germer, Dr. Gerhard Klassen, Dr. Martha Krakowski, Dr. Matthias Liebeck, Julia Romberg and Dr. Michael Singhof. I would particularly like to thank Daniel Braun and Philipp Grawe, who I'm proud to call my friends. I enjoyed common work and research the most. You made this time unforgettable.

Finally, I would like to thank my parents, Svitlana and Alexander, and my sister Alina for their support, my friends for their patience and my partner Sabrina for both. I am infinitely grateful to have you in my life.

Abstract

Proceeding digitization leads to a growing amount of computationally processible and storable information. In fact, at no time the amount of data was as huge as it is today during the Digital Revolution and far beyond the volume, which is manually manageable. Visual data is one of the most important among all data types because of its unsophisticated interpretability for humans and high information density. However, there is also a major drawback. Working on visual data is computationally expensive. The progressive development of the hardware in recent years, especially the graphic units, as well as its affordability allow complex analysis in terms of visual inspection. Consequentially, automation and quality assurance of processes get feasible based on the image data interpretation.

This work deals with the following question: **How can image data be computationally processed, analyzed and interpreted to allow automation of a conventionally manual process and assure its quality?** To answer this question, we consider three independent image source areas: medical imaging, underwater photographs and natural images. The showcased selection presents its strengths in different image characteristics representative for particular data. The medical imaging, consisting of CT- and MRI-recordings, is presented as 3D images and therefore contain depth information while being standardized to specific scanners. The underwater photographs compose a large-scale collection of images belonging to a coral reef monitoring project around the world and naturally differing in quality regarding sharpness and color balance. The natural images are taken from an ongoing sales process by rather primitive hardware and therefore contain several differences regarding visual obstacles and lower resolution as well as image quality in general.

Due to different data characteristics, diverse approaches are required. During examining the research question, we tackle a series of challenging *classification* and *object detection* tasks. As a result, in this thesis we present multiple novel *machine learning* and *deep learning* algorithms. Concerning medical imaging, we investigate severity scoring for lung tuberculosis in CT-recording and compare traditional feature engineering and deep learning. Moreover, we introduce a unique algorithm for the orientation estimation of prostate cancer patients in MRI-recordings. Apart from the medical application field, we present two competitive approaches for maritime inventory monitoring. Furthermore, we propose a novel approach for efficient counting and classification in retail applications. All research results presented in this work can be assigned to the field of AI and have the main focus on *computer vision*.

Overall, our approaches on various real-world data show convincing results regarding the main research question and show the potentials as well as limitations in applying com-

puter vision to solve quality assurance of processes and automation tasks. Furthermore, we propose several algorithms which are at least competitive to the state-state-of-the-art or even are state-of-the-art in the field of deep learning and computer vision.

Zusammenfassung

Die fortschreitende Digitalisierung führt zu einer wachsenden Menge an maschinell verarbeitbaren und speicherbaren Informationen. Die tatsächliche Datenmenge war zu keiner Zeit so groß wie heute während der digitalen Revolution. Die Tendenz bleibt steigend. Dabei ist deren Menge bereits heute größer als jene, die manuell verarbeitbar wäre. Unter allen dabei entstandenen Datenarten gehören Bilddaten aufgrund ihrer vorteilhaften Eigenschaften wie zum Beispiel der einfachen Interpretierbarkeit für Menschen und der hohen Informationsdichte, zu den wichtigsten aller Datenarten. Dennoch sind gerade diese sehr rechenlastig und erfordern für die Verarbeitung entsprechende Hardware. Die fortschreitende Entwicklung dieser in den letzten Jahren, insbesondere der grafischen Einheiten sowie ihre Erschwinglichkeit, nicht nur für große Rechenzentren, sondern auch für einfache Nutzende, schaffen Grundlage für komplexe visuelle Analysen und Inspektionen. Damit werden Automatisierung und Qualitätssicherung von Prozessen, die auf der Bilddateninterpretation basieren, realisierbar. Folglich beschäftigt sich diese Arbeit mit folgender Frage: **Wie lassen sich Bilddaten rechnergestützt verarbeiten, analysieren und interpretieren, um eine Automatisierung von traditionell manuell durchgeführten Prozessen und deren Qualität zu sichern?** Um diese Frage beantworten zu können, werden drei unabhängige Bildquellenbereiche betrachtet: medizinische Bilder, Unterwasseraufnahmen und gewöhnliche 2D-Aufnahmen. Die Auswahl weißt unterschiedliche charakteristische Bildeigenschaften vor, die für diese Datentypen repräsentativ sind. Die medizinische Datengrundlage setzt sich aus CT- und MRT-Bildern, die als 3D Aufnahmen vorliegen, zusammen und besitzt deshalb eine Tiefeninformation, die für die entsprechenden Aufnahmegeräte (Scanner) standardisiert ist. Die Unterwasserbilder bilden eine weitere Kategorie. Der vorliegende Datensatz setzt sich zusammen aus hochauflösenden Bildern von Korallenriffen, die auf der ganzen Welt im Rahmen eines Bestandsüberwachungsprojekts gesammelt wurden und sich dementsprechend in der Qualität hinsichtlich der Schärfe und der Farbbalance stark unterschieden. Die Sammlung der gewöhnlichen 2D-Bilder stammt aus dem laufenden Verkaufsprozess unterschiedlicher Supermärkte. Die Bilder wurden durch eine kostengünstige Hardware aufgenommen und verfügen über eine geringere Auflösung. Zusätzlich besteht eine Reihe von Schwierigkeiten in Bezug auf visuelle Hindernisse. Aufgrund unterschiedlicher Eigenschaften, die die verschiedenen Bildtypen mit sich bringen, sind in Bezug auf die Fragestellung unterschiedliche Herangehensweisen erforderlich. Während der Untersuchung der Forschungsfrage gehen wir eine Reihe herausfordernder Klassifikations- und Objekterkennungsaufgaben unter Berücksichtigung dieser Nuancen an. Die Lösungen dieser Aufgaben stellen wir in dieser Arbeit in Form von mehreren neuartigen Maschine Learning und Deep Learning Algorithmen vor. Auf den medizinischen Daten der Radiologie, wurde der Frage nachgegangen, wie sich automatisch der Schweregrad der Lungentuberkulose anhand von CT-Bildern ermitteln lässt. Dabei wurde traditionelles Feature Engineering und Deep Learning gegenübergestellt. Darüber hinaus führen wir einen einzigartigen Algorithmus zur Bestimmung der Orientierung von MRT-Aufnahmen von an Prostatakrebs erkrankten Patienten ein. Neben dem medizinischen Anwendungsbereich werden zwei State-of-the-Art Ansätze für die Bestandsüberwachung von Korallen vorgestellt. Außerdem präsentieren wir einen neuartigen Ansatz für effizientes Zählen und Klassifizieren für den Einsatz im Einzelhandel. Alle in dieser Arbeit vorgestellten Forschungsergebnisse lassen sich dem Bereich der KI zuordnen und haben den Schwerpunkt Computer Vision. Insgesamt zeigen unsere Ansätze auf den verschiedenen Datensätzen aus der realen Welt im Hinblick auf die Hauptforschungsfrage überzeugende Ergebnisse und decken sowohl die Potenziale als auch die Grenzen der Verwendung von Computer Vision zu Zwecken der Qualitätssicherung und der Automatisierung von Prozessen auf. Darüber hinaus stellen wir mehrere Algorithmen vor, die entweder konkurrenzfähig zu den aktuellen State-of-the-Art Verfahren im Bereich von Deep Learning und Computer Vision sind oder sogar selbst State-of-the-Art Verfahren bilden.

Contents

1 Introduction

"Just as electricity transformed almost everything 100 years ago, today I actually have a hard time thinking of an industry that I don't think AI (Artificial Intelligence) will transform in the next several years"

— Andrew Ng (A. Ng, 2017)

The *Digital Revolution* (also called the Third Industrial Revolution) has lasted since its beginning in the middle of the 20th century continuously. One can expect that the trend will continue and that we will be confronted with its development, as well as concomitant chances in the future. One of its largest fields is the *Digitization*, which means the process of transforming information into a digital format[1]. Among a large number of data types that this process made available, *digital image* is one of the most fundamental. Primarily, its most important advantage is the ability to picture the current as-is state, as it is perceived by humans. Coupled with the computational-friendly raster representation of images along with the possibility of lossless storage, the field of *digital image processing* (DIP) was created. According to Castleman (1996), this topic concerns the manipulation of images by computers. Furthermore, DIP involves diverse essential processes. These processes use images as input and provide either images or attributes as output. The first case includes methods such as image enhancement, image acquisition, image restoration, colour image processing, compression and wavelets, among others. The second case includes methods such as morphological processing, segmentation, description, representation and recognition. The processed input information can be passed on for further operations such as *digital image analysis*, which addresses the description and recognition of the image content. A methodology that simulates human vision and is therefore called *computer vision* (CV)(Pitas, 2000). Apart from simple digital imaging, complex imaging techniques such as MRI and CT provide an even higher information density and open up completely new possibilities for digital analysis. Fast capturing, the availability of low-priced hardware as well as the ability to store and share digital images lead to an increasing

[1] https://www.collinsdictionary.com/dictionary/english/digitize

volume of data and provide a reliable foundation for a set of technologies that can be summarized under the heading "Industry 4.0". The term describes mostly IT-driven changes in primarily but not exclusively business-related applications. For instance, high innovation capability, faster decision-making procedures and resource efficiency are some of its targets. Therefore increasing automation and inbound product and quality control is indispensable (Lasi et al., 2014).

For visual tasks, CV is the driving force behind computer-assisted diagnosis, process monitoring and quality assurance, as well as automation. Especially the latest achievements in *object detection* (OD) enable the localization of objects in larger numbers and with an ever-increasing variety of objects (Zou et al., 2019b). The monitoring of presence, condition and amount of relevant objects is the mainstay of a large number of automation of processes and quality assurance tasks. Both are strongly interrelated. To be more specific, processes possess specific characteristics that can be measured, analyzed, improved and monitored. Even though processes that run automatically need processes that control the output quality according to predefined specifications.

A wide range of technologies are associated with the Industry 4.0 concept: cloud services, Big Data, digital automation with sensors, analysis of virtual models and many more (Dalenogare et al., 2018). From the perspective of computer science, Big Data is particularly interesting: the concept is defined as the information asset that includes a high volume, variety and velocity and therefore requires special technology and analytical methods to be transformed into value (De Mauro et al., 2016). This value can go beyond understanding the data. It can be used to automate visual processes and to assure the quality of ongoing processes with computer-aided systems, which is our main aim. In this thesis, we mainly deal with the following question: **How can image data be computationally processed, analyzed and interpreted to allow automation of a conventionally manual process and assure its quality?** This question is examined below in three different and at first glance to a great extent independent areas. The three areas are: medical imaging, maritime inventory monitoring and retail applications. What these areas have in common is, on the one hand the kind of data basis they share, to be specific - images. On the other hand, they share the main focus the data is used for. Responsible and costly processes that traditionally usually are done by humans, which often is error-prone, should be handled computer-aided. In such a manner, the quality of the systems is expected to be improved, while reducing the error caused by humans.

The first section is structured as follows: In Chapter 1.1, the research field and main ideas of *Quality Assurance*(QA) are introduced. Then, in Chapter 1.2, we introduce the detailed research questions partitioned to the core application scenarios that are handled in this thesis. In Chapter 1.3 we present the contributions to the given research area. The introduction is finalized, giving an overview of this thesis in Chapter 1.4.

1.1 Automation and Quality Assurance

Each series of actions that is taken in order to achieve a result can be defined as a process (*Process definition, Cambridge* 2022). Therefore different types of processes exist, for instance sales processes (Oakes, 1990), research processes (Bouma et al., 2004) and diagnostic processes (Crombie, 1963) are model examples. In accordance with "Industry 4.0" and the possibilities it offers, a large number of until then manually handled processes need to be automated. Many processes are either to be solved based on visual information or visually controllable. Human resources may be appointed for *visual inspection* (VI) and *quality control* (QC). However, human labor can be expensive, slow and error-prone, especially compared to machine labor. A satisfying result of a process needs to be a measurable unit. In modern applications, quality and processing time are two desired essentials. While the acceptance of the processing time depends on its application field and may be of secondary importance, quality is always a key factor. The question therefore arises, what is quality? Beforehand: there does not seem to be any agreement; hence the term is interpreted differently in relation to diverse topics. In accordance with (Shewfelt, 1999), *quality* is defined as a series of attributes chosen based on precision and accuracy of measurements. These attributes depend on the particular context and must be therefore selected individually. Despite the affirming statement of (Shewfelt, 1999), which conveys that "internal validity" is provided by precision and accuracy of measurement to any scientific study, a number of other metrics may be meaningful.

Especially quality in the medical sense, also referred to as *quality of care*, is a vague term that can be interpreted differently according to personal needs and individual context. According to (Campbell et al., 2000), there is no widely accepted definition of terms such as *quality* or *quality of care*. It is assumed that the quality of care is measured by several factors, the availability of effective care with the aim of covering health benefits in relation to personal needs (Campbell et al., 2000). Preceding definitions emphasize very different aspects of quality of care. The agreement is to be found in two dimensions: effectiveness and efficiency (Donabedian, 1966; Maxwell, 1992; Association et al., 1992; D. S. O'Leary and M. R. O'Leary, 1992). Both properties are subject to positive sentiment, but just like QC, they leave much room for interpretation. The effectiveness and efficiency can only be measured in the context of a specific task and correspond to the specified metrics. Finally, these metrics monitor the quality and allow a comparison between the results and the expectations. The desire to ensure the best possible performance of a process gives rise to the desire for *Quality Assurance*. The online community for developers, architects and executives (TechTarget) defines QA as any systematic process of determining whether a product or service meets predefined requirements. Starting in the manufacturing industry, QA has since spread to the majority of industries, software development is no exemption (*QA Definition* 2022). QA and *quality improvement* (QI) go hand-in-hand in some cases, notably in medical care. While QA focuses on correcting errors in patients care quality, QI tends to focus on possibilities to improve quality by changing systems. QA relies on guidelines and standards. QI concentrates on a comparison to statistics against which the improvement is meant to be made (Schyve and Prevost, 1990).

Contrarily to medical application fields, QA for systems based on natural imaging provides, after launch of a reliable system, QI of underlying processes inherently. Automation and QA of processes share the commonality that both can be handled using CV. Consequently, their quality is only as good as the performance of the Machine Learning models behind them. These are monitored using firmly defined metrics and therefore monitor the quality of the processes. Further details on evaluation metrics that are chosen for this work are presented in Chapter 2.2.

1.2 Current Research Questions

The main research question this work deals with can be summarized as follows: **How can image data be computationally processed, analyzed and interpreted to allow automation of a conventionally manual process and assure its quality?** In order to answer this question, we chose several representative research areas with different requirements and needs. For this reason, this work is divided into three corresponding chapters: Medical Imaging QA, Maritime Inventory Monitoring and Retail Applications QA. Each of these fields has its own demands with reference to image data and the particular objectives. The latter depend on requirements that need to be fulfilled to achieve QA or automation. Therefore we define more detailed *research questions* (RQ) below.

In contrast to some related previous research, our studies provide a higher complexity of real-world data we use, as well as a wider spectrum of our investigations which consider different categories of images, origins and thematic context. Previous studies as well as related work are subject of the discussion in the corresponding chapters of this thesis.

1.2.1 Medical Imaging QA

The diagnosis of lung tuberculosis is made by experienced radiologists using tomographic images. Classification of disease severity is often done with the goal of an individualized treatment strategy based on the symptoms identified. This is referred to as *severity scoring*. Consequently, we want to deal with the following questions:

RQ1: Is an automatic severity scoring of lung tuberculosis from CT images feasible? The considered question contains the following aspects:

- Which features may be calculated using image data only?
- Which features are the most relevant?
- Is it possible to generate a report on the basis of 3D image data automatically?
- Which potential do neural networks offer for automatic report generation and severity scoring?

RQ2: How can sagittal rotation in MRI of prostate cancer patients be estimated using CV to monitor the quality of recordings?
The considered question contains the following aspects:

- Which sections of the recordings are particularly important for determining the rotation angle?
- Which neural network components out of a set of potentially meaningful and in which order contribute to a reliable performance?

1.2.2 Maritime Inventory Monitoring

Monitoring coral reefs is necessary to be able to prevent the effects of global warming and ecological destruction. For this purpose, large scale images of stocks must be recorded and compared at regular intervals. Therefore, annotation of recordings is required.

In theory, object detection is ideal for this task, since manual processing is almost impossible. However, one encounters a number of challenges. These include: High resolution and thus large sizes of input data, naturally unbalanced object distribution, fluctuating recording quality in terms of sharpness, color balance and many more. This raises the following question:

RQ3: Is an automatic localization and annotation of corals in large scale images feasible?

The considered question contains the following aspects:

- Can neural networks benefit from traditional feature engineering on large scale images?
- Do specific difficulties and limitations of OD occur in relation to large scale underwater images?
- Does an improvement of the image quality have a direct impact on the performance of OD?
- How can a fair data split in terms of train and validation splits be generated for highly unbalanced data?
- Does an ensemble of different algorithmic solutions provide more reliability?

1.2.3 Retail Applications QA

The results of previous RQs cannot be easily transferred to areas in which the time component plays an important role. One example of this is retail, which is limited in terms of time and hardware because of financial reasons. For that reason, we want to address the following questions:

RQ4: How can the quality of the sales process be ensured by counting and classifying barcode-free goods, such as fruits and vegetables, at local markets using CV?

The considered question contains the following aspects:

- Under which condition is the task an object detection task?
- How to manage object detection on budget hardware?
- Is it possible to avoid manual annotation work completely?

Furthermore, we consider alternative solutions and investigate for OD the following questions:

- How can the annotation effort be reduced and to what extent?
- Do representative super classes exist?
- Which accuracy value can be achieved using pseudo labels?

Research question	Research topic	Chapter	Publication
RQ1	Medical Imaging QA	Chapter 3.1	Bogomasov et al. (2018)
RQ1	Medical Imaging QA	Chapter 3.2	Bogomasov et al. (2019a)
RQ2	Medical Imaging QA	Chapter 3.3	Bogomasov et al. (2021)
RQ3	Maritime Inventory Monitoring	Chapter 4.1	Bogomasov et al. (2019b)
RQ3	Maritime Inventory Monitoring	Chapter 4.2	Bogomasov et al. (2020)
RQ4	Retail Applications QA	Chapter 5	Bogomasov and Conrad (2021)

Table 1.1: Research questions and contributions

1.3 Contributions

This thesis contains a mixture of required ML and CV fundamentals, published peer-reviewed papers, which were presented at international conferences and workshops, as well as additions to the published research. A schematic overview with regard to the RQ from the previous chapter is shown in Table 1.1.

Additionally, we published another work, methodically related but without a direct contribution to the listed research questions:

- In Bogomasov (2016), we examined how different sky conditions in images of mountainous area can be classified in order to select a proper segmentation algorithm for the present conditions.

- In Kerlin et al. (2022), we searched for ways to improve image quality in underwater images to reduce the impact of color casts. Furthermore, we investigated the impact of the backbone depth using faster R-CNN. Additionally, we tackled erratic annotations by proposing a merging strategy for predictions based on Non-maximum Suppression.

- In Bogomasov et al. (2023), we presented an application which is able to run inference on custom data for models created using the most popular machine learning frameworks (e.g. TensorFlow, PyTorch), visualize the output and evaluate it based on numerous provided filtering options. One of the major advances of the contributed application is that all operations run locally without leaving the working space. All these features allow detailed analysis and evaluation of the performance of customly constructed DL models which is a key to understanding the strength and weaknesses of each OD model. The shared application is free-to-use and particularly useful while working with licensed image data.

1.4 Outline of this Thesis

This thesis follows the following structure: Having presented the main question and related research questions in Chapter 1, Chapter 2 provides an overview over theoretical concepts and methods which form the basis of this work. The second Chapter is finalized defining required evaluation metrics for object detection, classification, regression and counting, as they build the methodical core of automation and QA strategies presented in this dissertation. Accordingly to the defined RQs, the thesis is split in the further course into QA in medical imaging and QA in natural imaging. In Chapter 3, we distinguish the problem of false diagnosis resulting in incorrect treatment of patients by medical experts. The special focus is on radiological imaging procedures. In the same chapter, we present approaches that are capable of automatic report generation and severity scoring of lung tuberculosis in CT. Furthermore, we provide an approach for automatic orientation estimation in MRI. Aside from medical application fields, we present two models for maritime inventory monitoring (Chapter 4). Chapter 5 presents an approach we contributed that differs from previous methods in its specific efficiency to meet the needs of the retail sector, which are also discussed in this chapter. Finally, Chapter 6 discusses the results and draws a conclusion, followed by an outlook.

2

Fundamentals

"It is a capital mistake to theorize before one has data."
— Sherlock Holmes

This chapter introduces fundamentals that form the core of this thesis. Since this work focuses on automation and quality assurance, the necessary fundamentals may be found in *machine learning*. For this reason, these field is briefly described first. Because the entire work focuses on handling visual information, the corresponding terms such as *computer vision*, *visual inspection*, *visual data storage* and *image file formats* are explained subsequently.

2.1 Machine Learning

ML has been defined as a branch of *artificial intelligence* (AI) that aims to perform predefined tasks using intelligent software. The backbone of such software is built by statistical learning methods, which form the developed machine intelligence. ML requires properly selected data to be able to learn. The data is mainly stored in a database (Mohammed et al., 2016). ML can additionally be seen as a set of methods for automated analysis of structure in data. These methods can be divided into two categories: *supervised* and *unsupervised* (Fisher et al., 2013). Both require properly prepared data.

In terms of *supervised learning*, we can assume the data having the following form:

$$\mathcal{D} = \{(x_i,\, y_i),\, i = 1, \ldots, n\}$$

with $x_i \in X$ is a data sample and $y_i \in Y$ its corresponding label.

In terms of *unsupervised learning*, the data is unlabeled:

$$\mathcal{D} = \{x_i \in \mathbb{R}^m,\, i = 1, \ldots, n; n, m \in \mathbb{N}\}$$

Furthermore, we call *machine learning* the process of deriving optimal parameters of the model M from data $\mathcal{D}$, s. t. a trained model can be defined as: $M : X \to Y$.

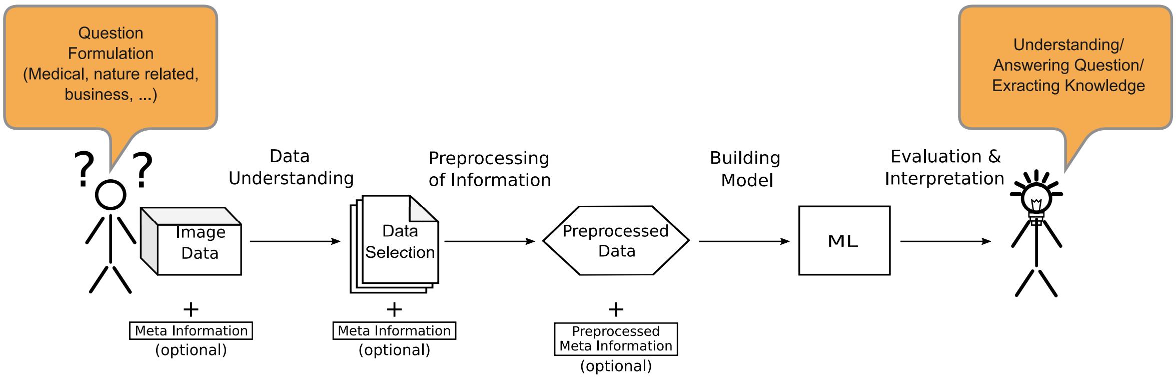

Question Formulation (Medical, nature related, business, ...)
? ?
Data Understanding
Preprocessing of Information
Building Model
Evaluation & Interpretation
Understanding/ Answering Question/ Exracting Knowledge
Image Data
Data Selection
Preprocessed Data
ML
+
Meta Information (optional)
+
Meta Information (optional)
+
Preprocessed Meta Information (optional)

Figure 2.1: Model building procedure. Inspired by (Brazdil et al., 2008)

Supervised learning can be distinguished into either a classification task or a regression task. While in the case of classification, the target label $y_i \in \{1, \dots, C\}$ is an element of a finite class, the target label in the case of regression is $y_i \in \mathbb{R}^q$ with $q \in \mathbb{N}$ and can be continuous. In both cases the aim is $M(x_i) = y_i$ which we call prediction. *Unsupervised learning* does not need target labels. It completely relies on understanding the data patterns itself. Commonly the aim of ML is called *generalization* which means the mapping of $x \in X$ s.t. $(x, y) \notin \mathcal{D}$ for any $x \in X$ to some $y \in Y$. The challenge is to avoid phenomena such as overfitting and underfitting. The first occurs when the training examples are memorized but no true generalization is reached. The second occurs in case when a ML model has an insufficient capacity or the training procedure is not fully completed (Bashir et al., 2020). The ability to learn is a key concept to the entire field of *artificial intelligence* (AI). Another strand of ML is called *reinforcement learning* (RL). It has a different approach behind. The idea behind RL follows an approach, which involves a learner interaction. The learner hat to discover which actions provide the most reward. The actions may not only affect the immediate reward but all the subsequent actions and rewards. Both characteristics - the trial-and-error and the delayed reward are the most distinctive features of RL (Sutton and Barto, 2018). Since RL is not part of this work, any deeper explanation will not be given (for a useful overview of RL, see (Y. Li, 2017)).

2.1.1 Data Mining

The process of automatically extracting valuable information from data sets is described as *data mining* (DM) (P.-N. Tan et al., 2016). The extracted information may be *descriptive* or *predictive*. While descriptive information provides understanding of patterns and structure of data, predictive information provides a prediction of one or more variables. Although the terms DM and ML are often used as synonyms, because of their common similarities, the idea behind DM is rather to learn structural patterns or behaviors from data, whereas ML is about independent machine operation (Fisher et al., 2013). Fig. 2.1 shows a traditional data mining procedure adopted to any kind of image data. Always starting from a formulation of a question, the image domain must be understood and data must be pre-selected, i.e. it must be divided into disjoint subsets. Subsequently, after preprocessing, the prepared data is passed to a model for training. Finally, the trained model is able to provide further understanding of the data, answer a posed question, or independently solve a given task if it meets the

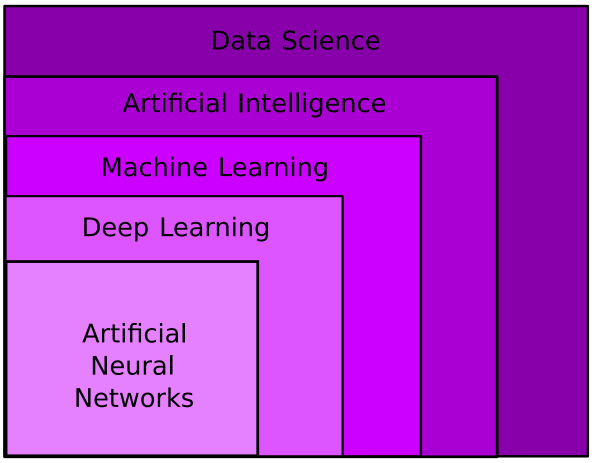


Figure 2.2: Data Science. Inspired by (Goodfellow et al., 2016)

predefined evaluation criteria. The special feature in this illustration is that there is no differentiation made between the different kinds of data science types. For that reason, it works well for both ML and DM.

2.1.2 Computer Vision

Computer vision (CV) is one of the essential fields of AI. One of the fundamental definitions denotes CV as a process that interprets an image and relates the result of this analysis to some action (Bisiani, 1987). This definition is quite powerful, because it assigns to a technical system the capability to see, understand and act.

Starting in the early 1970s at *computer science* (CS) impelling universities such as Massachusetts Institute of Technology, Carnegie Mellon University and Stanford by the early pioneers of AI, CV quickly received considerable attention. From the beginning, the key objective has been set to teaching the computer to depict the image content. Unlike the related *digital image processing*, which focuses on processing digital images by means of computer (Gonzalez and Woods, 2008) and aims to enhance images for further processing by vision algorithms (Parker, 2010), CV was meant to capture the three-dimensional world from a natural image and be used to fully understand the scene. In some way, CV intersects the field of *digital image analysis*, which is also about "extracting sense" from image data. The main difference lies in the level of "sense". There is also an overlap between image processing and image analysis. For example, recognition of regions can be categorized in a sense as both. Nevertheless, these three fields are not always clearly separable. Probably, a clear separation is also not necessary, since each of these research fields has been in the interest of research continuously. From its early days in universities until the early 2000s, CV dealt with a variety of topics such as edge and contour detection, quantitative image and scene analysis, image segmentation, and a whole host of other topics, with methods becoming more accurate respectively approaching the ground truth over the years (Feng et al., 2019). Convincing results on the whole range of topics summoned then a large number of opportunities for MV applications.

By implication, these fields partly belong to *artificial intelligence* (AI) that itself has the objective to emulate human intelligence (Gonzalez and Woods, 2008). AI is a small part of a huge field of *data science* (DS). DS includes subject areas such as *machine learning* and *deep learning* (see Fig. 2.2). All of these fields, especially CV, offer research-relevant approaches that could be used for automation and QA.

Generally, most CV algorithms tasks can be mapped to one of the three main CV tasks.

- Object recognition
- Object detection
- Scene understanding

While *object recognition* (OR) determines the presence of a specific object in an image, *object detection* (OD) goes beyond classification and appoints the location of a specific object of a known class. Both OR and OD are fundamental for Chapters 3.1, 4 and 5. In order to give a more complete "picture", scene understanding needs also to be mentioned. This field is about subdividing an image into meaningful segments (Feng et al., 2019) and is not part of this thesis.

CV offers a great variety of applications a reliable computer aided support, to name a few: vehicle guidance, automated inspection, analysis of remotely sensed images and bio-metric measurement (Davies, 2012). Monitoring and control surveillance should also not stay unmentioned. By way of example, automated recognition of traffic signs has been an important application of CV in cars for a long time and still is a sought-after feature for prospective buyers.

In most cases, a rough guiding principle of the feasibility of each CV solution, in the case of OR and OD, is the question of whether the human eye can see and recognize the object of interest. If it is the case, the computer is supposed to do the same and needs to be taught. In what way must the computer be trained to recognize the object of interest? - is the question that needs to be solved and hence the question that the research is concerned with. Particularly challenging and thus significant are scenarios in which an untrained eye either is not able to see and recognize the object of interest immediately or is not able to recognize it at all. This applies in particular to images that presuppose subject-specific knowledge, coming commonly from medical or other scientific sources. This dissertation deals with these kinds of questions in different application fields with the aim to simulate expertise by the computing unit.

2.1.3 Visual Inspection

A direct reference to the application of computer vision and at the same time the bridge to industrial and other business-oriented applications can be found in *visual inspection* (VI). Since the 1970s, automated VI, meaning "inspecting by looking", has been imbedded not merely in the industrial sector with a growth of double-digit percentage yearly (Beyerer et al., 2015). Beyerer et al. define the typical tasks of VI as follows:

- Object and pattern recognition for completeness
- Position and orientation
- Persistence of dimensions

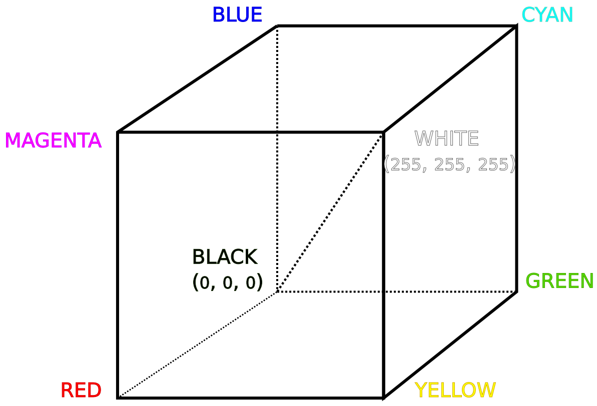


Figure 2.3: RGB color space cube

- Surface and texture condition
- Measurement of visual properties
- Identification of materials and surfaces
- Detection of defects

Davies (2012) defined the objective of automated visual inspection as the comparison of individual manufactured items with the preestablished standard with a view to the maintenance of quality. In order to extend the definition and reduce the limitation to manufacturing, we define a visual inspection as a process that aims to locate and compare individual objects to the expected ground truth with a view to the maintenance of quality.

The same author (Davies, 2012) separated VI into three stages:

- Image acquisition
- Object location
- Object measurement and scrutiny

In order to obtain a complete picture, the results after the stages need to be evaluated on real world data, paying special attention to influencing factors like: noise, occlusions, optical distortions, nonuniform lighting, shadows, reflections and rotations.

In general, VI being part of a visual inspection process relies on a number of technological methodologies. For the most part, CV and *machine vision* (MV). The often confused terms are clearly distinguishable. MV is associated with, but distinct from CV as well as the related fields: *image processing* and AI. However, it is less of a scientific topic, but rather a subset of systems engineering. As defined, it necessarily has to involve mechanical handling or other machine-related interaction (Batchelor and Waltz, 2001).

2.1.4 Visual Data Storage

Computationally, the storage and all arithmetic operations on image data in the computer are realized as matrix operations. The modern *Graphics Processing Units*

(GPUs) offer great parallelism for matrix-based operations, high memory bandwidth and support for single- as well as double-precision IEEE floating arithmetic (Che et al., 2008). They are many times faster than equally modern *Central Processing Units* (CPUs) on image data. Accordingly, CUDA GPUs are fundamental to computation for all contributions presented in this thesis.

Let $M_{n \times m \times c}$ be a set of all $n \times m \times c$ matrices with entries in $\mathbb{Z}$ where $n \in \mathbb{N}_{\neq 0}$, $m \in \mathbb{N}_{\neq 0}$ and $c \in \{1, 2, 3\}$. Because of the way visual data is stored in memory, we chose the following formulation for RGB images that describes it as it is:

$$I_{rgb} \in M_{n \times m \times c}(\{z \in \mathbb{Z} | 0 \leq z \leq 255\})$$

where $n \times m$ is the size of a 2 dimensional image with $x < n$ and $y < m$ and $c = 3$ which is the number of channels of the image. Although several mathematical models describe the way colors can be represented numerically (Joblove and Greenberg, 1978), the most common color space a.k.a. color model is RGB. The RGB color model, as visualized in Fig. 2.3, is an additive model, which means that the values of the three channels that stand for red, green and blue are added together to produce a new color. Each pixel value p is therefore represented as:

$$p_{rgb}(x,\, y) = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$$

We thus write:

$$I_{rgb} = (p_{rgb}(x, y))_{x<n, y<m}$$

The three bytes per pixel form a space of the size of 2^{3*8}, which results in a number of presentable colors of $\approx$ 16.78 millions. Pixels form the smallest unit of a digital image stored as a two-dimensional graphic. In the case of a three-dimensional image, a pixel is called *voxel* since it represents a volume. RGB images establish the basis for Chapters 4 and 5 that include work on natural photographs. Medical imaging, specifically MRI and CT images contain depth information furthermore they provide various meta information, which will be addressed in the subsequent course of this work.

The image material of the medical image data, which visually is a 3-dimensional image I_{3D}, can formally be reduced to a 4-dimensional matrix for the purpose of this work.

$$I_{3D} \in M_{n \times m \times d \times c}(\{z \in \mathbb{R}\})$$

where d is the depth dimension, representing the amount of slices in the 3D recording.

Two additional image types are essential to our work: grayscale and binary. Grayscale is a frequently used color space in the area of information retrieval and data mining in images and a common transformation method in the sense of preprocessing. It represents the intensities of an image, ranging from black to white.

Numerous works deal with the extraction of distinctive features on basis of natural images transformed to grayscale (Schulz-Mirbach, 1995; Tuytelaars and Mikolajczyk, 2008; Ojala et al., 2002; Forsyth and Ponce, 2011). Any transformation involves the risk of information loss in the first step. Sometimes such a reduction can lead to an increase of information in another place. In the case of true color images, for example, a transformation to grayscale is logically related to loss of color information.

However, grayscale can provide a better distinction between the background and edges. Furthermore, it can minimize computational costs and complexity compared to edge detection performed on color images. Under ideal circumstances, the detected edges may outline relevant objects within an image and thus provide the information basis for subsequent steps such as localization, quantification and further analysis.

Further is

$$I_{gray} \in M_{n\times m\times 1}(\{z \in \mathbb{Z} | 0 \leq z \leq 255\})\})$$

a grayscale image with $p_{gray}(x, y) = g$ accordingly.

In literature, there is a number of conceivable approaches for transformation from RGB to grayscale (Cadík, 2008). The most common methods are *Intensity*, which assumes an equal weighting of the three channels, as well as *Luminance* (Jack, 2011), which can be calculated using the following formula: $g = 0.2125 * c_1 + 0.7154 * c_2 + 0.0721 * c_3$

The transformation is carried out component-wise. Via the additive merging of the channel, certain image properties can be strengthened or weakened to the same extent. Theoretically, a weighted mean of the three channels in any proportion is imaginable.

Luminance has the advantage that it is designed to match human brightness perception and was proven to be a good choice for texture recognition (Kanan and Cottrell, 2012). Additionally, the method is computationally inexpensive because of its linear time complexity.

A further simplification of the image can be achieved via binarization. As a result, an image is created that consists only of the two colors black and white. One way of binarization is thresholding, where binarization is an operation $f : \{z \in \mathbb{Z} | 0 \leq z \leq 255\} \to \{0, 1\}$ with

$$f(p_{gray}(x, y)) = \begin{cases} 1, & if \, p_{gray}(x, y) > c \\ 0, & else \end{cases}$$

where c is the threshold that separates back- and foreground. Various methods for the determination of an optimal value for segmentation have been proposed (Dong and G. Yu, 2004; Moallem and Razmjooy, 2012), starting with the pioneer algorithm proposed by Otsu (1979). In this work, binary images are used as segmentation masks in Chapter 3.1 to restrict the region of interest to the area of the lungs and to increase both the speed and the accuracy of the presented algorithms.

2.1.5 Image File Formats

Great importance can be attached to a deep understanding of the way visual material is stored: a large number of *image formats* to hold the information has been developed over the last decades (Wiggins et al., 2001). The most commonly used for 2D-data are Joint Photographic Experts Group (JPEG), Tagged Image File Format (TIFF), Portable Network Graphics (PNG), BMP file format (Bitmap), Graphics Interchange Format (GIF), Encapsulated Postscript (Eps), complemented by a series of raw image file formats such as raw, cr2 or nef. Covering over 70% of all image formats used on 10 millions websites, PNG and JPEG were the most frequently used (*Technology Usage Statistics* 2022). Thus, JPEG being the most commonly applied format is used in all chapters of this work that handle with photographs. Certainly, its most assertive driving force is memory efficiency which is realized by compression. Although JPEG, for example as JPEG 2000 standard, is capable of lossless compression, such approaches are not noteworthy beneficial in most cases and therefore not convenient. The most present JPEG images are compressed with *discrete cosine transform* (DCT) (Miano, 1999). Different compression techniques have in common that they take advantage of patterns within an image with the aim to find an equivalent representation that allocates less space. Considering the fact that the human eye has difficulty recognizing the difference between compressed and uncompressed images, and due to the great similarity of colors, compressed data offers a great advantage, not only in storage or sharing, but also in image processing and analysis in particular. Thus, the way in which image data is stored is the first important step towards efficiency. In spite of all mentioned advantages, lossy compression, as such occurs during quantization of DCT coefficients, always means to a greater or lesser extent an information loss and can produce artifacts. While for most applications the threshold between the content of information and storage space, compression still is a good idea, for a few others it is not. Especially it is not recommended for compressing text and drawings as well as editing images repeatedly (Miano, 1999). Even for medical imaging, there is a high acceptance of the use of compression by American College of Radiology (ACR) and Canadian Association of Radiologists (CAR) standards (Koff and Shulman, 2006). In the case of medical images, two main groups of formats exist. The first aim to standardize the image by corresponding diagnostic modalities. The largest representative of this group, which has been widely accepted in a clinical context, is Digital Imaging and Communications in Medicine (Dicom) (Mildenberger et al., 2002). The second is intended to simplify post-processing analysis. It includes: MINC (*MINC File Format* 2022), Analyze (Robb et al., 1989) and Neuroimaging Informatics Technology Initiative (Nifti)(*NIfTI File Format* 2022). Along with visual information, descriptive information is stored inside of images. In simple formats at least resolution, pixel depth and the title provide a brief information about the origin of the file and its characteristics. More complex formats such as those used in medical imaging may contain deeper insights. Quantitative analysis in a clinical context requires image-related and image-specific information to be instantly available. This creates the need for specialized image formats. Commonly used formats for MRI and CT images are Dicom and Nifti. Each Nifti file consists of raw voxel intensities in *Hounsfield Units* (HU) (Hounsfield, 1973). Furthermore, the image contains the corresponding meta information i.a. slice thickness, image dimensions and voxel size in physical units. Each of these properties has an impact on the quality of the recording and requires great attention. Especially the latter may have a large impact on diagnosis accuracy along with the clinical decision making (Cooper et al., 2007; Shafiq-ul-Hassan et al., 2017). Dicom supports JPEG. JPEG2000 is a further development of JPEG which offers superior results to established image compression standards (Rabbani and Joshi, 2002) but is rather used in DICOM images than in natural photographs. While Dicom is limited to signed and unsigned integers as data types, Nifti additionally supports floats (Larobina and Murino, 2014). This means a great advantage for normalization but also all kinds of post-processing operations as well as the subsequent processing using DL. Because these show a beneficial effect on data scaled to the range -1 and 1. For this reason, a conversion might be advisable. The conversion from Dicom to Nifti is possible but not trivial (X. Li et al., 2016). The other way round, it is straightforward (Whitcher et al., 2011).

The part of this thesis, which is concerned with medical imaging, operates on Nifti, as such were provided by the challenge organization, if no further meta information is required, and Dicom otherwise.

2.1.6 Minimum Bounding Box

Object localization (OL) implies the need to save the position of each object in a memory efficient way. An approximate and still commonly accepted solution is the *Minimum bounding box* (BB). The idea behind BB is to enclose completely an existing object within an image in such a manner that its area is minimal, which leads to a unique and translational invariant result. BBs offer several advantageous properties:

- Linear computational costs
- Fast comparison using intersection metrics
- Efficient storage with only two spacial points and class label information, if needed

(Sidlauskas et al., 2018)

BB is not the most precise form of object representation. A few other, more accurate formulations have been proposed, which include: *Rotated minimum bounding box* (RMRB), *Minimum bounding circle* (MBC), *Minimum bounding ellipse* (MBE), *Convex hull* (CH) and *Minimum bounding n-corner* (NC) (Brinkhoff et al., 1993). Apart from, in some cases, more detailed delimitation, these representations are only interesting for selective fields of application. Additionally, they are more complex and less efficient in terms of the previously mentioned properties. Since none of the topics included in this thesis deal with the area, the work in the following is based on the conservative MBB variant.

Accordingly to the chosen notation, a bounding box can be defined as a 4-tuple $B = (x_1, y_1, x_2, y_2)$ in the case of 2D. Furthermore, we define $\bar{B} = \{(x, y)|x_1 \leq x \leq x_2, y_1 \leq y \leq y_2\}$ as all the points (x, y) that lie inside of a bounding box B. A formalization for the 3D case is also possible but will be left out since it is not relevant to this thesis. While OL expects only the objects to be localized, the aim of *object detection* is not only to localize the object, but also to recognize it. Therefore, object detection can be seen as the power set $\mathcal{P}$ of bounding boxes containing single class information c. Thus, $f : M_{n \times m \times 3} \to \mathcal{P}(\{(\varsigma, \hat{B})|0 \leq \varsigma \leq 1, \hat{B} \in \mathbb{R}^4 \times \mathbb{N}_0^+\})$ and $I_{rgb} \mapsto f(I_{rgb})$. Multi-class mapping is also possible, but not part of this thesis. Each of the bounding boxes corresponds to a specific class c, so that $\hat{B} = (x_1, y_1, x_2, y_2, c)$. Furthermore, a distinction between ground truth BB and output prediction BB, generated by an ML model, has to be made. While a ground truth BB is a tuple of 5 elements, prediction BB perceives an additional element called confidence score ς. Hereafter ς is referred to as confidence. This score is usually used for filtering to ensure the resulting output object to have a certain minimum score.

Therefore in terms of OD, we can assume the data to have the following form:

$$\mathcal{D} = \{(I^k, \{(\varsigma_l^k, \hat{B_l^k})|1 \leq l \leq n_k\})|1 \leq k \leq K, n_k \in \mathbb{N}\}$$

where $I^k \in X$ is an image, $\hat{B_l^k}$ is the l-th BB belonging to image k and $\varsigma_l^k \in \{\varsigma \in \mathbb{R} \mid 0 \leq \varsigma \leq 1\}$ its corresponding confidence.

2.2 Evaluation Measures

The quality of an ML-based assurance system can only be as good as the performance of the ML processes behind it. The evaluation of the performance, i.e. bench-marking of the degree to which each model is able to generalize, requires quality measures that enable a numerical comparison. Since this work covers several categories classification, regression and object detection, we will introduce measures for each category respectively.

2.2.1 Classification Measures

A wide range of metrics was introduced and compared to evaluate classification results (Ferri et al., 2009). Four cases can arise for each data sample during a classification process:

- True positive (TP) - A sample has been correctly classified to belong to the positive class
- False positive (FP) - A sample has been falsely classified to belong to the positive class
- True negative (TN) - A sample has been correctly classified to belong to the negative class
- False negative (FN) - A sample has been falsely classified to belong to the negative class

These prediction results create space for various metrics. In the following we consider the most commonly used:

$$\begin{aligned} Precision &= \frac{\#TP}{\#TP + \#FP} & Recall &= \frac{\#TP}{\#TP + \#FN} \\ Accuracy &= \frac{\#TP + \#TN}{\#TP + \#FN + \#FP + \#TN} & F_1 &= \frac{2 * Pr * Rec}{Pr + Rec} \end{aligned} \tag{2.1}$$

In general, none of these metrics considered alone is sufficient for evaluating a classification process. While $Precision(Pr)$ shows how many objects classified to a certain class C indeed belong to this label, $Recall(Rec)$ indicates a ratio of how many objects of class C are classified as C correctly. The disadvantage of Pr is that it does not provide any information about how many samples are not labeled correctly, whereas Rec does not take into account how many samples of other classes were falsely labeled as C. $Accuracy(Acc)$ behaves differently. It reflects the ratio of correctly classified data samples in the sum of all samples. Next to all of these F_1 reveals as a harmonic mean between Pr and Rec. It is also worth mentioning that besides F_1, further weightings exist. However, these are less common and more suitable for context-related questions, where one of the two measures is more important. In fact, individual metrics complement each other. In order to achieve reliable evidence, a combination of measures is required.

2.2.2 Regression Measures

In contrast to classification, regression is a function that maps input data onto a continuous space. For this reason, a categorical comparison, as presented in the case of classification, is not suitable for the evaluation. Rather, a distance measurement needs to be calculated, to evaluate the prediction quality. Two measurements are widely used, absolute mean error (MAE) and the root mean square error ($RMSE$). For every $(x_i, y_i) \in \mathfrak{D}$, let $\hat{y}_i = M(x_i)$ be the prediction for x_i made by model M, then $RMSE = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$. Where $y_i \in \mathfrak{D}$ denotes the ground-truth value. To be listed accordingly, $MAE = (\frac{1}{n})\sum_{i=1}^{n}|y_i - \hat{y}_i|$. The range of possible values for both MAE and RMSE is not bounded above. Therefore, both results should not be considered by themselves, but only in connection with the data distribution. A typical way is to consider the spread of the values (i.e. standard deviation (SD) and variation (Var)). While $Var = (\frac{1}{n})\sum\limits_{i=1}^{n}(\hat{y}_i - \mu)^2$, standard deviation is its root value $SD = \sqrt{VAR}$. In addition, boundary values $\underset{x \in \mathfrak{D}}{min} f(x)$ and $\underset{x \in \mathfrak{D}}{max} f(x)$ are often given in order to face the outliers.

2.2.3 Object Detection Measures

The question of a proper bench-marking of an object detection algorithm is a more complex topic that was discussed over a longer period of time. Already as early as the beginning of the 2000s, metrics such as Area-based Recall and Area-based Precision, Average Object Area Recall, Average Detected Box Area Precision and also Localized Output Box Count Precision have been introduced. These metrics promised a meaningful evaluation (Mariano et al., 2002). Metrics, based on Precision and Recall, were later on further developed by implicating of the overlap of objects including bounding boxes.

Common objects in context (COCO), one of the most important data sets for OD ever published as part of an object detection challenge, included an extensive selection of twelve evaluation metrics, made explicitly for the evaluation of the performance of OD (T.-Y. Lin et al., 2014a). The primary challenge metric was the *Averange Precision* (AP), which was already used in the equally well-known “Pascal Visual Object Classes Challenge” (VOC) (Everingham et al., 2010a). This metric has the advantage of taking into account both the class match information and the perceptual overlap (i.e. *Intersection over Union* (IoU)) of the corresponding BBs as well. As a result, only BBs that overlap over a larger area than a certain accepted threshold τ and also belong to the same class have a positive impact on the metric. Only BBs that fit this precondition are counted as TP. For two BB B and B' let $\bar{B}$ and $\bar{B}'$ be the corresponding sets of points that lie in B and B', respectively. Then the overlapping area of two BB with the same class is calculated as $IoU_{>\tau}(\bar{B}, \bar{B}') = \frac{area(\bar{B} \cap \bar{B}')}{area(\bar{B} \cup \bar{B}')}$. Two essentials Pr and Rec need to be calculated in the first step. Subsequently, both components have to be put in relation to each other and thus form a curve (PrRec), where recall is mapped to the x-axis and precision to the y-axis. Comparable to F_1-Measure, the PrRec curve is obtained to make an easy observation of the trade-off between the two metrics. This curve is designed as an interpolational approximation of precision $Pr_{interpolated}$. The interpolation is calculated over eleven equally distanced recall levels taking the maximum precision measured for which the corresponding recall rises at least to the value of Rec. Respectively, $Pr_{interpolated}(Rec) = \underset{\widehat{Rec}:\widehat{Rec}\geq Rec}{max} Pr(\widehat{Rec})$, where $Pr(\widehat{Rec})$ is the value of precision at recall level $\widehat{Rec}$. The area enclosed by the curve, is called AP and can be calculated as follows: $AP = \frac{1}{11} * \sum_{Rec\in\{0,0.1,...,1\}} Pr_{interpolated}(Rec)$ (Everingham et al., 2010b). While for classification tasks the average over all classes is calculated and thus called mAP, no distinction is usually made in the case of OD. Later on, a more precise approximation including a 101-point interpolation has been used by COCO challenge organization (*COCO evaluatation metrics,* 2020). However, the version that is never omitted is the PASCAL VOC ($mAP_{0.5}$). The index indicates the least required overlap of the bounding boxes to be counted. The standard is 0.5, which corresponds to an overlap of 50%. Additionally, in some cases, mAP_0 is also expected. It indicates great results, if the detection successfully matches the class name, but without the consideration of the corresponding position.

2.3 Deep Learning

Deep learning is a conglomerate of computational models containing multiple processing layers that allow learning data representations with multiple layers of abstraction (Voulodimos et al., 2018). This property offered in recent years an advantage for different fields of AI applications, *natural language processing* (Otter et al., 2020), *clustering* (Min et al., 2018), *transfer learning* (Weiss et al., 2016) and *visual understanding* (Y. Guo et al., 2016), just to name a few. Various vision tasks, e.g. such as *image classification* (Rawat and Z. Wang, 2017), *object detection* (Jiao et al., 2019a), *image retrieval* (W. Chen et al., 2021), *human pose estimation* (Zheng et al., 2020; Yucheng Chen et al., 2020) and *semantic segmentation* (Garcia-Garcia et al., 2018; Y. Guo et al., 2018; Lateef and Ruichek, 2019) are the driving force behind the development of architectures of all kinds. CV-related architectures can be roughly divided into three major categories: *Convolutional Neural Networks* (CNNs), the "Boltzmann family" including *Deep Belief Networks* and *Deep Boltzmann Machines* as well as the *Stacked Autoencoders* (Voulodimos et al., 2018). Due to the fact that a substantial part of this thesis concerns with CNNs, a brief explanation of their origin and theoretical background together with their prevalent buildup will be provided in the following.

It has been over 60 years since the idea for the first neural network was presented. Back then, the underlying idea presented a probabilistic model for information storage and organization in the brain called Perceptron (Rosenblatt, 1958). At that time, the presented theory set a milestone and inspired the development of NN architectures with different constructions and designs but consisting of just a few hidden layers, while the compatibility remained a main issue. To this day, the research strongly coupled with technological development, provided architectures of growing complexity. Considering architectures for object detection in the course of time, a steady increasing of depth can be observed. Going from five layers (LeCun et al., 1989), over twelve layers in AlexNet (Krizhevsky et al., 2012), up to nineteen layers in VGG (Simonyan and Zisserman, 2014), twenty-two layers in GoogleNet (Szegedy et al., 2015), the depth increased up to 152 layers in ResNet (K. He et al., 2016). The currently deepest network counts 1200 layers (Huang et al., 2016). A development that is exciting merely from a research

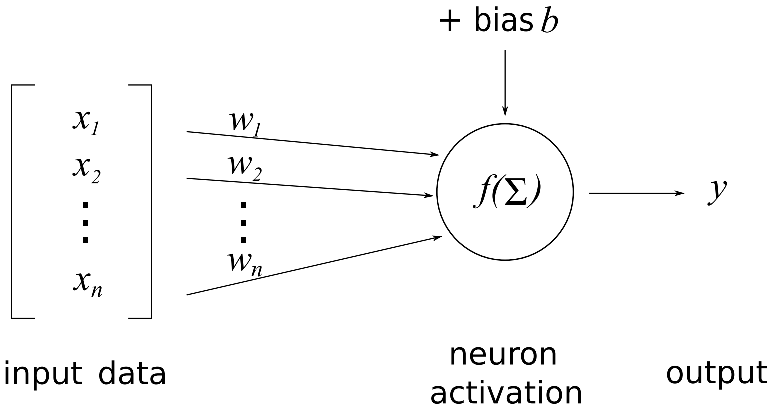


Figure 2.4: Schematical NN with one neuron

point of view, since a meaningful application is rarely found due to a lack of sufficient resources.

To understand the theory behind the complex models, we have to start with very simple ones. Fig. 2.4 visualises a NN with just one neuron. It takes an input vector $X = [x_1, \ldots, x_n]^T$ and solves the function $f : X \to Y$ that can be defined as: $f(x) = \sigma(b + \sum_{i=1}^{n}(x_i * w_i)) = y$, where $W = [w_1, \ldots, w_n]^T$ is the corresponding weights that are optimized during training and b is the bias term which is also learned for each neuron and allows to shift the activation output. In total, the given scheme provides n trainable parameter and additionally one tunable. Accordingly, to the construction idea, σ is the so-called *activation function*. An essential requirement for a NN is, similar to previously discussed ML, the ability to learn. Activation functions have a central role in improving the learning process of a NN and help avoiding characteristic learning problems such as the *vanishing gradient problem* (VGP) - more on this later. Renouncing activation functions would lead to a limitation of the NN to learn only a linear relation between input and the expected output (Apicella et al., 2021). The most commonly used activation function remains *Rectified Linear Unit* (ReLU), $ReLU(x) = max\{x, 0\}$. Its popularity can be traced back to its simplicity and efficiency. If a restriction in terms of the range of value is reasonable, frequently $\sigma(x) = 1/(1 + 1/e^x)$ and $tanh(x)$ are taken use of (Bingham et al., 2020). These activation functions are generic and are used regardless of the specifics of the individual application fields.

On the side, Glorot et al. (2011) proved that NNs with rectifier nonlinearities instead of widely used sigmoids perform much greater on image recognition tasks. Additionally, NNs using sigmoidal activation functions can also suffer from VGP, which may happen if lower layers gradients of a NN are almost 0, because higher layer values are nearly -1 or 1, depending on the valid range (Maas et al., 2013; Nwankpa et al., 2018). In general, VGP slows down the optimization convergence or, in the worst case, leads to a weak local minimum. Whenever a unit, e.g. a neuron, is not activated, its gradient is 0 which is not ideal since in an environment where the optimization is performed based on gradient, a unit might never be activated. Therefore, a saturated unit, if not activated, gets a value close to zero but not a hard zero. In this particular case, we speak of *leaky rectifier linear units* (lReLU) (Maas et al., 2013). Surprisingly, leaky rectifiers perform almost identically to standard rectifiers. To give an example, both have been compared on CIFAR (Xu et al., 2015) and LVCSR (Maas et al., 2013). In the further course of the research, a series of modifications has been introduced. However,

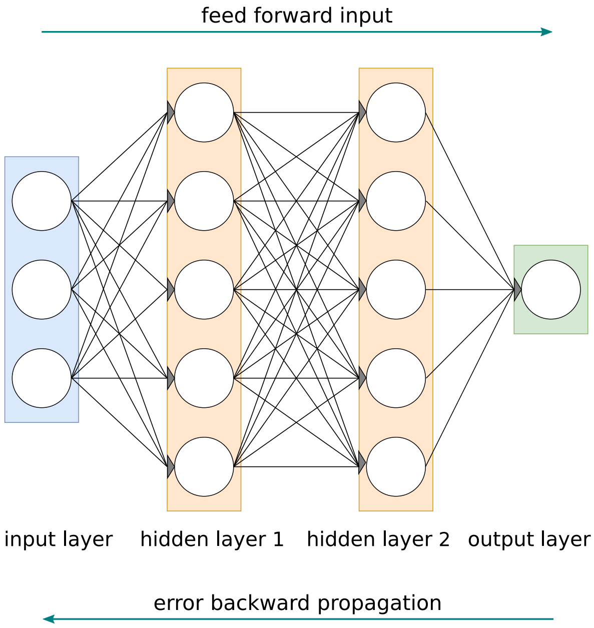


Figure 2.5: Backpropagation in a neural network with two hidden layers

ReLU remains the most approved.

The brief explanation above serves as an example of NN's functionality and general structure. For an extended overview of prevalent activation functions, the survey of Dube (2021) is recommended.

Going one step further, a simple NN can be extended and gain depth. Fig. 2.5 shows an enlarged fully connected NN with two hidden layers. Usually, hidden layers consist of neurons along with any thinkable operations that take previous layers' input and apply commonly a linear transformation followed by overturning linearity. The aim of intermediate layers is to convert input information into "something" the subsequent layers are able to use. A more precise specification depends on the construction and the field of application. However, the internal values are learned during the training while the loss is minimized. The loss value is minimized using optimization algorithms, usually referred to as *optimizer*. Among optimization algorithms, *gradient descent optimization* (GDO) is one of the most popular. It is also considered as “black box optimization” (Ruder, 2016). The convenience of not having to comprehend the lapse in detail is probably one of the major reasons for its success. The idea behind gradient descent optimization is the minimization of the objective function $\Theta(P)$, where $P \in \mathbb{R}^d$ is a set of the model parameters. The minimization is realized by updating the parameter set in the reverse direction of the gradient of the function $\nabla_P \Theta(P)$ in relation to the parameters. Each update is performed with a certain step-size also referred to as *learning rate* λ. The choice of step size is essential. If λ is chosen too large, a global minimum might be skipped. On the downside, if λ is too small, the training is lengthened unnecessarily and might end at a local minimum. According to (Ruder, 2016), GDO can be roughly divided into three categories: batch gradient descent (Géron, 2020), mini-batch gradient descent (Khirirat et al., 2017) and the most established stochastic gradient descent (SGD) (Robbins and Monro, 1951; Kiefer and Wolfowitz, 1952). These in turn have their own successors and optimizations. Gradient is calculated via *backpropagation* by calculating the gradient of the loss function and updating the weights. The backpropagation (Rumelhart et al., 1986) algorithm is divided into two logical steps: *forward pass* and *backward pass*. The first pass denotes the computation run from the input all the way to the output and is finalized by calculating the loss based on the output value compared to the ground truth information. The second pass goes all the way back through the computation graph, while partial derivatives with respect to the parameter terms are computed (Dube, 2021). From a practical point of view, the loss is a default optimization metric. During training, a combination of several metrics is common. Especially OD benefits from a combination of several values, since a differentiation of the individual components of the NN gets feasible. For this purpose, we select the metrics presented in chapter 2.2 and expand these individually if required regarding to application scenario. Taking a closer look at SGD shows that after each step a recalculation is provided on each training sample (x_i, y_i) over a predefined learning rate λ by bringing to update $P = P - \lambda * \nabla_P \Theta(P)(P, (x_i, y_i))$ (Ruder, 2016). Thus, the learning rate has a multiplicative effect on the optimization. However, a perfectly suitable value is difficult to choose in case no insight information is available. An ongoing trend is not to keep the learning rate constant, but to reduce it dynamically while setting an initially high value as proposed by Zeiler (2012). Even though dynamically changing learning rate improves the training time, it still does not provide from passing the local optima. Qian (1999) contributed the idea to help the NN out during training from a local minimum using a function termed *momentum*. The idea behind momentum is to multiply a non-negative floating-point number m to the previously calculated weight w. In closing it takes the form: $\Delta w_t = m\Delta w_{t-1} + (-\nabla_P \Theta(P))$. By the addition of the previous time step value multiplied by the momentum, momentum smoothes out variations along different directions. Still another major problem may occur. Even though input data may be normalized, standardized or both, the output of the activation function may become increasingly large. If during optimization a specific neuron weight becomes an outrageously large value compared to other neurons, it may have a cascading impact on the successive neurons and cause instability. Preventing huge values may be handled using a strategy named *batch normalization* (Ioffe and Szegedy, 2015). The strategy follows the idea to include two arbitrary trainable parameters $\gamma^{(i)}, \beta^{(i)}$ for each activation $\alpha^{(i)}$. Therefore, the output value $y^{(i)}$ is scaled and shifted $y^{(i)} = \gamma^{(i)}\hat{\alpha}^{(i)} + \beta^{(i)}$ (Ioffe and Szegedy, 2015). Thereby, normalization is included in the gradient process per batch basis and reduces instability during training.

Data is processed in batches. The *batch size* is a parameter that can affect both the duration of the training and the generalization as well. Therefore, the ideal size is addressed in many publications (G. Zhang et al., 2019; T. Wang et al., 2020; Smith et al., 2017; Kandel and Castelli, 2020). A major insight is that no general recommendation can be given; rather, the size depends on many factors, such as the area of application, interaction with other components of the system, but also and for most the image properties. However, the following applies with restrictions: increasing batch size provides a speed-up during training but may lead to a diminishing return (Golmant et al., 2018; Yuxin Chen and Krause, 2013), though increasing is not always viable. The availability of hardware and its properties, especially for image data, is a factor that quickly sets a limit. F. He et al. (2019) highlighted two important rules for the training of DL models. Firstly, during employing SGD to train DL networks, the best generalization is achieved when the batch size is not too large and the learning rate too small. Secondly, the ability of a DL model to generalize is negatively correlated

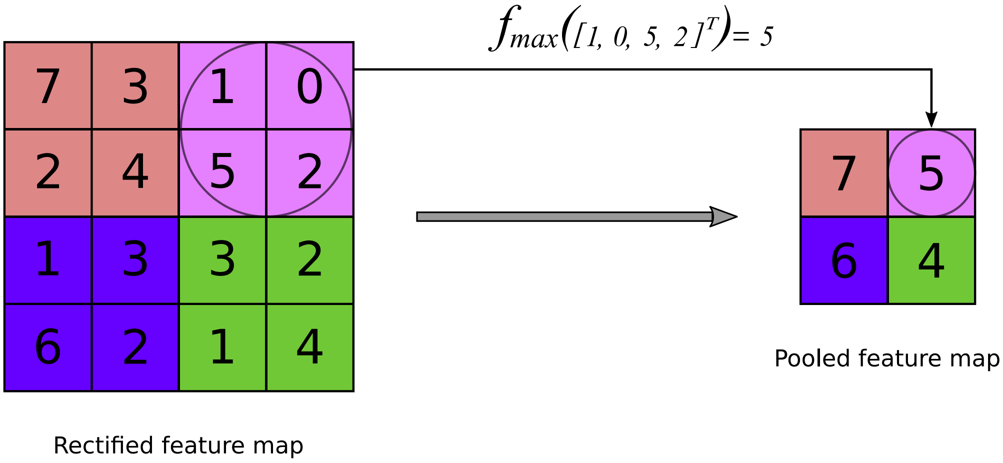


Figure 2.6: Max pooling operation with a stride of 2 and a 2 × 2 filter visualised on exemplary data

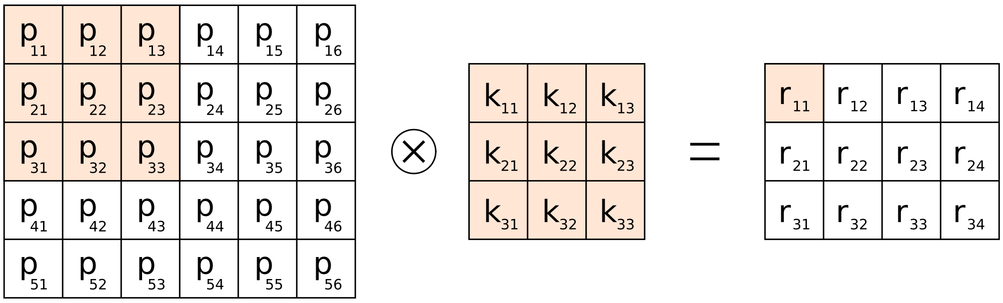

p_{11} p_{12} p_{13} p_{14} p_{15} p_{16}
p_{21} p_{22} p_{23} p_{24} p_{25} p_{26}
p_{31} p_{32} p_{33} p_{34} p_{35} p_{36}
p_{41} p_{42} p_{43} p_{44} p_{45} p_{46}
p_{51} p_{52} p_{53} p_{54} p_{55} p_{56}
$\otimes$
k_{11} k_{12} k_{13}
k_{21} k_{22} k_{23}
k_{31} k_{32} k_{33}
$=$
r_{11} r_{12} r_{13} r_{14}
r_{21} r_{22} r_{23} r_{24}
r_{31} r_{32} r_{33} r_{34}

Figure 2.7: Schematic visualization of a convolution operation computed on an 6 × 5 image. For instance, the highlighted result of the convolution operation for r_{11} is then calculated by the following equation: $r_{11} = \sum_{i=1}^{3} \sum_{j=1}^{3} p_{ij} \cdot k_{ij}$

with the ratio of batch size to the learning rate. However, both rules are valid only for SGD-based approaches. These formulations are also quite vague and lose their validity with an increasing number of hyper parameters. Thus, all in all even the basics of DL have not been sufficiently researched. The described fundamentals already allow the development of a simple but sustainable neural network for simple classification and regression tasks. Subsequently, we will have a look at the basics required for processing of visual information.

2.4 Convolutional Neural Networks

The most important class of ANN in terms of visual imagery is *convolutional neural networks* (CNN). The key concept of each CNN is its eponymous convolutional operation. A typical CNN consists of a convolution with a number of linear filters leading to a linear output, nonlinearity via activation functions (e.g., ReLU), and local pooling e.g. max or average pooling). Fig. 2.7 shows a simplified convolutional operation on an one-channel 6 × 5 image. A field of view called *kernel* k, most commonly of shape 3 × 3, is moved one by one over the image matrix. For the purpose of reduction of the resolution, a larger step size also referred to as *stride*, can be selected. Each movement is followed by element-wise multiplication and addition. For instance, the highlighted result is then calculated by the following equation: $r_{11} = \sum_{i=1}^{3} \sum_{j=1}^{3} p_{ij} \cdot k_{ij}$. Due to the limitation of the example showing only the first result, the mirroring of coefficients, which is common for convolutional operations, is omitted for simplicity in the calculation. Since the kernel usually operates on existing values the output is reduced in shape w.r.t. the input. Usually, a convolution is followed by an operation known as *pooling*. The aim of pooling is to reduce spatial sensitivity. In addition to a more compact representation, better robustness against noise and clutter is also expected (Boureau et al., 2010). Therefore a sliding window of size $s \times s$ (e.g. 2×2 or 3×3) is moved over rectified feature maps (the result of previous operations) in an arbitrary step size. Each time the values inside the window are processed using a commutative combination rule. A large number of calculation strategies is conceivable (Gholamalinezhad and Khosravi, 2020). Generally, the pooling operations are split into two categories: rank-based and value-based (Bera and Shrivastava, 2020). The latter are widely used. Among these, the most common is *max pooling* (Ranzato et al., 2007). It is calculated using the following equation: $f_{max} = max_i(x_i)$, where x is a vector containing the activation values from a selected pooling region. Its exemplary representation is visualised in Fig. 2.6. Finally, the output is processed by an activation function. The result r is named: *feature map*. Usually, feature maps are used as input for further convolutions subsequently. The depth of the operations, performed in a row, forms the “deepness” in the DL architectures. The overall goal of the convolution is the extraction of distinctive features. In contrast to features that might be predefined by the developer, the features in a CNN are not fixed since the weights of each kernel are learned during the training procedure. Different types of implementation for convolution layers have been proposed: 1×1-convolution (M. Lin et al., 2013), dilated convolution (F. Yu and Koltun, 2015), transposed convolution (Dumoulin and Visin, 2016), flattened convolutions (Jin et al., 2014), depthwise or spatially separable convolutions (Chollet, 2017), grouped convolution (Krizhevsky et al., 2012), pointwise grouped convolution (Xiangyu Zhang et al., 2018), etc. They differ not only in the way the source pixels are processed, but also in the computing load. For example, the 1×1-convolutions are much faster to be calculated, than their ordinary variant, as described previously. Therefore they are popular for the dimensional reduction in CNNs (Szegedy et al., 2015). This work makes use of different advantages of the listed techniques.

These basics already allow the development of a simple image classification network. Subsequently, we will have a look at the basics required for object detection.

2.4.1 Object Recognition

Object recognition (OR) can be defined as the act of finding an accumulation of pixels that describe an object of interest and the assignment of a name to it. The assigned name is then called: *label* (Parker, 2010). Usually, the object has characteristic patterns, sometimes also referred to as *features*. Indeed, contrary to patterns, which are “chunks” of visual data, features can be a calculated representation of any specific area.

Traditionally, OR is started by finding elementary patterns in low-level objects, then patterns of a higher level are searched for. One after another, these patterns form the complete object representation. To give a vivid example, the easiest way to recognize a car might be to search for tires, windshields, head- or backlights and doors at first and build up the car from these pieces as the next stage.

Parker (2010) summarized the general procedure of traditional OR as a four-steps approach:

- Object candidates isolation
- Finding of descriptive features
- Measurement of feature ambiguity for each label
- Searching for other features in case measurement results are not satisfying, followed by repeated measurement

The quality of a visual OR model depends on its stability against real world difficulties such as noise, lighting, orientation, scale, overlapping and others.

2.4.1.1 Object Detection Using DL

In the past 20 years hardly any other area on the field of ML received as much attention as object detection did (Zou et al., 2019a; Jiao et al., 2019b; Liu et al., 2020). The main goal of object detection, as mentioned previously, is the analysis of the presence of certain predefined instances of objects in visual data. In contrast to object recognition, object detection additionally is about providing positioning information to each object instance. Therefore it is one of the most challenging disciplines in computer vision, if not the most challenging since there are several conditions that have an impact on successful detection. The first is localization: each object instance, independently of its size, rotation and illumination, needs to be localized. The second is classification: each found instance has to be classified. Both presuppose the extraction of object-specific features from labeled data. Additionally, the features have to be learned on a great level of abstraction which in turn allows the ability to recognize similar but previously unseen objects. Therefore, deep learning is great for this task. Generally speaking, there are two types of detectors: Single-stage and two-stage. Both have in common that they make use of CNNs for feature extraction. However, there are essential differences in the structure, which in turn influence crucial network properties such as efficiency and accuracy. Both properties are essential for OD-based automation and quality assurance systems. The two types of detectors are part of this thesis and therefore briefly described in the following.

2.4.1.2 Two-stage Detectors

Multi-stage detectors follow the principle of subdividing the logical steps that lead to the detection of one or more objects. As is usual, detectors consist of two steps. For the most part, the first step is about generating object proposals based on features extracted from image data. Then in the second step, the most probable object regions are classified.

R-CNN (Girshick et al., 2014) is presumably the most prominent two-stage detector. Fig. 2.8 illustrates its detection procedure. Starting with an image, bottom-up region proposals are extracted taking advantage of selective search (Uijlings et al., 2013; X. Wang et al., 2013). Right after, for each proposal features are computed using a large

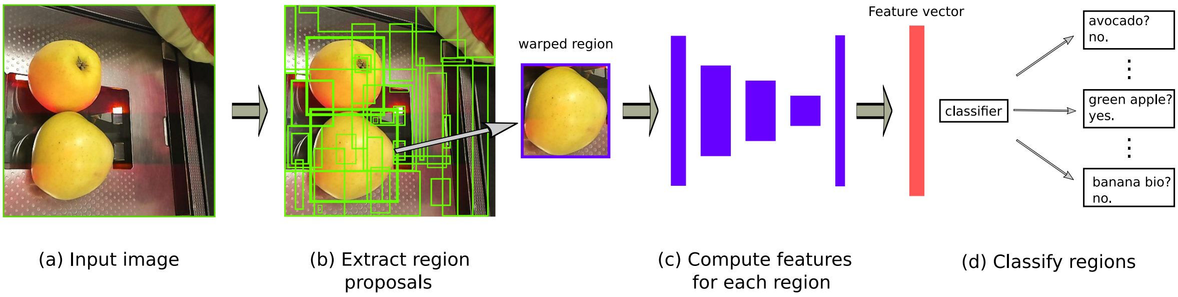


Figure 2.8: Two-stage R-CNN (Girshick et al., 2014) detector visualized on custom data set presented in Chapter 5.2

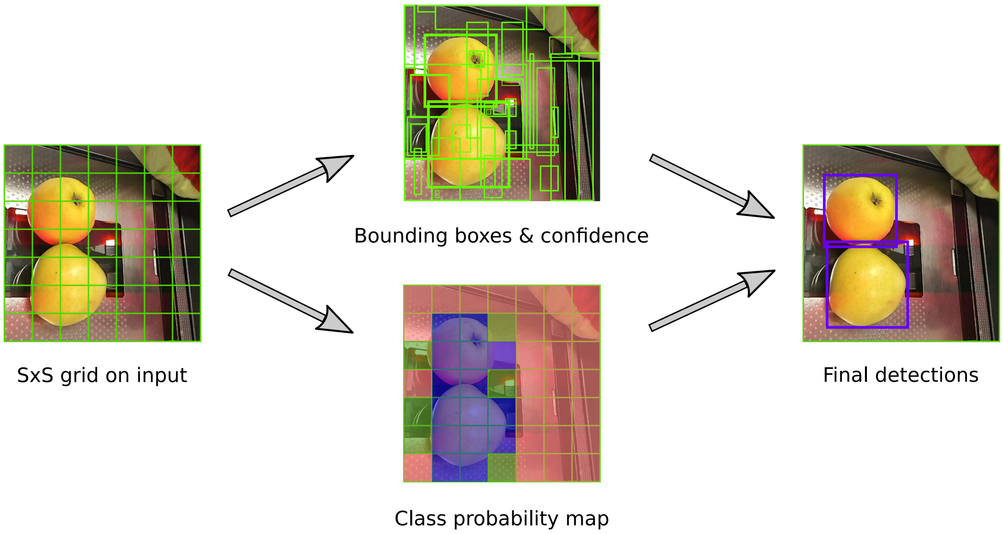


Figure 2.9: Single-stage detector visualized on custom data set presented in Chapter 5.2

CNN. Finally, each proposal is classified making use of class-specific linear SVMs. The individual steps have been optimized over time, but the general process has remained unchanged (Girshick, 2015; Ren et al., 2015).

2.4.1.3 Single-stage Detectors

In contrast to multi-stage detectors, single-stage detectors, as the name suggests, work in one go.

Yolo (Redmon et al., 2016) is one of the most popular single-stage object detection frameworks. Instead of using a classifier to perform object detection, as seen in Fig. 2.8, the detection is framed as a regression problem to spatially separated BBs and related class probabilities. A single network includes the entire object detection pipeline. Therefore, the training can be considered an end-to-end optimization problem. Fig. 2.9 visualizes the idea behind bounding box regression. Initially, the image is divided into a $S \times S$ grid. Then, for each grid cell, the model regresses bounding boxes, including corresponding confidence and class probabilities. In Fig. 2.9 cells with a higher probability of containing the object of interest are highlighted in blue. Being a pioneer architecture, Yolo has a few limitations: Each grid cell can only have one class, which restricts the recognition of nearby objects sharing a cell. Additionally, since the learning is grid-based, errors are threatened the same for small and large bounding

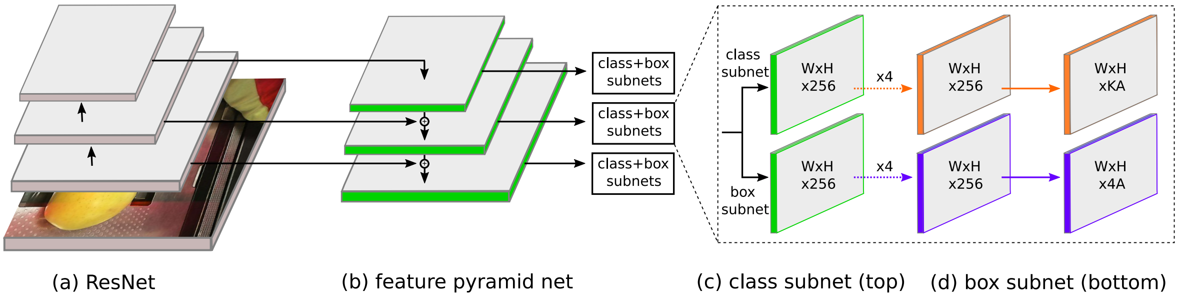


Figure 2.10: An illustration of the one-stage RetinaNet (T.-Y. Lin et al., 2017b) architecture is adopted to the custom data set presented in Chapter 5.2

boxes as well. However, errors in a small box have a greater effect on IoU.

Taking everything into consideration, we can recognize that Yolo in its first version is not suitable for images with imbalanced object distribution, as they occur in real world data.

One of the most outstanding architectures for imbalanced object distribution is RetinaNet (T.-Y. Lin et al., 2017b). Its composition, as illustrated in Fig. 2.10, is quite straightforward. Starting with a backbone consisting of a feedforward ResNet (K. He et al., 2016) architecture and a *Feature Pyramid Network* (FPN) a multi-scale convolutional pyramid is generated. Feature maps are primarily used for object detection. Since feature maps naturally decrease in shape during consecutive convolutions in a CNN, each level of the pyramid is used for object detection at a different scale. As visualized in Fig. 2.10 (b) a FPN has an inherent multi-scale pyramidal hierarchy. Therefore they contribute to instances of objects of a different size or the same instances with discrete features. For this reason, a Feature Pyramid Network (T.-Y. Lin et al., 2017a) is highly beneficial. On top of this formation, two sub-networks are attached. The first is used to calculate classifying anchor boxes. The second aims to regress anchor boxes to ground-truth object bounding boxes. The combination of these two enables objects of interest of different sizes to be completely enclosed.

RetinaNet is particularly interesting for real world applications, because of its built-in consideration as a rule that the foreground objects usually cover much less image area than the background. In contrast to the previously published one-step architectures such as Yolo, this circumstance is solved to a great extent via so-called *focal loss* (FL). This loss function is an extension to the widely accepted *cross entropy* (CE). Including a focusing parameter $\gamma \geq 0$, it is defined as $FL(p_t) = -\alpha_t(1 - p_t)^\gamma log(p_t)$, where p_t is the estimated probability of the class with a specific label and $\alpha \in [0, 1]$ a weighting factor such as the inverse class frequency. In case $\gamma = 0$, FC is equivalent to CE. As $\gamma = 0$ grows, the error on easy examples is downgraded and the loss isw reduced. The loss for hard negative examples grows counter-actively.

Because of several disciplines that object detection brings together, it certainly may be considered one of the most complex and demanding research fields in CV. However, it creates opportunities for real world scenarios like no other. Visual control and hence quality assurance are a prime example of a successful integration.

3

Quality Assurance in Medical Applications

"Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less."
— Marie Curie

A starting point for this research is Quality Assurance in medical applications, not least because it allows diagnostic and treatment errors to be avoided or at least reduced. As early as 2000, Donaldson et al. (2000) draw attention on the necessity of QA, naming number of affected patients who have suffered from medical errors. They pointed out that medical errors caused annually at least 44 thousand deaths in the United States (US). According to other sources, the number of humans affected was estimated even higher. The fatal outcome is not only dramatic for the patients themselves and their relatives but also detrimental to the economy. In the US, the annual additional costs in 2008 have been estimated to be at least \$17 billion dollar due to measurable medical errors (Van Den Bos et al., 2011). The measured value naturally forms the lower bound of spending, the actual costs are likely to be considerably higher. Therefore, error prevention is of primary concern, not only because of medical reasons, but also financial. The reasons for errors might be diverse. In about 75% of all diagnostic errors cognitive issues are involved. These are not cohering to deficiency in personal experience or expertise but to flaws in data (Nendaz and Perrier, 2012). Quality improvement and avoidance of mistreatment using AI have been studied for several cases (Davenport and Kalakota, 2019; K.-H. Yu et al., 2018; K.-H. Yu et al., 2018). The findings show that there is a rising need for all application scenarios since no medical diagnosis is guaranteed to be completely accurate. A proper strategy for a high quality treatment as a result of an image-based diagnosis requires first of all a reliable and error-free image acquisition. Additionally, a proceeding diagnosis could be accompanied by a computer-aided solution or might even be replaced by it.

In the following subchapters, we will present our contribution to diagnostic automation and QA of examinations on selected radiological topics, as such are not only very important from an ethical point of view but also from the view of CV because of the way the image data is generated. On the one hand, in comparison to regular 2D images, the radiological imaging receives an additional dimension, namely the depth, which creates the spatiality. The third dimension builds up a sequence of individual 2D recordings (called slices). In doing so, organs, bones but also soft tissue objects e.g. injuries, damages and particularly tumors, including cancer distributed over a number of successive slices, are combined. On the other hand, the properties of recordings can differ greatly depending on the acquisition device. In practice, different devices are commonly used, which increases the complexity of applications. In this context, we will address these challenges regarding the research questions. More specifically, we will examine whether an automated determination of the severity scores can be calculated using images of the lungs of patients with tuberculosis. Furthermore, an investigation of to what extent the image quality of the prostate of cancer patients can be ensured in terms of the ideal orientation of the recording. The reason for both investigations is that the data, as well as the diagnosis, are prone to errors and therefore require a quality control for automation and support.

3.1 Severity Scoring of Lung Tuberculosis from CT Images

Pulmonary Tuberculosis is a serious disease which is caused by bacterial infection of the lungs. If not treated properly, this illness can have fatal consequences to this day. Even nowadays it still is widespread and concerns the health care system on a daily basis (Lowbridge and Ralph, 2020). Multiple occurrences are common, and even worse any organ might be involved. However, in almost 75 % of the cases, it affects the lungs (Suárez et al., 2019). For this reason, this work focuses on tuberculosis that occurs in this particular area. Potential medical treatment of tuberculosis depends on the course of the disease and the associated symptoms. These symptoms reveal the progression of the illness and can be classified in severity scores. This kind of classification is currently done manually and is made possible by the latest technological developments in the field of radiology. The basis for such a classification is provided by *computed tomography* (CT). During image acquisition the tomographic signal is processed by the computer to generate "slices" of the body, i.e. cross-sectional images. These slices are provided as 3D-images and form a digital model of a part of a human body or even its complete representation. The offered spatial information enables a more precise diagnosis in comparison to conventional imaging technologies such as 2D-based X-Ray. The reason for this is obvious since disease-typical abnormalities can be located and identified more easily in 3D because of spatial characteristics. However, relying only on manual examinations may always be risky. Errors easily may occur while assigning a severity score to an examination. Such mistakes can have a serious impact on the appointed treatment. Therefore, we investigate **whether automatic severity scoring of lung tuberculosis from CT images is feasible?** Our main goal is to contribute a descriptive classification framework based on custom features that provides information about the influence of different kinds of irregularities in lungs on severity scoring. Such a development brings a whole series of advantages with it. On the one hand, an automatic classification could be used for QA of diagnosis, on the other hand, it could highlight and visualize the findings and thus offer a supporting function.

During our research on meaningful features, we simulate the thinking of medical experts who, in their manual examination, look for irregularities specific for tuberculosis within the lungs. We implemented and evaluated such features as lung calcification, lung wateriness, pulmonary cavities, infection ratio, Hounsfield histograms or lung shape comparison. The feature choice in our approach was for the most part based on our own observations that could be approved by medical studies published in the literature in recent years. However, while building these features, we entirely forego sources other than image data. In the following, the obtained features were used for the prediction of the severity score and the severity level of tuberculosis using different classifiers. Among the chosen classifiers, we evaluated *support vector machine* (SVM), *k-nearest-neighbor* (kNN), but also *decision tree* (DT), *random forest* (RF) and *linear regression* (LR). All of the mentioned topics are part of the next chapter.

Feature-Based Approach for Severity Scoring of Lung Tuberculosis from CT Images

Kirill Bogomasov, Ludmila Himmelspach, Gerhard Klassen, Martha Tatusch, and Stefan Conrad

Heinrich-Heine-Universität Düsseldorf, Institut für Informatik
Universitätsstraße 1, 40225 Düsseldorf, Germany
{bogomasov,himmelspach,klassen,tatusch,conrad}@cs.uni-duesseldorf.de

Abstract. Nowadays tuberculosis is still a widespread disease that causes worldwide more than one million deaths and ten million new infections every year. As part of ImageCLEF 2018, we investigated whether the severity of the disease can be determined from CT scans, only. We therefore extracted features from the images which we then tested with several classifiers. Afterwards we chose the best combinations of different feature sets and classification models. Our best approach is based on three features, namely cavitation, cavity tissue, and infection ratio. Combined with random forests we achieved rank 10 regarding the RMSE measure.

Keywords: feature extraction · image classification · lung tuberculosis

1 Introduction

In 2018 tuberculosis was listed as one of the top ten causes of death worldwide [1]. Depending on the severity degree of the disease different medical treatments are necessary. To this day the distinction of the severity degree has been executed by medical experts based on diverse information including the results of the mycobacterial culture test, pleural fluid and cerebrospinal fluid (CSF) analyses, lesion patterns in radiological images of the lungs, patient's age, duration of treatment and others [10]. In patients with tuberculosis, computed tomography (CT) is often performed for analyzing the lesion patterns in the lungs. The human-based analysis of the existing data is an expensive and time-consuming task. Additionally, a manual classification can be error-prone. In contrast to the manual examination of CT scans, a computer-based method could lower the error rate and simplify the procedure.

In this paper we present a feature-based approach for severity scoring of lung tuberculosis exclusively based on CT scans. This work is a contribution to the severity score tuberculosis task of ImageCLEF 2018 [6]. Besides the tuberculosis degree determination, the main goal of our approach was to create a descriptive classification framework that provides information about the influence of different kinds of irregularities in lungs on severity scores.

2 Feature Extraction

We extracted features from CT images assuming that medical experts look for irregularities in the lungs that are typical for tuberculosis while analyzing CT scans in the context of severity score determination. In the medical literature, different kinds of irregularities and lesions in the lung associated with pulmonary tuberculosis are described. Our feature choice is for the most part based on this description. In this section, feature extraction methods are described that we used in our approach. Since the most of our feature extraction methods worked on binary images, we binarized all CT images using IsoData method [15] in a preprocessing step. Hereby we also used lung masks which were extracted by the algorithm that was published in [7].

2.1 Lung Calcification

Calcification is significant to the disease pattern of tuberculosis[2]. We assume that the identification and quantification of chalk within lung lobes can be a meaningful feature. Hounsfield Units (HU)[4] of chalk vary around 700 depending on its density. These HU values overlap with those of bones (300 HU - 1500 HU), which are often located in the boundary area of the masks. Hence, a simple thresholding approach is not sufficient. In order to avoid misclassification, we therefore had to adapt the size of the masks as long as parts of the bones were contained. Finally, pixel with the value $\geq$ 700 are counted because we regard those as calcifications. In detail our approach contains the following steps:

(1) Slice-wise CT scan preparation: Set values below 700 to -3024 (no density)

(2) Slice-wise boundary analysis:

```
2.1 Boundary identification in mask slices:
    $boundary[i] = mask[i] - erode(mask[i]), with\ i \in \{0, slices(mask)\}$
2.2 Boundary extraction from CT scans:
    $bscan[i] = boundary[i] * scan[i]$
2.3 Adaption of masks:
        while (max(bscan[i]) $\geq$ 700) do
            mask[i] = erode(mask[i])
            Step 2.2
        end while
```

(3) Summation of pixels with value $\geq$ 700 in $mask[i] * scan[i]$

With *erode* [9], the function which executes an erosion on a binarized image slice.

2.2 Lung Wateriness

Water accumulation is a potential concomitant of an existing tuberculosis disease. Nevertheless we assumed that an existing tuberculosis infection weakens the immune system of the patient and may lead to trace-diseases. There are several indispositions that are associated with fluid retention within the lungs or in the pleural space between the lungs and the ribs, such as pleural effusion[18]. Our assumption was that water effusion is a clear evidence for an advanced level of infection or traces of a cured serious illness. The process of searching the water retention was based on the HU[4]. The searching algorithm is the same as in Section 2.1.

2.3 Pulmonary Cavities

One of the classic indicators of lung tuberculosis are the pulmonary cavities which occur in 50 percent of patients [13]. According to [12] pulmonary cavitation formed as a result of tuberculosis is a site of very high mycobacterial burden. They may lead to transmission of the infection to other humans and they are associated with emergence of drug resistance. Furthermore, in [11], the authors reported about the relationship between the cavity wall thickness combined with the diameter of the lesion and the malignancy of the disease. Therefore, we extracted the size of pulmonary cavities and cavity walls from CT images as further features for severity scoring of lung tuberculosis. Although pulmonary cavities may also occur as a result of other diseases like lung cancer [13], the presence of additional pulmonary diseases in patients with lung tuberculosis may increase the degree of tuberculosis.

We extracted the pulmonary cavities from single CT image slices as dark spots completely surrounded by light tissue. Since we wanted to avoid finding similar structures in other parts of CT images than lungs, we used the lung masks that were provided by the organizers of the task using the approach described in [7]. Due to variations of Hounsfield Units in different regions of cavities and cavity walls in different CT images, first, we binarized the CT images as described

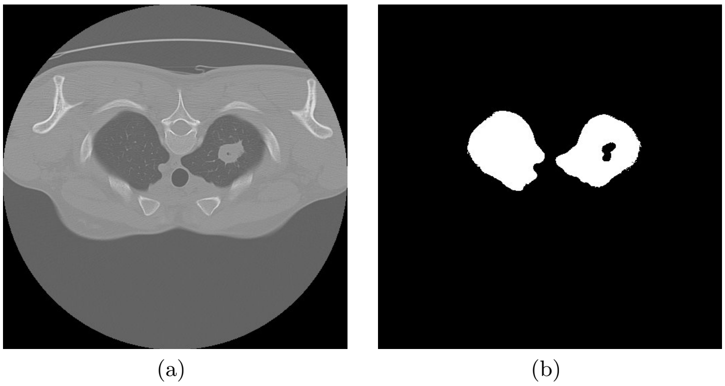


Fig. 1. An example of missing pulmonary cavitation in the mask: (a) CT scan showing cavitation in the left lung, (b) corresponding lung mask.

above. We had to make some adjustments to the lung masks because they did not cover the entire lung and often the cavities were cut out from the masks (compare Figure 1). Therefore, we closed all holes in the masks. Since we also closed the holes that correctly indicated bronchi, we cut out the middle part of the lung masks to avoid incorrectly recognizing bronchi as pulmonary cavities.

After processing the lung masks we performed the pulmonary cavitation search in binarized CT images as follows: First, we removed all objects smaller than 20 pixels because it is unlikely for a cavitation to be of such small size and analyzing such objects would unnecessarily require processing time. Since bronchioles scanned across and shadows caused by breathing and body movements could be falsely recognized as cavities, in the second step, we closed all holes that were smaller than two pixels. Obviously, the internal parts of undesired objects could be larger than 2 pixels, but, on the other side, we had to prevent erroneously discarding parts of real cavities. Because cavity walls are usually thicker than bronchiole walls, in the third step, we performed morphological opening with a 2 × 2 square to discard undesired objects remained after the second step. We considered all holes that were completely surrounded by walls as pulmonary cavities after performing these three preprocessing steps. For performance reasons we estimated the volumes of pulmonary cavities by simply summing up the pixels of found cavities and cavity walls, respectively, over all CT scan slices in the file.

2.4 Infection Ratio

Pulmonary tuberculosis is an infectious lung disease whose bacilli spread through the lungs and cause lung tissue damage. Depending on the type of tuberculosis different types of lesions occur in the lungs (see Figure 2). That makes it difficult to estimate the amount of the affected part of the lungs automatically. Since the infection of the most tuberculosis types cause a thickening of the lung tissue which can be recognized in CT scans, we simply estimated the ratio of the lung tissue to the entire lung volume. In our approach we did not differentiate between healthy and affected lung tissue which is a difficult task, our approach is based on the assumption that the lung tissue ratio compared to the lung volume is

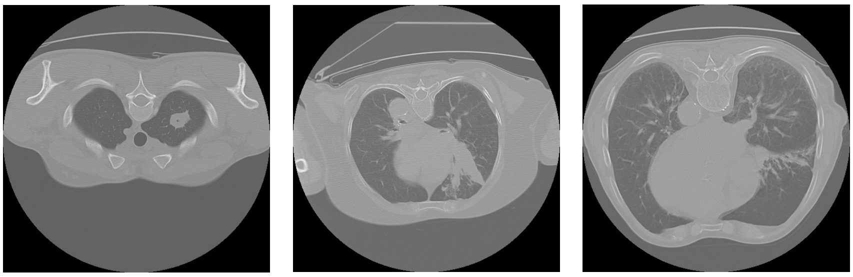


Fig. 2. Different types of lesions in the lung.

smaller in healthy persons than in persons suffering from tuberculosis. In order to highlight the lung lesions, we first binarized the CT images as described above. After the binarization we simply counted the number of white pixels and related it to the number of pixels in the lung mask [7].

2.5 Hounsfield Histograms

Since its introduction in 1972 [16] the technology of X-Ray computed tomography (CT) has been continuously refined. Over time several different devices with different parameter sets were developed [5]. The different technology and the parameters do not only concern the distance and time between images, but also the Hounsfield Units represented in the final image [14]. That leads to the problem, that the same object can have different Hounsfield Units on different images [5]. As there is no information provided what hardware and what parameters were involved in creating the scans for the dataset, it is difficult to look for certain Hounsfield Units which are comparable throughout all scans.

In order to overcome this problem we decided to compare intervals of Hounsfield Units with the help of histograms. As the intervals of Hounsfield Units of different tissues overlap, it is difficult to determine reasonable bins. Therefore we divided the interval of $[-1024, 3000]$ into 20 equal sized bins. In the classification task every bin has been regarded as a single feature.

2.6 Lung Shape Comparison

Since we assume that the degree of pulmonary tuberculosis can correlate with the overall health of a patient, we considered the shape of the lungs, as well. We also assume that the difference between the shapes of the two lungs can provide information about the patient's health. To obtain a comparison measure, it is sufficient to look at the masks.

Note, that the following procedure needs all slices to be processed separately. In order to compare the different lungs, all masks have to be divided into two separate lung masks. Afterwards the contours of the relevant regions are calculated. In [17] a border following method is introduced to describe the contour of an object. The silhouettes of the lungs are calculated and stored in a vector of points using the findContours-method of OpenCV [3]. They can then be matched with OpenCV's matchShapes-method. A measure for the match can be computed by the following equation:

$$I(A, B) = \sum_{i=1}^{7} |m_i^A - m_i^B| \tag{1}$$

with

$$m_i^A = sign(h_i^A) \cdot log(h_i^A) \quad \text{and} \quad m_i^B = sign(h_i^B) \cdot log(h_i^B) \ ,$$

where h_i^A and h_i^B are the Hu Moments of the contours A and B [8].

Now there are comparison values for each slice of the CT Scan but it is desired to receive one representative measure for the match. So finally, the average value over all slices has to be calculated.

3 Classification Methods

We used different classifiers for predicting the severity score and the severity level of tuberculosis using the feature set obtained from the feature extraction step described in Section 2. In the training phase of classification models, we performed feature selection based on the cross-validation mean square error for severity score on the training set. We tried different classification methods including the multi-class support vector machine (SVM) with RBF kernel, the k-nearest-neighbor (kNN) algorithm, and the multi-layer perceptron classifier with different parameter settings. Below we describe the best classification models with respect to the mean square error on the training set and the way of predicting the severity score and the severity level of tuberculosis.

3.1 Decision Tree

Using the Chi Square method it turned out that 13 of the 17 features are the most meaningful. Apparently the histograms of the higher Hounsfield Units are not very informative. Therefore, all features have been considered for the decision tree, except for the histograms of ranges with values greater than 50.

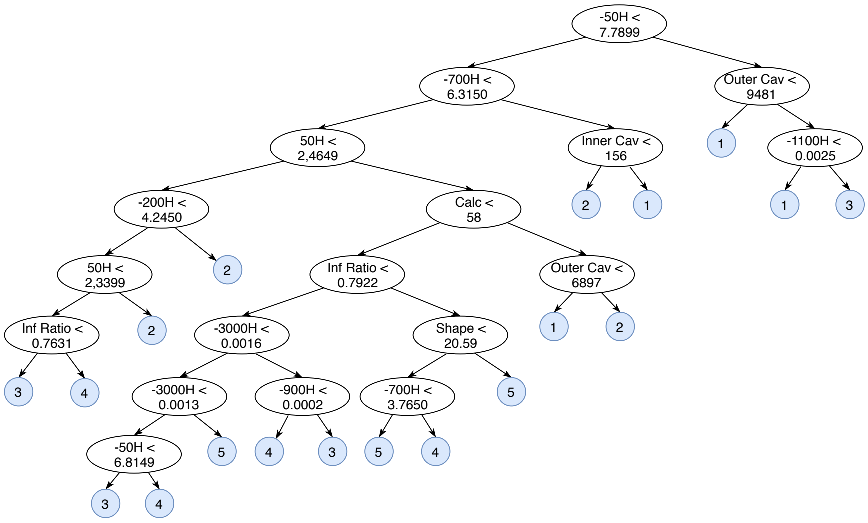


Fig. 3. Illustration of the decision tree using 13 features.

In Figure 3 the structure of the resulting decision tree is demonstrated. The numbers followed by an H stand for the ranges of the histograms. The assigned classes of the leaf nodes are highlighted in blue. Arrows to the left indicate that the condition of the parent node applies. Arrows to the right represent the non-applicability. The classifier was fitted using the Gini Impurity, a minimum fraction of 3% per leaf and a minimum quality gain of 0.01. The structure shows that most scores of the same severity class (”LOW”/”HIGH”) share similar features. But the scores 3 and 4 are often separated by only one property, as well. According to the paths of the decision tree, the distinction between score 3 and 4 is the most difficult task and probably represents the biggest source of errors regarding the AUC score, since both degrees belong to different severity classes.

3.2 Random Forests and Linear Regression

Using random forests and linear regression as classifiers we interpreted the prediction of the severity score and the severity level of tuberculosis as a regression problem. In our first attempt, we converted the severity level into the numbers 0 and 1, where 1 means ”HIGH” and 0 means ”LOW” severity level. We used the random forest classifier with the maximum depth value of 2 because the larger values led to overfitting of the classification model. We calculated the severity scores from the predicted severity levels by dividing the severity level values into intervals.

In the second test, we trained two separate classification models for the severity score and the severity level prediction. Since the severity score and the severity level values mismatched for some data items, we adjusted the severity score values depending on the corresponding severity level values at the extreme boundaries of the severity level. In particular, we set the severity score values to 1 if the corresponding severity level values were higher than 0.95. If the severity level values were below 0.22, we set the corresponding severity score values to 5 regardless the values that were predicted for the severity score before. Here, we also used the random forest classifier with the maximum depth value of 3. Additionally, we submitted a linear regression model which we trained on a subset of the training set for which we achieved the best results with respect to the cross-validation mean square error for the severity score. We used only a subset of the training set because linear regression is sensitive to outliers that we assumed in the training set due to variations in the mean square errors in different cross-validation runs.

Due to performance variations for the severity level in different cross-validation runs for the random forest model trained in the second test, we assumed that the classification model for prediction of the severity level overfitted on the training set. Therefore, in the third test, we reduced the maximum depth value to 2 for the random forest model. Furthermore, we refrained from the adjustment of the severity score values based on the severity level.

4 Evaluation and Results

A maximum of 10 runs could be submitted by each group per subtask in the ImageCLEF 2018 TB task. This section shows the final performance results of our feature-based approach in the severity scoring challenge (subtask 3). The final ranking was based on the root mean square error (RMSE) for the severity score. Table 1 summarizes the results for different runs of our approach ordered by the ranking provided by the subtask organizers. Additionally, we listed the results of the best runs with respect to the root mean square error (RMSE) and the Area Under the ROC Curve (AUC) submitted in the competition.

The best run of our approach was obtained by the random forest classification model with severity score adjustment described in the second test in Section 3.2 (indicated as Rnd_Frst_depth_3 in the table) using only three features: size of cavity, size of cavity tissue, and the infection ratio. Although our best run achieved the tenth rank regarding the RMSE measure, it was ranked sixteenth according to the AUC measure. In order to improve the results regarding the AUC measure too, we performed feature selection based on the cross-validation AUC value for the severity level on the training set using the same classification method. The so selected features were calcification, infection ratio, size of cavity, and the third, the sixth and the tenth bins of the histogram. Although this run was only ranked on the 25th placed regarding the RMSE, it achieved the eight place regarding the AUC measure.

Our second best run was achieved by the random forest classification model with the maximum depth value of 2 without severity score adjustment and the linear regression model trained on the subset of the training set (indicated as Rnd_Frst_depth_2 and Lin_Reg_part in the table) on the same feature subset as our best run. The performance results of our approach that calculated the severity score from the predicted severity level values by the random forest classification model (indicated as Rnd_Frst_score_by_level in the table) were the worst among the regression based approaches described in Section 3.2. Although the AUC value for the severity level was the same as for our best run, the RMSE value for the severity score calculated based on the severity level was much worse than for the separate severity scores prediction model.

Table 1. Results for our top 5 runs for Subtask 3 – Severity scoring.

Classification model	Features	RMSE	$\text{Rank}_{\text{RMSE}}$	AUC	Rank_{AUC}
–	–	0.7840	1	0.7025	6
–	–	0.8934	5	0.7708	1
Rnd_Frst_depth_3	cav., cav. tissue, inf. ratio	0.9626	10	0.6484	16
Rnd_Frst_depth_2	cav., cav. tissue, inf. ratio	0.9768	13	0.6620	13
Lin_Reg_part	cav., cav. tissue, inf. ratio	0.9768	14	0.6507	15
Rnd_Frst_depth_3	calc., inf. ratio, cav., hist. bins 3,6,10	1.1046	25	0.6862	8
Rnd_Frst_score_by_level	cav., cav. tissue, inf. ratio	1.2040	29	0.6484	17

The decision trees unfortunately performed worst. They only achieved the ranks 32-34 regarding the RMSE measure. The severity class was determined on the basis of the received scores. For this, two methods were used. In the first approach, the values 1, 2 and 3 represent the class "HIGH", and 4 and 5 belong to class "LOW". In the other method the probability p of a high severity was calculated by the formula $p = \frac{5-\hat{y}}{4}$, where $\hat{y}$ stands for the predicted severity score of the decision tree. The results showed that the first method scored significantly better AUC values. Our best decision tree even reached the ninth rank in regard to the AUC measure.

5 Conclusion

In this paper we have shown that our feature-based approach is competitive to other participants of the ImageCLEF 2018 challenge [6]. With our best methods we achieved rank 10 regarding the RMSE and rank 8 regarding the AUC measure. Almost all features in our approach were extracted using the lung masks provided by the organizers of the task. These masks were created by an automatic segmentation algorithm [7] that failed to recognize especially large lesions in the lungs in some cases. Consequently, our feature extraction algorithms also failed to work in such cases. Therefore, we assume that an optimization of the masks could lead to a more precise feature extraction and improvement of the final results of our approach. As the reproduction of Hounsfield Units of CT scanners may vary, further information about the hardware and the used parameters could lead to an improvement of the results. These could also be used to determine reasonable bins for the Hounsfield histograms.

Finally, the feature choice in our approach was for the most part based on own observations that could be approved by medical studies published in the literature in recent years. We believe that we could improve the feature extraction and consequently the final results of our approach by consulting medical experts specialized in treatment of pulmonary tuberculosis.

References

1. World health organization. Website (2018), http://www.who.int/en/news-room/fact-sheets/detail/tuberculosis; visited on 31. May 2018.
2. Ball, R., Greene, C., Camp, J., Rowntree, L.: Calcification in tuberculosis of the suprarenal glands: Roentgenographic study in addison's disease. Journal of the American Medical Association **98**(12), 954–961 (1932)
3. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
4. Brooks, R.: A quantitative theory of the hounsfield unit and its application to dual energy scanning. Journal of Computer Assisted Tomography **1**(4), 487–493 (1977)
5. Cropp, R.J., Seslija, P., Tso, D., Thakur, Y.: Scanner and kVp dependence of measured CT numbers in the ACR CT phantom. Journal of Applied Clinical Medical Physics **14**(6), 338–349 (2013)

6. Dicente Cid, Y., Liauchuk, V., Kovalev, V., , Müller, H.: Overview of ImageCLEF-tuberculosis 2018 - detecting multi-drug resistance, classifying tuberculosis type, and assessing severity score. In: CLEF2018 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org <http://ceur-ws.org>, Avignon, France (September 10-14 2018)
7. Dicente Cid, Y., Jiménez del Toro, O.A., Depeursinge, A., Müller, H.: Efficient and fully automatic segmentation of the lungs in ct volumes. In: Proceedings of the VISCERAL Anatomy Grand Challenge at the 2015 IEEE International Symposium on Biomedical Imaging (ISBI). pp. 31–35. CEUR-WS (2015)
8. Hu, M.K.: Visual pattern recognition by moment invariants. IRE Transactions on Information Theory **8**(2), 179–187 (1962)
9. Jankowski, M.: Erosion, dilation and related operators. In: Proceedings of 8th International Mathematica Symposium (2006)
10. Koegelenberg, C., A Balkema, C., Jooste, Y., Taljaard, J., Irusen, E.: Validation of a severity-of-illness score in patients with tuberculosis requiring intensive care unit admission. South African Medical Journal **105**(5), 389–392 (2015)
11. Nin, C., de Souza, V., Alves, G., do Amaral, R., Irion, K., Marchiori, E., Hochhegger, B.: Solitary lung cavities: Ct findings in malignant and non-malignant disease. Clinical radiology **71**(11), 1132–1136 (2016)
12. Ong, C.W., Elkington, P.T., Friedland, J.S.: Tuberculosis, pulmonary cavitation, and matrix metalloproteinases. American journal of respiratory and critical care medicine **190**(1), 9–18 (2014)
13. Parkar, A., Kandiah, P.: Differential diagnosis of cavitary lung lesions. Journal of the Belgian Society of Radiology **100**(1) (2016)
14. Richter, A., Hu, Q., Steglich, D., Baier, K., Wilbert, J., Guckenberger, M., Flentje, M.: Investigation of the usability of conebeam ct data sets for dose calculation. Radiation Oncology **3**(1), 42 (2008)
15. Ridler, T., Calvard, S.: Picture Thresholding Using an Iterative Selection Method. IEEE Transactions on Systems, Man and Cybernetics **8**(8), 630–632 (1978)
16. Subburaj, K.: CT Scanning –Techniques and Applications. IntechOpen (2011)
17. Suzuki, S., Abe, K.: Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing **30**(1), 32–46 (1985)
18. Vorster, M., Allwood, B., Diacon, A., Koegelenberg, C.: Tuberculous pleural effusions: Advances and controversies. Journal of Thoracic Disease **7**(6), 981–991 (2015)

3.2 Feature and Deep Learning Based Approaches for Automatic Report Generation and Severity Scoring of Lung Tuberculosis from CT Images

In the previous Chapter 3.1, we presented a framework that is able to forecast the severity score based on image data only. The results are promising, especially compared to related work, described in the previous publication. However, there is still room for improvement. To close the gap in the following chapter, we will improve our results by further investigations including meta information such as the position of the affected lung, presence of calcification, presence of caverns, pleurisy and lung capacity decrease. All these features are part of a common CT report. Thus, we will extend our framework for the previously presented ability of severity scoring by the new feature, namely the automated CT report generation. Both tasks are part of the ImageCLEFmed Tuberculosis challenge and therefore originated by medical experts. Besides our research on custom features, we will investigate the possibility of a feature-independent processing. Therefore we engaged in the development of a novel DL approach for CT report generation as well as severity scoring. This could also be helpful for physicians during manual diagnosis while providing less predictable insights. Both approaches will be discussed in the further course.

Feature and Deep Learning Based Approaches for Automatic Report Generation and Severity Scoring of Lung Tuberculosis from CT Images

Kirill Bogomasov, Daniel Braun, Andreas Burbach, Ludmila Himmelspach, and Stefan Conrad

Heinrich-Heine-Universität Düsseldorf, Institut für Informatik
Universitätsstraße 1, 40225 Düsseldorf, Germany
{bogomasov, daniel-braun,andreas.burbach,
ludmila.himmelspach,stefan.conrad}@hhu.de

Abstract. The paper presents two approaches for automatic Computed Tomography (CT) report and tuberculosis (TB) severity scoring which were two subtasks of ImageCLEFtuberculosis 2019 challenge. While our first approach uses image processing techniques for feature extraction from CT scans, our second approach uses artificial neural networks (ANN) for predicting probabilities for different lung irregularities associated with pulmonary tuberculosis and tuberculosis severity assessment. The results showed that our feature-based approach is still a competitive method that achieved rank 3 of 54 in the severity scoring subtask and rank 7 of 35 in the CT report subtask.

Keywords: automatic CT report · tuberculosis severity scoring · medical image classification · feature extraction · deep learning

1 Introduction

The tuberculosis task [5] of the ImageCLEF 2019 [10] challenge consisted of two subtasks dealing with analysis of Computed Tomography (CT) images of patients suffering from pulmonary tuberculosis. The aim of subtask #1 was the tuberculosis severity assessment based on CT scans. The subtask #2 was dedicated to the automatic generation of a CT report including the information about the left and right lung affection, presence of calcifications, presence of caverns, pleurisy, and lung capacity decrease. Both subtasks shared the same data set consisting of CT images and additional patient's meta data including information about education, imprisonment, disability, comorbidity, and others.

Last year our team participated in the severity scoring subtask at ImageCLEFtuberculosis 2018 challenge [6]. Our feature-based approach achieved rank

10 of 36 regarding the RMSE measure [2]. This result showed that our methods could compete with more complicated and computationally intensive methods in the field of deep learning. Since our feature-based approach provided a descriptive image classification framework, we decided to improve and to adapt it to the requirements of both subtasks of the ImageCLEF 2019 challenge [5]. On the other hand, taking the last years research trends into account, we developed a new deep learning-based approach.

2 Feature Based Approach for Automatic CT Report Generation and Tuberculosis Severity Scoring

In this section we describe our feature-based approach for automatic CT report and severity score prediction from CT scans. The main motive for developing a feature-based approach was the ability not only to predict the probabilities for different lung irregularities but also the ability to mark them in CT scans. This could also be helpful for physicians during manual assessment of CT scans. Furthermore, our approach provides information about the influence of different lung damages and additional patient's data on the tuberculosis severity score.

2.1 Preprocessing

Some features that we used for the automatic CT report were extracted from the original CT scans, while other features were easier to extract from binary images. Therefore, we binarized all CT scans using IsoData method [13]. We used lung masks for extraction of all features for the CT report task. Some of the lung masks that were provided by the organizers of the task [7] still did not cover large lesions. For this reason we decide to use our own lung masks extracted by the segmentation algorithm described in [4]. This algorithm examines the silhouettes of extracted masks for irregularities and reconstructs the masks. Although the reconstructed lung masks did not perfectly cover the entire lung, they still contained more lung pixels than lung masks provided by the organizers of the task.

2.2 Automatic CT Report Generation

Presence of Calcification Pulmonary calcification in CT scans was determined for left and right lung separately depending on the number of pixels that were identified as part of calcification. Since different Hounsfield Unit (HU) ranges for pulmonary calcification in CT scans were proposed in the literature [3, 8, 12] and the Hounsfield Units were not standardized in CT scans in the data set, we decided on a relatively large range between 300 HU and 3000 HU. In this way, we were able to identify calcifications of different density. On the other hand, our range for calcification contains the HU range for bones that were often erroneously covered by the lung masks. To reduce the presence of bones in the examined lung area, we adjusted the lung masks in a preprocessing step by removing pixels of their boundaries along the z-axis using morphological erosion function [11] with a disk of radius four pixels. Since many CT scans contained noise patches that could be erroneously classified as calcified nodules, we removed all objects smaller than 10 pixels that were identified as calcifications. Finally, we added up the pixels of found calcifications over all CT scan slices along the z-axis in the file. If either left or right lung or both contained more than 400 calcification pixels, we stated the probability of presence of lung calcifications as 1 otherwise as 0. This threshold value was determined based on the cross-validation Area Under the ROC Curve (AUC) value for presence of calcification on the training set.

Since Hounsfield Unit range for plastic and metal overlaps our range for calcification, our method for detection of calcification presence tended to false positives for patients that had medical appliances in the lung. To prevent misclassifications in such cases, the shape of found calcifications could be additionally examined.

Presence of Caverns At ImageCLEFtuberculosis 2018 [6], we used a simple approach for detection of pulmonary caverns. The principal idea of the method was detecting caverns as dark spots surrounded by light tissue in binarized CT image slices along the z-axis [2]. The main weak point of our approach was that trachea and bronchi were incorrectly recognized as caverns. Therefore, we cut out the middle part of the lung to avoid false positives. Unfortunately, that workaround has led to many false negatives because our method did not detect caverns that were either completely or partly located in the cut out part of the lung. For this reason we improved our last year approach for detection of pulmonary caverns by examining the entire lung.

The Fleischner glossary defines pulmonary cavities as thick-walled gas-filled spaces [9]. The main difference to trachea and bronchi is that cavities are completely covered by cavity walls. Therefore, we validated a cavern in a binarized CT scan slice along the z-axis as such only if its pixels were detected as pixels of a cavern in the CT scan slices along the x- and y-axes. We estimated the volumes of pulmonary caverns and their walls for right and left lung separately by adding up the pixels of validated cavities and cavity walls over all CT image slices along the z-axis. We used these four features for training a linear regression model for predicting the presence of caverns.

Our improved method reliably detected caverns in CT scans in the training set as long as the distances between the slices in the scans were not too large so that all cavity walls were depicted in the CT images. Unfortunately, our approach still produced false positives due to artifacts on the images mainly caused by the heartbeat of patients. Therefore, an additional preprocessing step is needed for elimination of artifacts in CT scans.

Presence of Pleurisy Pleurisy is inflammation of pleura which is a thin membrane that covers the lungs [1]. Since inflammation often leads to thickening of the tissue and pleura thickening increases the distance between the lung and bones, in our approach for pleurisy detection, we compared the average distance between the boundaries of the lung masks and bones in images along the z-axis in patients with and without pleurisy. For that purpose we overlayed the lung masks and the bone masks which represent pixels of the original CT scan with Hounsfield Units between 300 and 3000. In the resulting image, we calculated the average distance between pixels of the lung mask boundaries and the nearest bone pixels. Then we averaged the distances between lung and bones over all CT scan slices along the z-axis for right and left lung separately and used them for training a linear regression model for pleurisy prediction.

Lung Capacity Decrease The lung capacity is the maximum amount of air that the lung can hold. Some kinds of lung tissue damage caused by Mycobacterium tuberculosis (MTB) bacteria may decrease the capacity of the lung. Since an automatic detection and classification of different types of lung lesions from CT scans is a challenging problem, we predicted the probability of the lung capacity decrease based on the estimated ratio of the lung tissue to the entire lung volume. Assuming that the lung tissue ratio compared to the lung volume is larger in patients with decreased lung capacity than in patients with normal lung capacity, in our approach, we did not differentiated between healthy and damaged lung tissue. Similar to our last year approach [2], we calculated the ratio of the lung tissue as a relation of white pixels in the binarized CT image to the number of pixels in the lung mask averaged over all slices along the z-axis. Finally, we trained a linear regression model for lung capacity decrease prediction using the ratios of the lung tissue for left and right lungs as features.

Right and Left Lungs Affected Mycobacterium tuberculosis (MTB) bacteria causes more kinds of lung damage than calcifications, caverns, pleurisy, and lung capacity decrease. Therefore, the estimation model for probability of lung affection based on the probabilities for lung damage described before did not achieve satisfactory results on the training set. On the other hand, raw feature values that we extracted for predicting the probability of aforementioned lung damage, provided more information about further lesions in the lung. For this reason, we used the number of calcification pixels in the lung, average distance between the lung and bones, and the ratio of lung tissue to the lung volume for left and right lung, separately, as features for training random forests models for predicting the probabilities of affection of lungs.

2.3 Tuberculosis Severity Scoring

At ImageCLEFtuberculosis 2018 [6], our system achieved its best results for tuberculosis severity score prediction using three features: the cavern volume, the volume of cavern walls, and the infection ratio [2]. This year we used data from the CT report task combined with provided patient's meta data. Using linear regression as classifier, we obtained the 5-fold cross-validation AUC of approximately 0.8 for severity score on the training set. The most important features for severity score prediction were the probability of left and right lung affection, information about the imprisonment, the probability for pleurisy, and information about education. Although some features seemed to play an insignificant role, their elimination diminished the AUC value for severity score. Since some features from meta data were very important for severity score prediction, we tested our linear regression model on the training set only using patient's meta data. We obtained AUC value of approximately 0.75. On the other hand, the linear regression model trained only using data from CT report achieved the same AUC value. Although we were aware that feature values predicted for the CT report task were inaccurate to some degree, we used them combined with provided patient's meta data for training a linear regression model for TB severity score prediction.

3 Deep Learning Based Approach

Deep Learning has been applied on solving medical relevant research questions. Among other things it is used for classification of brain and lung tumors. Thus, Liu and Kang [17] for example achieves an AUC value of 0.981 with their ANN on the LIDC-IDRI data set [18] for the binary classification of lung cancer.

In addition to the classification of the CT scans into the predefined disease stages, the task can be subdivided into a further subtask, namely the segmentation. We suspect that the occurrence of disease-typical symptoms, such as calcification, caverns and pleurisy, may help in the subsequent classification. The topic of the localization and classification of objects is the subject of many scientific publications.

Some of the most promising approaches are based on the U-Net architecture [15]. This is shown, for example, by the fact that the winner of the 2018 BraTS Challenge used a U-Net variant [16]. The BraTS data set contains of CT scans of brain tumor patients and is therefor similar to the given tuberculosis data.On the one hand an advantage of the U-Net architecture is that the network considers the semantic context of the entire image during segmentation, on the other the hand U-Net architecture needs only a small amount of training examples to produce good results. Regarding the low amount of training data of the two tuberculosis tasks, this is a sufficiently important feature. We will use one architecture for both tasks, severity scoring and CT report, with the only difference being the number of final classifications to represent the different amount of possible labels. Isensee et al. showed that the architecture of the U-Nets is already so high-performant that a meaningful pre- and post-processing offers a greater potential for improvement than the change of the architecture [14]. Therefore, we start our processing pipeline with preprocessing and extend the architecture of the original U-Net [15] by an additional classification CNN. Afterwards we finish our approach with postprocessing. The exact explanation follows in the next sub-chapters.

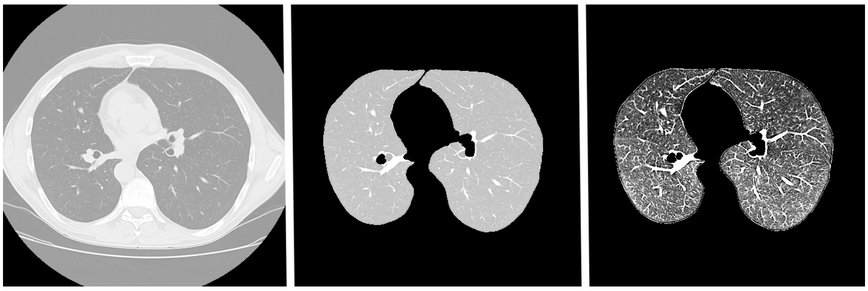


Fig. 1. Left: No preprocessing. Middle: Only segmentation. Right: Full preprocessing.

3.1 Preprocessing

The data set contains several anomalies which make preprocessing necessary. The CT scans in the given data set have 3 different values $\{-3024, -2048, -1024\}$ for "outside of body" - mark. Probably because the images were taken by different scanners and are not standardized. For this reason, some serious jumps can be found in the value ranges of the Hounsfield Units. Beside of that, there are even higher values for some noisy pixels. Similar to [19], we used a four-stage preprocessing to standardize the CT scans.

- Step 1: Remove empty gap. "NULL"-representing pixel values outside of body are often much lower than the values inside. To prevent that no area of the examination remains empty, each "NULL" - representing pixel is replaced with the next higher value.
- Step 2: Removing noise by range limits. The new value range is limited to [-1000; +2000]. Outside pixel values are set to the limit value.
- Step 3: Min-max normalization to [0,1].
- Optional Step 4: In the following the lung area is segmented with the binary masks from the original data set 1. Finally we reduced the image size by removing "0"-values in border area.

Figure 1 shows the three options of preprocessing.

3.2 Architecture

As mentioned previously, our chosen architecture is based on the original U-Net approach, but we changed the original 2D convolutional layer to 3D. Additionally we added a final classification CNN, based on the well known VGG19 architecture [20], for a binary output, since we have a two-classes problem. Figure 2 shows a draft of the resulting network architecture.

During the training and in the later classification we limit the input to 16-slice sliding windows, which contains coherent slices along z-axis, for two reasons. First, this reduces the requirements on GPU memory. Second, we now have a fixed input depth without the need to scale it. This not only serves to reduce

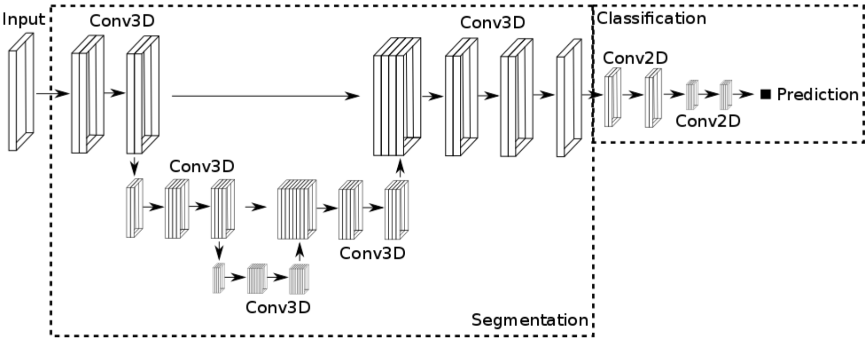


Fig. 2. The architecture of the proposed network.

the requirements on GPU memory, but has also proven to be a useful value to enhance the precision. Complementary, a more accurate prediction is produced, because of a several classification results for each image. In the next step, we halve the image, separating it into left and right lungs. This distinction is not taken into account during training. Finally, we scale the input data to 192 × 256 with a bilinear interpolation. This results in an input tensor of 192 × 256 × 16.

For the U-Net, as segmentation network, we chose a depth of four with a number of eight filters for the first convolutional layers. We use maxpooling for the downscaling path and a transposed convolution for the upscaling path. Furthermore, batch normalization is applied after each convolution and a dropout value of 30% for the last convolutional layer in the downscaling path. As activation function we use the rectified linear unit. To get our segmentation mask we use a convolutional layer with filter size one in each direction. For task one this results in one segmentation mask, due to the fact that we have a binary classification. Contrary to that, we use five segmentation masks for task two. Even though there exist six labels in the task, we only need one probability to distinguish the affection of the left or right lung due to the splitting of the lung as preprocessing. An example of different segmentation masks for task two can be seen in Figure 3.

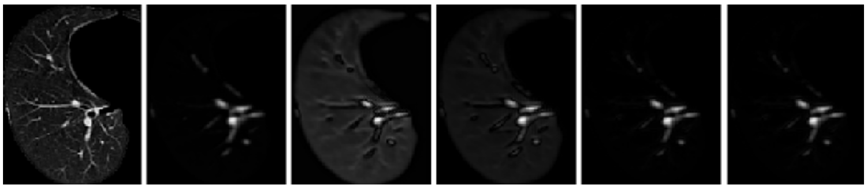


Fig. 3. The left image shows the input slice. All others show the activations in the different segmentation masks.

Algorithm 1 Definition of Max-Rule

```
Require: $\tau \in \mathbb{N}$
  if $|D_{left} - D_{right}| > \tau$ then
      $P \Leftarrow max(S_{pos} \cup S_{neg})$
  else
      if $|S_{pos}| = |S_{neg}|$ then
          if $1 - max(S_{pos}) < min(S_{neg})$ then
              $P \Leftarrow max(S_{pos})$
          else
              $P \Leftarrow min(S_{neg})$
          end if
      else
          if $|S_{pos}| > |S_{neg}|$ then
              $P \Leftarrow max(S_{pos})$
          else
              $P \Leftarrow min(S_{neg})$
          end if
      end if
  end if
  return $P$
```

The segmentation mask is used as input in our final classification CNN. For the CNN we also use a depth of four with eight as number of filters for the first convolutional layers. Like for the segmentation network, batch normalisation for all and a 30% dropout for the last convolutional layer are applied. A leaky rectified linear unit is used as activation. The final layer is a dense layer with one neuron to represent the probability of the label. In task one we have one classification network, but for task two we use five independent classification networks, one for each label.

3.3 Postprocessing

The network predicts the class of 16-slice windows of the CT scan. To get an overall prediction P for a whole CT-scan, an aggregation of a set of predictions has to be made. Therefore we divide each CT scan into three sections of same size. For each of these sections a prediction p_i with $\{p_i \in \mathbb{R} | 0 \leq p_i \leq 1 \ \wedge i \in \{1, \ldots, 6\}\}$ is calculated. Taking into account the left and right half, we get a total of six results.

Now we propose four methods to merge these six partial results p_i into one final result P.

1. Average: The result is defined as $P = \overline{\{p_i | i \in \{1, \ldots, 6\}\}}$, namely the average prediction value over all partial predictions.
2. Max-Rule: For this rule we define D_{left} and D_{right} as the number of lung slices in z-direction of the left respectively right lung. Also let S_{pos} be the set of positive predictions for which holds that $p_i \geq 0.5$ with $i \in \{1, \ldots, 6\}$.

Similary, S_{neg} is the set of negative predictions defined as $S_{neg} = \{p_i < 0.5 | i \in \{1, \ldots, 6\}\}$. Like Algorithm 1 shows, we first check if part of the lungs is missing. This occurs due to the fact that the size of the left and the right lung can diverge due to the preprocessing while reducing the zero values at the image borders. Consequently, we make the assumption that this difference is a sign of serious illness. Therefore, if the difference between D_{left} and D_{right} exceeds a threshold $\tau \in \mathbb{N}$, the maximal partial prediction value p_i is chosen as probability. If the depth of the lung does not differ too much, we let the majority decide and therefore choose the maximum respectively the minimum value from the set, S_{pos} or S_{neg}, that has more elements. If the two sets have an equal amount of elements, the value with the smallest distance to the respective target value 0 or 1 is chosen.

3. Average-Rule: Similar to Max-Rule, the only difference is that the calculation of the resulting prediction value P does not select the maximum or minimum but the average over all values of the corresponding result set S_{pos} respectively S_{neg}.
4. Confidence correction: For each window of a CT scan from the validation data set, consisting of 16 slices, the coefficient which is necessary to change the prediction of the respective window, is calculated so that the classification result is the correct class.

4 Evaluation and Results

This section shows final performance results of submitted runs in the severity scoring (subtask #1) and CT report (subtask #2) challenge. The final ranking in the severity task was done based on the Area Under the ROC Curve (AUC) value, while the final ranking in the CT report task was done based on the average AUC value. Table 1 summarizes the results for Top-10 submitted runs with the highest AUC value and the best run for our deep learning-based approach for

Table 1. Short overview of submitted runs for subtask 1 – Severity scoring.

Group name	Run	AUC	Accuracy	Rank
UIIP_BioMed	SRV_run1_linear.txt	0.7877	0.7179	1
UIIP	subm_SVR_Severity	0.7754	0.7179	2
HHU	**SVR_HHU_DBS2_run01.txt**	**0.7695**	**0.6923**	**3**
HHU	SVR_HHU_DBS2_run02.txt	0.7660	0.6838	4
UIIP_BioMed	SRV_run2_less_features.txt	0.7636	0.7350	5
CompElecEngCU	SVR_mlp-text.txt	0.7629	0.6581	6
San Diego VA HCS/UCSD	SVR_From_Meta_Report1c.csv	0.7214	0.6838	7
San Diego VA HCS/UCSD	SVR_From_Meta_Report1c.csv	0.7214	0.6838	8
MedGIFT	SVR_SVM.txt	0.7196	0.6410	9
San Diego VA HCS/UCSD	SVR_Meta_Ensemble.txt	0.7123	0.6667	10
...	...	...	...	...
HHU	run_6.csv	0.6393	0.5812	27

Table 2. Short overview of submitted runs for subtask 2 – CT report.

Group name	Run	Mean AUC	Min AUC	Rank
UIIP_BioMed	CTR_run3_pleurisy_as_SegmDiff.txt	0.7968	0.6860	1
UIIP_BioMed	CTR_run2_2binary.txt	0.7953	0.6766	2
UIIP_BioMed	CTR_run1_multilabel.txt	0.7812	0.6766	**3**
CompElecEngCU	CTRcnn.txt	0.7066	0.5739	4
MedGIFT	CTR_SVM.txt	0.6795	0.5626	5
San Diego VA HCS/UCSD	CTR_Cor_32_montage.txt	0.6631	0.5541	6
HHU	**CTR_HHU_DBS2_run01.txt**	**0.6591**	**0.5159**	**7**
HHU	CTR_HHU_DBS2_run02.txt	0.6560	0.5159	8
San Diego VA HCS/UCSD	CTR_ReportsubmissionEnsemble2.csv	0.6532	0.5904	9
UIIP	subm_CT_Report	0.6464	0.4099	10
...	...	...	...	...
HHU	CTR_run_1.csv	0.6315	0.5161	12

severity scoring task. Table 2 lists the results for Top-10 submitted runs with the highest mean AUC value and the best run for our deep learning-based approach for CT report task. In the following subsection we describe the results for our approaches in detail.

4.1 Evaluation Results for the Feature Based Approach

Since we used results from the CT report task for TB severity score prediction, it is more sensible to start describing results for the CT report task. As highlighted in Table 2, our best run for the feature based approach was ranked on the seventh place. In this run we predicted the probabilities for lung irregularities as described in Section 2.2. In our second best run, we predicted the probability of presence of caverns only based on the number of cavern pixels in left and right lungs, separately, omitting the pixels of cavern walls. This run was ranked on the eighth place which is a worse result. Unfortunately, we did not receive the detailed evaluation results, so we can not comment on the performance of our approach regarding prediction of other lung irregularities.

In severity scoring task, the best run for our feature based approach was ranked on the third place among 54 submitted runs. In this run we predicted the severity score using patient's meta data and the results from our best run in CT report task. The prediction of severity score in our second best run was based on patient's meta data and the results from our second best run in CT report task. Although we did not submit a run for TB severity score predicted only on the basis of provided patient's meta data, the results for these two runs showed a positive impact of results from the CT report task on the tuberculosis severity score prediction.

Table 3. Deep Learning-based Approach for Severity Scoring.

Run name	AUC	Accuracy	Preprocessing	Postprocessing	Data
run_06	**0.6393**	0.5812	-	method 1	validation split
run_08 [1]	0.6258	**0.6068**	mixed	method 1	validation split
run_04	0.6070	0.5641	complete	method 1	validation split
run_07	0.6050	0.5556	complete	method 3, $\tau = 5$	all data
run_03	0.5692	0.5385	complete	method 3, $\tau = 10$	validation split
run_05	0.5419	0.5470	segmentation only	method 1	all data
baseline	0.5103	0.4872	complete	method 2, $\tau = 5$	validation split
run_02	0.4452	0.4530	complete	method 4	validation split

4.2 Evaluation Results for the Deep Learning Based Approach

For our evaluation we used different input data. We differentiated between train-/validation split and the complete dataset as training basis. The validation set consists of 10 images.

For Severity Score Task we set up the preprocessing, as shown in Table 3. For our runs, we used either full preprocessing, just segmentation or no preprocessing at all. *Run_08* is an exception, therefore we took an average of *run_5*, *run_6* and *run_7*. Table 3 shows the list of postprocessing configurations of each run.

The highest AUC score is achieved by *run_06*. In this case the network got the raw input data. We presume that the good AUC score is due to the fact that the network finds relevant points outside our region of interest, which is removed through preprocessing. This can be supported by the fact that the segmentation alone generates the worst results. However, the accuracy of *run_06* is lower than that of *run_08*. It is interesting that no neural network from those three, that we calculate the average on, can achieve such a high accuracy by itself. It seems that the networks found different features and learned differently, so in the connection they complemented each other and the accuracy increased. Surprisingly, with an accuracy of 0.453, *run_02* score performed significantly worse than the other constellations. Presumably, this is because of our validation set size of only 10 images, which is potentially too small. And thus, the calculated coefficients cannot be generalized.

Since we had only a limited amount of runs for CT reportings, we decided to use only those constellations, that were trained on the whole data set. Because it seemed to be more reasonable to train on more data. Table 4 shows the results. The greatest value for Mean AUC of 0.6315 and Min AUC share *CTR_run_1* and *CTR_run_2*. Compared to the third run, this shows, that for this task the preprocessing may be more valuable as for task 1. *CTR_run_3.csv* shows rather moderate results of 0.561 Mean AUC, which is still better than random, but still leaves space for improvement.

[1] conglomerate of run_5, run_6 and run_7

Table 4. Deep Learning-based Approach for CT Report.

Run name	Mean AUC	Min AUC	Preprocessing	Postprocessing	Data
CTR_run_1.csv	0.6315	0.5161	complete	method 1	all data
CTR_run_2.csv	0.6315	0.5161	complete	method 1	all data
CTR_run_3.csv	0.5610	0.4477	segmentation only	method 1	all data

5 Conclusion

In this paper we have shown that our feature-based approach is still competitive to our deep learning-based method and to methods of other participants of the tuberculosis task. Our best run achieved the third place regarding the AUC value in the severity assessment subtask and the seventh place regarding the mean AUC value in the CT report subtask. Although the results obtained by our approach are promising, we still see potential for improvement of our approach to achieve even better results in both subtasks.

Regarding that our neural network was not as deep as other networks in the literature, our results are promising. Especially the U-Net architecture seems to be beneficial and can be a good starting point for more research. Our preprocessing was only beneficial for subtask #2, which is surprising and therefore it would be interesting to investigate which parts of the lung had an effect on the resulting predictions. Data augmentation unexpectedly led to bad results in our first tests and we therefore refrained from using it. But we like to further investigate the usefulness of data augmentation for this task in combination with our network. Furthermore, we will test the network on other data sets, especially with segmentation data to train the U-Net separately. We hope that by this the segmentation layers will find meaningful areas, that can show us symptoms of such diseases. And regarding the results for subtask #2, more training epochs would be surely beneficial too and therefore the training will continue.

References

1. Berger, H.W., Mejia, E.: Tuberculous Pleurisy. Chest **63**(1), 88 – 92 (1973)
2. Bogomasov, K., Himmelspach, L., Klassen, G., Tatusch, M., Conrad, S.: Feature-Based Approach for Severity Scoring of Lung Tuberculosis from CT Images. In: Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum (2018)
3. Brooks, R.A.: A Quantitative Theory of the Hounsfield Unit and Its Application to Dual Energy Scanning. Journal of Computer Assisted Tomography **1**(4), 487–493 (1977)
4. Burbach, A.: Automatic Lung Extraction from CT Scans. Bachelor's Thesis (2018)
5. Dicente Cid, Y., Liauchuk, V., Klimuk, D., Tarasau, A., Kovalev, V., Müller, H.: Overview of ImageCLEFtuberculosis 2019 - Automatic CT-based Report Generation and Tuberculosis Severity Assessment. In: CLEF2019 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org, ISSN 1613-0073, <http://ceur-ws.org/Vol-2380/>, Lugano, Switzerland (September 9-12 2019)

6. Dicente Cid, Y., Liauchuk, V., Kovalev, V., , Müller, H.: Overview of ImageCLEF-tuberculosis 2018 - detecting multi-drug resistance, classifying tuberculosis type, and assessing severity score. In: CLEF2018 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org <http://ceur-ws.org>, Avignon, France (September 10-14 2018)
7. Dicente Cid, Y., Jiménez del Toro, O.A., Depeursinge, A., Müller, H.: Efficient and fully automatic segmentation of the lungs in ct volumes. In: Proceedings of the VISCERAL Anatomy Grand Challenge at the 2015 IEEE International Symposium on Biomedical Imaging (ISBI). pp. 31–35. CEUR-WS (2015)
8. Grewal, R.G., Austin, J.H.M.: CT Demonstration of Calcification in Carcinoma of the Lung. Journal of Computer Assisted Tomography **18**(6), 867–871 (1994)
9. Hansell, D.M., Bankier, A.A., MacMahon, H., McLoud, T.C., Mller, N.L., Remy, J.: Fleischner Society: Glossary of Terms for Thoracic Imaging. Radiology **246**(3), 697–722 (2008)
10. Ionescu, B., Müller, H., Péteri, R., Cid, Y.D., Liauchuk, V., Kovalev, V., Klimuk, D., Tarasau, A., Abacha, A.B., Hasan, S.A., Datla, V., Liu, J., Demner-Fushman, D., Dang-Nguyen, D.T., Piras, L., Riegler, M., Tran, M.T., Lux, M., Gurrin, C., Pelka, O., Friedrich, C.M., de Herrera, A.G.S., Garcia, N., Kavallieratou, E., del Blanco, C.R., Rodríguez, C.C., Vasillopoulos, N., Karampidis, K., Chamberlain, J., Clark, A., Campello, A.: ImageCLEF 2019: Multimedia retrieval in medicine, lifelogging, security and nature. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the 10th International Conference of the CLEF Association (CLEF 2019), LNCS Lecture Notes in Computer Science, Springer, Lugano, Switzerland (September 9-12 2019)
11. Jankowski, M.: Erosion, dilation and related operators. In: Proceedings of 8th International Mathematica Symposium (2006)
12. Khan, A.N., Al-Jahdali, H.H., Allen, C.M., Irion, K.L., Al Ghanem, S., Koteyar, S.S.: The calcified lung nodule: What does it mean? Annals of Thoracic Medicine **5**(2), 67–79 (2010)
13. Ridler, T., Calvard, S.: Picture Thresholding Using an Iterative Selection Method. IEEE Transactions on Systems, Man and Cybernetics **8**(8), 630–632 (1978)
14. Isensee, F. et al.: No New-Net. In: Crimi, A., Bakas, S. (eds.) Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. pp. 234-244. Springer International Publishing (2019).
15. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. International Conference on Medical image computing and computer-assisted intervention. pp. 234-241. Springer (2015)
16. Isensee, F. et al.: Brain tumor segmentation and radiomics survival prediction: contribution to the BRATS 2017 challenge. In: International MICCAI Brainlesion Workshop. pp. 287-297. Springer (2017)
17. Liu, K., Kang, G.: Multiview convolutional neural networks for lung nodule classification. In: Int. J. Imaging Syst. Technol, vol. 27, pp. 12-22. Wiley (2017). https://doi.org/10.1002/ima.22206
18. Armato III et al.: Data From LIDC-IDRI. The Cancer Imaging Archive. 2015
19. Braun, D., Singhof, M., Tatusch, M., Conrad, S.: Convolutional Neural Networks for Multidrug-resistant and Drug-sensitive Tuberculosis Distinction. In: CLEF2017 Working Notes, CEUR Workshop Proceedings, Dublin, Ireland. CEUR-WS (2017)
20. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: arXiv 1409.1556. arXiv preprint (2014)

3.3 Orientation Estimation in MRI of Prostate Cancer Patients: When Simple Models Perform Better

Inspired by the success of the previously referenced ImageCLEFtuberculosis challenge, we set the goal to make a transition of our findings and algorithms to a productive environment. Convinced of our intent and in accordance with present necessities, the Institute of Radiology of Heinrich-Heine University has agreed to a cooperation. The pursued goal includes the idea of developing a Quality Assurance system that accompanies MRI-recordings procedure for prostate cancer patients and monitors the image quality during the acquisition. According to the agreement, the core research question, in this case, is **how can sagittal rotation in MRI of prostate cancer patients be estimated using CV?** This question has a prior importance for the reliability of diagnosis. In fact, standardized and controlled image acquisition is an essential component of QA. In general, different aspects may contribute to the quality of recordings. In particular, a correctly adjusted angle enables more precise recording and allows a better insight into the tissue. It also reduces the overlapping of organs in the field of view. Therefore the image acquisition has to be monitored, to avoid imprecise and flawed imaging. To answer this question, the data basis is changed again. DICOM recordings were provided for investigation. These images differ in that meta information, such as the optimal angle rotation, is presented. The optimal angle is not chosen arbitrarily, but as a consensus of several radiologists with many years of professional experience. Despite many efforts to automate QA processes in the medical application field, there is no work directly related to the issued RQ, which means that we are the first to deal with this important topic. This makes our work way more challenging but also more eminent. During our investigation, we tackle a series of real-world data challenges, such as unbalanced distribution in labeling, different image data origin and diverse image dimensions regarding the depth. All these difficulties are addressed in the following paper.

Orientation Estimation in MRI of Prostate Cancer Patients: When Simple Models Perform Better

1st Bogomasov Kirill
Department of Computer Science
Heinrich Heine University
Duesseldorf, Germany
bogomasov@hhu.de

2nd Grings Thomas
Department of Computer Science
Heinrich Heine University
Duesseldorf, Germany
thomas.grings@hhu.de

3rd Christian Rubbert
Department of Diagnostic and Interventional Radiology
Medical Faculty
Duesseldorf, Germany
Christian.Rubbert@med.uni-duesseldorf.de

4th Lars Schimmöller
Department of Diagnostic and Interventional Radiology
Medical Faculty
Duesseldorf, Germany
Lars.Schimmoeller@med.uni-duesseldorf.de

5th Conrad Stefan
Department of Computer Science
Heinrich Heine University
Duesseldorf, Germany
stefan.conrad@hhu.de

***Abstract*—Magnet Resonance Imaging (MRI) is an important modality in the diagnostic workup of prostate cancer. Poor image quality is critical for detection of tumors and classification according to the Prostate Imaging Reporting and Data System (PI-RADS v2.1). Because of that a precise image acquisition is highly important. Therefore a fully automated quality check is crucial.**

In this paper we present the first step of an automated multi-step quality check, which consists of deep learning-based orientation estimation for prostate MRI. It is a new field of application in terms of medical quality control. The proposed method achieves a mean absolute error of less than two degree regarding the optimal axial orientation based on the sagittal view. By this means, we achieve values which improve the axial orientation up to 36 % on provided examination data. The perfect setting enables the best possible viewing angle and reduces the risk of overlooking a tumor.

***Index Terms*—Image Orientation, Deep Learning, Quality Control**

I. Introduction

The demographic change affects society and individuals. The average life expectancy rises, the human population ages correspondingly. Advancing age is the most important risk factor for cancer[1]. Due to a rising age a higher percentage of people get cancer in their lifetime. This increases the burden on medical staff and also on the medical system [1]. An effective treatment both medicinal and operational requires the most precise diagnosis and tumor staging possible. The detection of prostate cancer is currently based on a combination of prostate-specific antigen measurement (PSA) and magnetic resonance imaging (MRI). While the PSA testing is almost standardized in these times, the image quality still depends on various setting parameters for example due to medical devices and external factors by means of patients characteristics. Magnetic Resonance allows organs to be seen in a detailed non-invasive way due to cross-sectional images. The acquisition of MRI sequences is planned manually for each patient. Due to the much higher in-plane resolution of 2D MRI when compared to 3D MRI, MRI of the prostate is usually acquired as 2D imaging in a sagittal, coronal and axial orientation with regard to the prostate. Structured reporting of prostate MRI examinations is conducted according to the Prostate Imaging Reporting and Data System (PI-RADS), which requires a division into different sectors and zones. The latter is significantly alleviated by imaging perfectly aligned to the transversal axis of the prostate. Experienced staff use sagittal images to plan the best possible transversal alignment. An automatic estimation of the optimal angle is a significant improvement. It saves time and reduces subjectivity since no more manual intervention is needed. To tackle this task we investigated several leading approaches for rotation angle estimation from different application fields. We adapted them to own needs and compared to our own algorithm. In this paper we present a new deep learning based approach which allows a reliable orientation estimation of transversal MRI of the prosta

II. Related Work

Angle estimation in 2D images is not a novel field of research. There are some established procedures in the field of document analysis. Often, those methods are applied in mobile applications for transfer of image data to text documents. In this case the image data needs to be rotated for the perfect crop of a rectangular shot. These methods mainly work with edges such as borders of the scanned documents, lines within text areas and other rather strong features like dark letters on a bright background [2]. Another, currently more popular application field is natural images, which is not very related in terms of context but is still much more researched in terms of orientation estimation. Joshi et al. [3] applied convolutional neural networks on real world photographs with the goal of determining the correct orientation. They defined

[1] https://www.cancer.gov/about-cancer/causes-prevention/risk/age

four categories for the task of 0°, 90°, 180° and 270° degrees and achieved an average accuracy of up to 98.5% with a pre-trained VGG-16 architecture. Cao et al. [4] used self made low-level features inspired by the biological simple cells of the visual cortex to estimate the overall image orientation of a natural image. They achieved comparable results. Fischer et al. [5] presented another deep learning approach for orientation prediction of arbitrary natural photographs. In contrast to prior works that just classify the orientation in different predetermined angles, the network regresses the orientation angle. The authors differentiate between three different levels of difficulty ±30°, ±45°, and the full circle.

The largest graphic cards manufacturer in the world NVIDIA, also took advantage of convolutional neural networks being able to estimate orientation of natural images. The presented CNN-based system learns detecting useful road features from front view image data combined with steering angle as the training signal [6]. The CNN architecture which was empirically found, has a depth of only nine layers, including a normalization layer, five convolution layers, three fully connected layers and is finalized with a Dense Layer. The network shows that smaller architectures are quite capable of solving given tasks and in some cases even do better than larger ones.

Several publications deal with estimation orientation in medical field of research: [7] presents a deep learning approach for estimating the fiber orientation from MRI data. The mean angular error is split into categories of 15° steps. Unfortunately the algorithm cannot be used for finer data with an entire angular spectrum. In addition, the results are evaluated on synthetic data and are neither comparable nor capable to the given task. Zhang [8] proposes an encoder decoder based approach for recognition of orientation in CMR images. Still the task is simpler because the rotation recognition is reduced to a classification problem with only 8 classes, 30° steps each.

The closest work in terms of medical data is presented by Baltruschat et al. [9]. The authors used transfer learning to take benefit from a very deep network, namely ResNet trained on ImageNet to determine the rotation of X-Ray images of human hands. The task is still much easier, since the bone information of hands alone is sufficient to calculate the rotation angles. Additionally a full rotation spectrum is not realistic in terms of prostate examination scenario. However, it makes the investigation elementary since the features are more clearly distinguishable.

In contrast to the previous work, our data is more complex and contains a much smaller range of orientation angles. To the best of our knowledge none of the published researches deal with such a fine spectrum of angles of only 33.5° within the presented dataset and also on such a complex application field.

III. Experimental Setup

The sequence of our processing pipeline is shown in Fig. 2. The input data was firstly pre-processed and augmented. In the following step a large selection of different neural network architectures was trained. Our choices were based on architectures of the State-of-the-Art and also on the selection from the related work. In the following steps test data was predicted, evaluated and graphically analyzed. We also trained a large number of custom architectures to outperform the previously generated results.

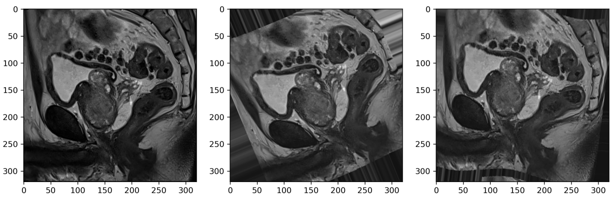


Fig. 1. (a) Original examintation, (b) shifted and rotated, (c) shifted, rotated and modified brightness

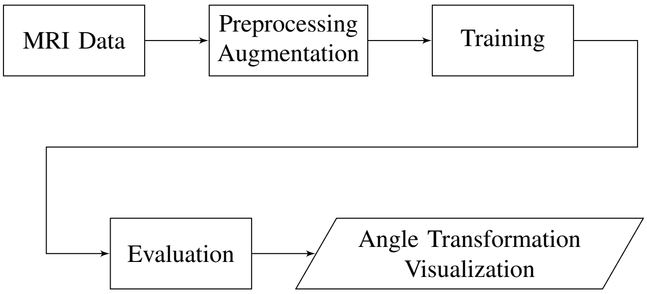


Fig. 2. Experimental workflow

1) MRI Data: Prostate MRI examinations from the Department of Diagnostic and Interventional Radiology of the Düsseldorf University Hospital were retrospectively included (01/2018 to 06/2019). An expert uroradiologist (10 years of experience) reviewed each examination for perfect alignment of the transversal imaging of prostate. Imperfect acquired examinations were excluded. Finally, a total number of 200 MRI examinations were included, each of which included the perfect alignment angle G. The given data is presented as DICOM. All recordings were made with Siemens scanners ”Prisma” and ”Skyra”. The recordings naturally differ in depth. The depth of the sagittal point of view varies between 18 and 36 slices. For normalization purposes we only consider the middle 18 slices of all images. Since the prostate is usually not visible on the beginning and ending slices of every MRI, we did not loose any information. This leads to a total of 3600 image slices that are used only for training and validation. For a fair split, the data was analyzed and split into groups in such a manner that as far as possible the whole spectrum of ground truth angles γ with $\{\gamma \in \mathbb{R} | 88.2^\circ \leq \gamma \leq 121.7^\circ\}$ is present in each of the groups. Finally 150 MRIs were used for training, 40 for validation and another 10 for prediction. The orientation γ within the test data was as follows $\{120.49, 113.00, 110.69, 109.49, 107.10, 104.59, 102.89, 101.00, 99.39, 96.89\}$

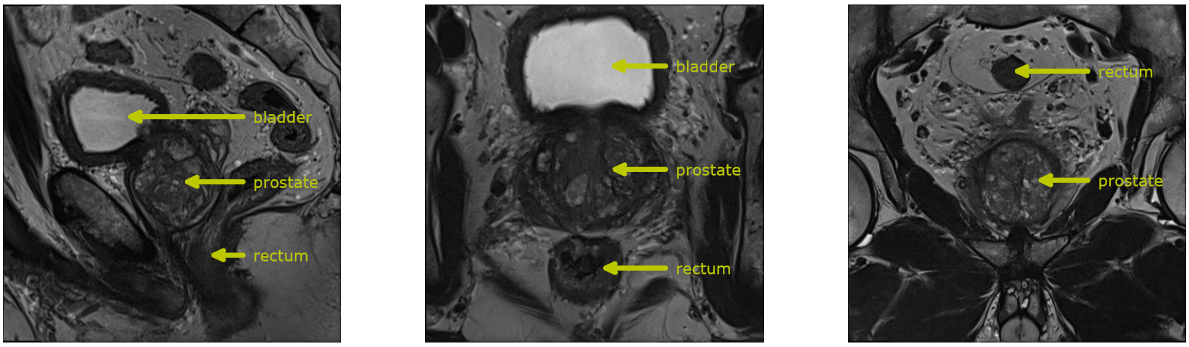


Fig. 3. (a) Sagittal view, (b) Transversal view, (c) Coronar view

2) Preprocessing and data augmentation: Firstly we prepared the image pixel values prior to augmentation, such as normalizing the value from [0, 4095] to [0, 1] by dividing all pixel values by the largest pixel value of each image. Secondly we scaled each image if its size was larger than default. 320 × 320 was chosen as default because it is the size of the smallest images of the dataset. Since we are working in most cases with Grey-scale data, the used image resolution is 320 × 320 × 1 . In cases of working with pre-trained networks, the input is transformed to RGB with 320 × 320 × 3 dimensional input by replicating the existing channel. Regarding the given data distribution (Fig. 4), we recognized that the data is overall highly unbalanced. To tackle this problem, we applied techniques of data augmentation. Therefore we used only methods which manipulate the input data in a way that the output is close to the real. Those methods are shifting, changing of brightness and distortion (Fig. 1). Since the dataset is rather small and the manual annotation is very labor intensive, we subsequently rotated the images to balance the angle distribution within the generated dataset as shown in Fig. 4. Doing so, we enlarged the existing images by five in total.

DICOM images contain among other things information about sagittal, transversal and coronal orientation. For given task only sagittal information is necessary by reason that it defines the orientation which is needed for a proper examination of prostate as can be seen in Fig. 3.

Furthermore the given task is regarded as a regression problem, since a subdivision of existing orientation angles into sub classes for a classification task would lead to a loss of accuracy and also a reduction of significance of the evaluation.

3) Baseline: We have selected several architectures of the gold standard and modified them to regression output. For this we replaced the head with fully connected layers ending with a single neuron. Our selection contains large models such as InceptionV3 [10], Xception [11] and ResNet50 [12] which proved outstanding accuracy in several application fields. Besides very deep solutions we went for compact and more computation-efficient architectures like ShuffleNet [13], MobileNet [14] and MobileNetV2 [15]. Their advantage is the ability to achieve comparable results with a smaller number of trainable parameters and being therefore more suitable for smaller datasets. In addition, we implemented and trained the architecture for self-driving cars which we mentioned in the related work and thus created a baseline. In the following we call it ”NVIDIA”. All used weights were the result of 500 epochs of training, since no further improvement in any experimental run with a higher epoch setting was observed.

4) Architecture: During experimental research, we investigated hundreds of different neural construction constellations which covered various depths, convolutional kernel sizes, normalization techniques, activation functions etc. Separately we searched for the greatest parameters using Auto-Keras [16]. Therefore, a varying amount of convolutional blocks {1, . . . , 6} with up to two layers either convolutional or separable convolutional, was build. We also worked with with different kernel sizes {(3, 3), (5, 5), (7, 7), (9, 9)} and dropout rates {0, 0.25, 0.5}. Finally, a flatten and global average pooling layers finalized the architecture search space.

The following architectures proved to be the most accurate for angle estimation and outperformed any architectures tested by Auto-Keras. The preprocessed and augmented data is fed to the input layer. The input layer is chosen to have the size of 320 × 320 × 1, since it is the size of the smallest image and also because we worked with single channel input data. Each convolution layer is followed by Batch Normalization, Max Pooling and Dropout. The rate for Dropout was set to 0.4. Experimentally, this value showed to be the best. Since Adam showed the greatest results in medical images of ten most popular optimizer [17], we applied it with the recommended learning rate of 0.01. For each of the seven convolution layers we used Mish as activation function, since it showed to be the most promising out of the set of activation functions. In contrast to the trend of only using small filter sizes of, for example, 3 × 3, we started with rather larger filters of 11 × 11 and kept the size at each level. Following this strategy we extracted more information from the adjacent pixels right from the start. The architecture is finalized by a flattening layer which is connected to a Dense layer with one neuron for regression output, powered by Softmax. The final architecture is shown in [18]. The batch size was set to 18, though all image slices of an examination are processed at once.

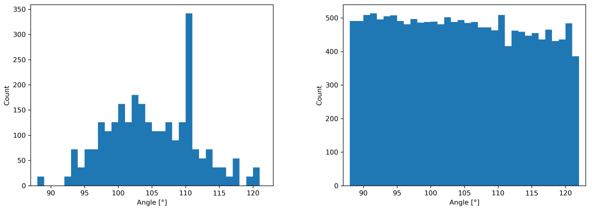


Fig. 4. (a) Angle destribution, (b) Balanced angle destribution

5) Prediction: The network regresses the angle for each of 18 slices of the MRI examination out of the dataset separately. The overall prediction P is calculated from the aggregation of a set of predictions p_i with $\{p_i \in \mathbb{R} | -1 \leq p_i \leq 1 \land i \in \{1, ..., 18\}\}$. The resulting angle is than defined as $P = \overline{\{p_i | i \in \{1, ..., 18\}\}}$, namely the average prediction over all partial results. Further the angle needs to be transformed to the 360° scale. Finally the mean absolute error $MAE = |G - P|$, where G is the ground truth angle, is calculated.

6) Visualization: Before the predicted angle p_i can be visualized, it firstly need to be transformed to a gradient g according to the following formula:

$$g = \frac{1}{\tan \arccos p_i} \tag{1}$$

Then the calculated gradient is used for visualization of the estimated orientation line.

Fig. 5 shows a few prediction examples. Both the ground truth and the predicted orientation are visualized. The distance between the ground truth line and the prediction line are visually marginal for image (a) with only 2.25° deviation of the ground truth value. Image (b) shows a large deviation of 5.47°.

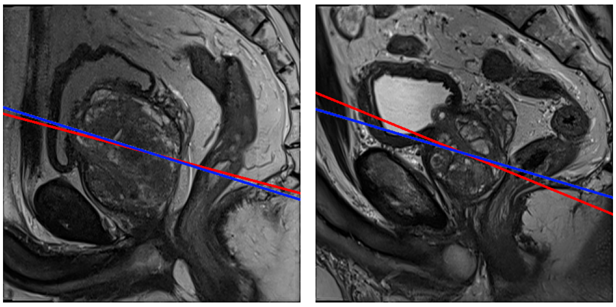


Fig. 5. Predicted angle (blue), ground truth (red)

IV. Results

Tab. I illustrates the results for all selected architectures. We used pre-trained weights for all architectures that these were available for, because transfer learning was proven to be effective for small datasets [19]. All these weights were generated on the ImageNet data.

Our evaluation differentiates between end-to-end training procedure and training where we freeze the body and retrain only the regression head. The MAE shows the mean absolute error for all predictions compared to the corresponding ground truth. Our goal was to get as close to the perfect orientation (ground truth) as possible. In terms of average prediction a random estimation value based on the dataset is valued by 6.19°. Several architectures reached this goal. Particularly noteworthy are NVIDIA with a MAE of 4.20°, MobileNet with 4.73° and our proposed architecture with 3.81°. It is also noticeable that networks with many millions of parameters performed significantly worse than smaller ones. In contrast to ResNet50 and InceptionV3 with ≈ 25 millions trainable parameters, our proposed architecture has almost one million parameters and NVIDIA's more than seven millions. We suspect that these numbers are approximately an optimal range of parameters for this kind of data.

The key elements that lead to great values were Batch Normalization [20] and Mish activation function [21]. We assume that Batch Normalization on one hand allowed higher learning rates and on the other added a little noise to the network. This leads to a decreased overfitting. Mish outperforms other already established activation function like ReLU on medical data because of its non-monotonic property, which helps stabilizing the networks gradient flow, hence preserving small negative values. This activation function behaves differently compared to commonly used ReLU of which the differentiation is 0 for negative values.

V. Discussion

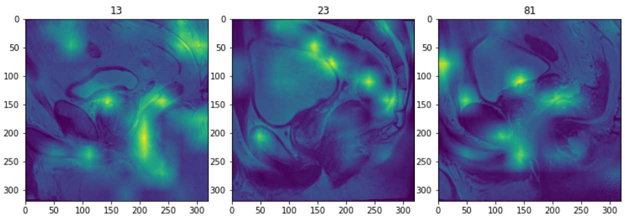


Fig. 6. Vizualization of neural activity

Fig. 6 shows heat maps that are created using our custom made architecture. Areas of highest neural activity are highlighted. It seems that rather dark regions or regions that are close to a high contrast change are the most relevant. As it happens, radiologists sometimes use the relative position of patient's rectum in relation to his prostate and mark the tilt angle orthogonally for angle estimation. We suspect a similar relationship based on the heat maps. Considering the small size of the test dataset, we want to emphasize that it was generated in such a way that the entire range of existing angles is present. This is obviously more convincing than a random split, since it contains not only angles with a smaller deviation which would tend to result in a more accurate prediction, but also those that have major differences to the ground truth.

In addition, it must be taken into account that even after balancing the data distribution of orientation angles using augmentation, in some cases only single images were available and therefore had been added multiple times to the set in

TABLE I
EXPERIMENTAL RESULTS

Architecture	Pre-trained	End to end	MAE	Min. Error	Max. Error	*Std*
InceptionV3	✓	✓	5.34°	0.51°	12.05°	3.90°
ResNet50	✓	✓	5.50°	0.73°	11.00°	3.60°
ResNet50Baseline	✓	✗	5.22°	1.30°	13.14°	3.33°
ShuffleNet	✗	✓	6.14°	0.65°	14.91°	4.01°
MobileNet	✓	✗	6.01°	1.63°	16.26°	3.91°
MobileNetV2	✓	✗	4.73°	0.00°	13.08°	3.96°
MobileNetV2	✗	✗	5.53°	0.34°	11.77°	3.44°
MobileNetV2	✓	✓	5.20°	0.50°	8.97°	2.89°
NVIDIA	✗	✓	4.20°	0.02°	12.96°	3.31°
Proposed model	✗	✓	**3.81°**	**0.67°**	**7.88°**	**2.68°**
Xception	✓	✓	4.82°	1.36°	10.13°	2.90°

a modified form as shown in Fig. 4. This tends to lead to overfitting. Hence, the ground truth data was annotated manually by medical professionals, there still may be small inexactness in the ground truth data. Therefore, the results are quite satisfying. Unfortunately, the dataset on which the NVIDIA architecture was evaluated is not publicly available. A test run of the self-created architecture on this would be quite informative. There are a few limitations of this work. On one hand we are facing a classical problem of a small dataset. The proposed architecture performed best on the validation dataset as well as on the presented test data. For a meaningful evaluation, we have split the data in such a way that the angle distribution is balanced in test data. However, we do not know whether the networks generalization ability is sufficient for angles outside the range we considered. On the other hand, all the data is made by only two scanners of the same manufacturer which share the same protocol. It is difficult to say whether the proposed network architecture would perform just as well on the input generated by another hardware. A simulation on synthetic data could provide some insights. For this a deep understanding of hardware and protocols is required.

In future work we will consider more rather rare edge cases with particularly large deviations from the optimum, which require additional data acquisition. In addition, we will try to include areas that are identified as particularly relevant through heat map information.

VI. Conclusion

To the best of our knowledge, this work presents the first deep learning approach for orientation estimation of prostate image data in MRI. In this paper we have shown that the architecture we have presented allows orientation estimation of prostate MRI data in a quality, which is at least as good as State-of-the-Art architectures in this research field achieve. We suppose that the information density of the training data, that the network needs to be able to carry out a correct regression is not large enough for very deep architectures. This might be the reason for large networks such as InceptionV3 or ResNet50 performing a little less accurate than mobile networks such as MobileNetV2, thus overfitting. The architecture that is proposed by NVIDIA's research team outperforms all tested networks on the MRI dataset except the model we propose.

We proved that convolutional neural networks are able to learn subtle features that are necessary for prediction of canonical orientation of MRI. We also found out that certain image areas are more important for orientation estimation. In the near future we will use these findings and investigate, whether preceding segmentation input of prostate will lead to improvement. Furthermore, we will look out for additional appropriate MRI data to be published, which we might be used as a second dataset for evaluation.

Finally, we will contemplate an MRI quality rating system, which will consist of several examination steps and operate while image registration. Firstly, axial orientation estimation, as introduced in this paper. Secondly, evaluation of sharpness, clarity and several other factors along with the decision, whether an injection of a contrast agent is necessary.

Acknowledgment

We would like to thank Christian Rubbert und Lars Schimmoeller for constructive feedback as well as the generous support and also for providing the data which made this work possible.

References

[1] K. Christensen, G. Doblhammer, R. Rau, and J. W. Vaupel, "Ageing populations: the challenges ahead," *The lancet*, vol. 374, no. 9696, pp. 1196–1208, 2009.

[2] M.-W. Lin, J.-R. Tapamo, and B. Ndovie, "A texture-based method for document segmentation and classification," *South African Computer Journal*, vol. 2006, no. 36, pp. 49–56, 2006.

[3] U. Joshi and M. Guerzhoy, "Automatic photo orientation detection with convolutional neural networks," in *2017 14th Conference on Computer and Robot Vision (CRV)*. IEEE, 2017, pp. 103–108.

[4] Z. Cao, X. Liu, N. Gu, S. Nahavandi, D. Xu, C. Zhou, and M. Tan, "A fast orientation estimation approach of natural images," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 11, pp. 1589–1597, 2015.

[5] P. Fischer, A. Dosovitskiy, and T. Brox, "Image orientation estimation with convolutional networks," in *German Conference on Pattern Recognition*. Springer, 2015, pp. 368–378.

[6] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[7] S. Koppers and D. Merhof, "Direct estimation of fiber orientations using deep learning in diffusion imaging," in *International Workshop on Machine Learning in Medical Imaging*. Springer, 2016, pp. 53–60.

[8] K. Zhang and X. Zhuang, "Recognition and standardization of cardiac mri orientation via multi-tasking learning and deep neural networks," in *Myocardial Pathology Segmentation Combining Multi-Sequence CMR Challenge*. Springer, 2020, pp. 167–176.

[9] I. M. Baltruschat, A. Saalbach, M. P. Heinrich, H. Nickisch, and S. Jockel, "Orientation regression in hand radiographs: a transfer learning approach," in *Medical Imaging 2018: Image Processing*, vol. 10574. International Society for Optics and Photonics, 2018, p. 105741W.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[11] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[13] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[16] H. Jin, Q. Song, and X. Hu, "Auto-keras: An efficient neural architecture search system," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1946–1956.

[17] M. Yaqub, F. Jinchao, M. S. Zia, K. Arshid, K. Jia, Z. U. Rehman, and A. Mehmood, "State-of-the-art cnn optimizer for brain tumor segmentation in magnetic resonance images," *Brain Sciences*, vol. 10, no. 7, p. 427, 2020.

[18] "Orientation estimation architecture," https://github.com/BogoK/OrientationEstimationArchitecture/blob/main/OrientationEstimationArchitecture.pdf, accessed: 2021-07-07.

[19] R. Barman, S. Deshpande, S. Agarwal, U. Inamdar, and M. Devare, "Transfer learning for small dataset," 03 2019.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[21] D. Misra, "Mish: A self regularized non-monotonic activation function," *arXiv preprint arXiv:1908.08681*, 2019.

4

Maritime Inventory Monitoring

"Learn how to see. Realize that everything connects to everything else"
— Leonardo da Vinci

4.1 A two-staged Approach for Localization and Classification of Coral Reef Structures and Compositions

As already shown in the previous chapter, the quality assurance and automation of processes is already promising for medical applications. However, there are some limitations. One of the reasons, as seen previously, is the data itself. Therefore, the following chapter will deal with the analysis of image sources, which differ greatly from the medical ones. Further insights into process automation and QA are expected from this breakdown.

In recent years, the changes in climate and nature have attracted great attention from researchers and public as well. The experts agree that protective actions are necessary. An appropriate intervention for environmental protection requires a continuing monitoring. One substantial environment is coral reefs. Coral reefs are highly endangered and need to be protected. Many marine biologists deal with this topic on a daily basis. The importance of early detection of changes on coral reefs is high (Fabricius and De'Ath, 2004). Some publications deal with the detection of damage to coral reefs in terms of coverage (Nurdin et al., 2015) or coral reef environmental change (Zhou et al., 2018). However, the total area covered by coral reefs is as large as 284 300 km^2 (M. Spalding et al., 2001). An area of this size can not be reviewed by experts manually. **This raises the question of whether an automatic localization and annotation of corals in large scale images is feasible.** In the following publication we tackle this question. In particular, we investigate how much computer vision can contribute to this task. During our further investigation and in contrast to the previous chapter, the image type changes from 3D images to 2D. In addition, we are moving away from classification towards object localization. In this context, we can not use previously introduced algorithms. Therefore, we needed new appropriate baseline architecture. At the starting point of our study, we could not find any related work even roughly similar to the RQs. Particular attention while searching for a baseline architecture was paid to the fact that the images are of a large scale, meaning a high resolution. Not many publications on the topic of object localization deal with this constriction regardless of data origin. High resolution images have special importance for satellite imaging and aerial imaging. This is why there is a demand to automatically localize the objects in this area (Ševo and Avramović, 2016), (W. Guo et al., 2018), (Tayara and Chong, 2018). Even though the area of application is different, the idea behind the mapping of the largest possible area to a small number is very similar. However, the algorithms developed for this kind of data are not applicable to coral images since the objects of interest in geospatial data are often of a homogeneous shape and mostly non-overlapping. For this reason, the rather simple methods for object detection are not transferable. In terms of maritime environment Object Detection is rather researched above the water surface (Farahnakian et al., 2018), (Makantasis et al., 2013), (Farahnakian et al., 2018). Several publications deal with underwater object detection using sonar imagery (Williams, 2011), (N. Wang et al., 2017), (Galceran et al., 2012), (Kim and S.-C. Yu, 2017), (Mandhouj et al., 2012) or remote sensing (Cao et al., 2016). However, these works focus on rather large objects with predominantly simple complexity. Speaking of recognition of large underwater objects (Rizzini et al., 2015), (Zhu et al., 2016) should not stay unmentioned since contrary to the previously mentioned publications, they deal with natural imaging. Nevertheless, the objects are of a similar large shape.

The provided ImageCLEF data set differs from standard data sets used for OD bench marking, such as COCO, in its significant complexity and small number of images. The complexity is caused by a highly variable object size, as well as a large number of tiny objects. What makes it even more difficult is the fact that the data is highly imbalanced. However, inspired by the success of the One-Stage-Detectors in several application fields (Jiao et al., 2019a), as well as our developments in the field of traditional feature engineering presented in previous chapters, which were able to keep up with the deep learning concepts, we decided to investigate how we can combine both techniques. In the following paper, we present a novel two-stage procedure for localization and classification. One of the main advantages of the proposed approach is that a detailed evaluation is possible. This allows a deep insight into strengths and weaknesses and thus a further discussion. Subsequently, we hope to draw attention to the complexity of the problem and raise research interest on the given application field.

A two-staged Approach for Localization and Classification of Coral Reef Structures and Compositions

Kirill Bogomasov, Philipp Grawe, and Stefan Conrad

Heinrich Heine University, Universitätsstraße 1, 40225 Düsseldorf, Germany
http://dbs.cs.uni-duesseldorf.de
{bogomasov,grawe,stefan.conrad}@hhu.de

Abstract. In this paper we present the approaches that achieved the first place in this years ImageCLEFcoral challenge. The task of the challenge was the localization and classification of corals within images of sea ground. Therefore we had to extract bounding boxes for each coral and labeling them with the specific type of substrate.
We applied a state-of-the-art deep learning approach (YOLO) and also developed a two-staged approach, using a grid along with two classifiers. One that classifies the tiles of the grid, the other that classifies the found boxes.
We had moderate results using YOLO and discovered that locating the corals is the most challenging part. Furthermore class imbalance and intersecting boxes, made the problem even harder.

Keywords: Image Segmentation · Image Classification · Object Localization

1 Introduction

Climate change is one of the major problems of the 21st century. Its impact is growing every year and therefore researched a lot. Since corals are a significant part of the maritime environment they are affected by the climate change in many ways [6]. Corals have their own self-contained and over many decades developed ecosystem, which is why the influence of damage to coral reefs can have serious consequences for every maritime organism. Every year the danger of complete destruction of coral reefs becomes more realistic. To sophisticatedly plan protective procedures, coverage of current stocks are required. For this purpose, images of the sea ground are currently viewed and annotated manually, which is nearly impossible for the whole considered surface area. This raises the question of whether an automatic localication and annotation of the coral is feasible. We will address this question in this paper. Therefore we use this year's

ImageCLEFcoral dataset [2] as the base for our research and also participated in their challenge, which is part of the ImageCLEF 2019 [8]. The task can be divided into two logical subtasks, localization and classification of objects. This is a wide spread research field of computer science, with many fields of application. The automotive industry seems to be an obvious field of research [3] and is commercially relevant. Today, driver assistance systems are ubiquitous. Recognition of road signs is a part of it. At first images of the road are taken via vehicle camera while driving. Second road signs are searched and classified in these recordings. The greatest results in such application scenarios are achieved by artificial neural networks. YOLO [11] showed one of the best results. The application scenario can be transferred very well. The localization and labeling of corals is similar, because the images are taken automatically and contain corals in unknown areas.

2 Data

The training set, which we define as dataset (A), contains 240 images with 6670 annotated substrates. Generally there is a differentiation between 13 substrate types. Which are: “Hard Coral – Branching, Hard Coral – Submassive, Hard Coral – Boulder, Hard Coral – Encrusting, Hard Coral – Table, Hard Coral – Foliose, Hard Coral – Mushroom, Soft Coral, Soft Coral – Gorgonian, Sponge, Sponge – Barrel, Fire Coral – Millepora and Algae - Macro or Leaves” [2]. For the submitted runs a test set containing 200 raw images is used, which correct labels and boxes were not available at the time of the publication.

Table 1: **Substrate types** with their relative frequency in the training set.

Class label	Relative frequency
c_algae_macro_or_leaves	0.0046
c_fire_coral_millepora	0.0015
c_hard_coral_boulder	0.1549
c_hard_coral_branching	0.1280
c_hard_coral_encrusting	0.0528
c_hard_coral_foliose	0.0082
c_hard_coral_mushroom	0.0258
c_hard_coral_submassive	0.0031
c_hard_coral_table	0.0009
c_soft_coral	0.5223
c_soft_coral_gorgonian	0.0024
c_sponge	0.0808
c_sponge_barrel	0.0145

The substrate types have an unbalanced distribution, as shown in table 1. Furthermore does the quality of the images vary, as well as the resolution. Some of the images contain a measurement white line, which is an obstacle while image processing.

2.1 Investigating the Dataset

When investigating the dataset, the problem of overlapping boxes appeared to us. Many of these bounding boxes fully contained or intersected with other boxes. To be more specific, only 2672 of 6670 bounding boxes do neither overlap or are contained in a bigger one. For this reason, we had started to investigate whether the substrates differ from each other at all, why we searched for meaningful features. These features were extracted from extracted bounding boxes. We applied a classical approach using SIFT [9]. Furthermore we calculated structure[5], texture[7] and color histograms in another approach. We used the calculated features to train a k-Nearest Neighbors classifier. The considered neighborhood k was set to $[3, 25] = \{k \in \mathbb{N} | 3 \leq k \leq 25\}$. Setting k to a higher value would lead to a strong dominatation of the neighborhood by frequent classes. With a train-/validation split of 80 : 20 we got our best results on a combination of texture, structure and color features. The following values show that the rare substrates are basically not found. In this way, we did not succeed in improving these values.

2.2 Augmentation

The amount of given data is remarkably low. Usually even a pre-trained neural network requires a larger data set, why we decided to use data augmentation. To generate new data, we used the following methods: noise and blur[1]. Other augmentation methods did not seem practical, since it would change bounding boxes. Therefore, we generated a second dataset (B) and could triple the data set size. Within the new dataset, which consists of substrate bounding boxes, we kept the class distribution, due to the probability of finding a frequently represented substrate type is significantly higher than that of a rare one. Also because balancing the dataset would require to cut frequent substrate types, which did not seem appropriate regarding the low number of annotated corals.

2.3 Sharpening

The images vary in quality and many of them are out of focus or blurry. To counter this and to create an improved dataset (C), we increased the contrast of entire images and highlighted the details. For this purpose, each pixel value was replaced by the weighted average of its 3 × 3 neighborhood. The following matrix shows the filter:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 12 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

3 Approaches

The challenge of the annotation and localization task is to find corals within images of sea ground, define bounding boxes for each coral and label them with their specific type of substrate.
We applied one state-of-the-art deep learning approach and additionally developed an own one. These two are presented in the following subsections, whereas the focus lies on explaining our own approach.

3.1 YOLO

In contrast to comparable neural networks, like "fast R-CNN" [4], which locate and classify objects multiple times for various regions of an image, the YOLO architecture passes the whole input image at once. That is achieved by dividing each image into square cells inside of which bounding boxes are predicted. In our work we scaled input images to a size of 608 x 608 pixels, because of the many corals contained in each image. This is the largest resolution we tested on our GPU and was the most promising. The classification process is basically a regression problem, which leads from image pixel values to bounding boxes with their class probabilities in one go. Part of the training is the optimization of predicted class probabilities, which defines the bounding boxes. In doing so, the calculation of each box considers features of the entire image. Therefore YOLO has the advantage of making less background errors as R-CNN, because more context information is taken into account. YOLO also outputs a confidence, which is calculated as the product of the precision of an object and its intersection over union (IoU). In a later step, this is multiplied by the conditional class probability of an object. Finally an output confidence is obtained, which describes how probable the particular class of the box is and how well the predicted bounding box fits this particular object.

Limitations However, there are some limitations. On the one hand each cell of the grid predicts only two boxes, which share the same class label. This is an algorithmic limitation on the number of objects with different labels, if the objects are close to each other. On the other hand the authors of YOLO mention that they treat errors in small bounding boxes the same way they treat large bounding box errors. Because of that, errors in small boxes have a larger impact on IoU, which leads to incorrect localization.

3.2 Own Developments

We developed a two-staged approach that first locates and then labels the substrates. Both of these steps make use of machine learning, to be more precise classification algorithms. This leaves room to improve the classification task, e.g. by evaluating different classification algorithms.

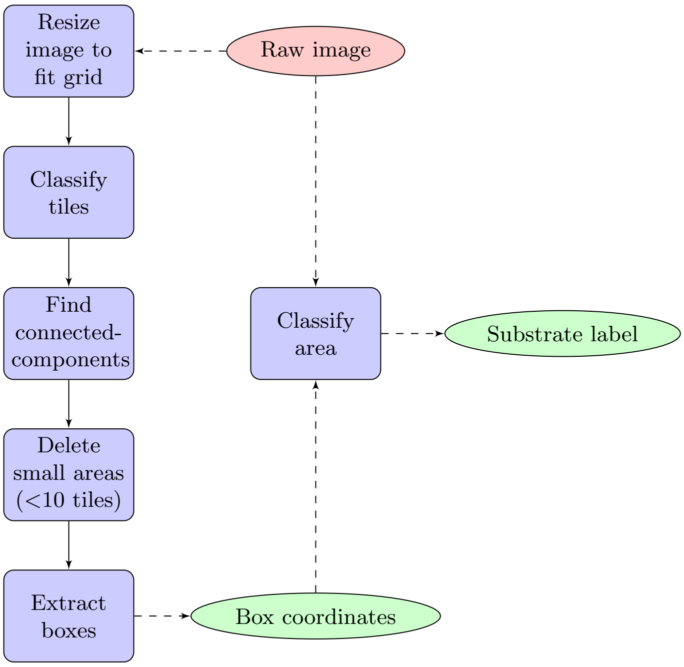


Fig. 1: **Flow diagram of our approach.** Red clouds are input data and green ones output data. The blue boxes are steps described in text.

One advantage of this two-staged approach is, that the two stages are independent from each other, which makes it possible to combine different algorithms and approaches together. The algorithm of our approach is shown in figure 1.

Locating Substrates The main idea behind locating substrates is based on the assumption, that the coral images have coral and non-coral areas. Such non-coral areas should look relatively similar for all coral types. This is quite different for images showing objects like cars or birds. Following our assumption, we segmented an image in coral and non-coral areas. First the image is divided in a grid, small enough to predict all boxes. To classify these areas we used a grid and then extract features from the tiles of this grid. We used a square of a fixed size for the tiles, which is based on the size of the smallest boxes in the training set, i.e. the integer average of the smallest width and height. Based on the training set, we recommend to use a tile size of 12×12, so that the smallest box can be located completely without background. To ensure that all tiles of an image have the same size, i.e. image size is the whole multiple of the tile size, the image is scaled to fit.

Next we extracted features for every tile of each of the training set images. For this purpose we used concatenated feature vectors consisting of features describing the color, texture and shape. For color, normalized histograms are used which describe the characteristics regarding the color [10]. The textural features of the tiles are modeled by Haralick texture features [5], which applies co-occurrence matrices on the gray scale level. Lastly the shape is represented by Hu moments [7]. Hu moments are invariant to translation, rotation and scale. All of these characteristics are useful for the domain of coral images.

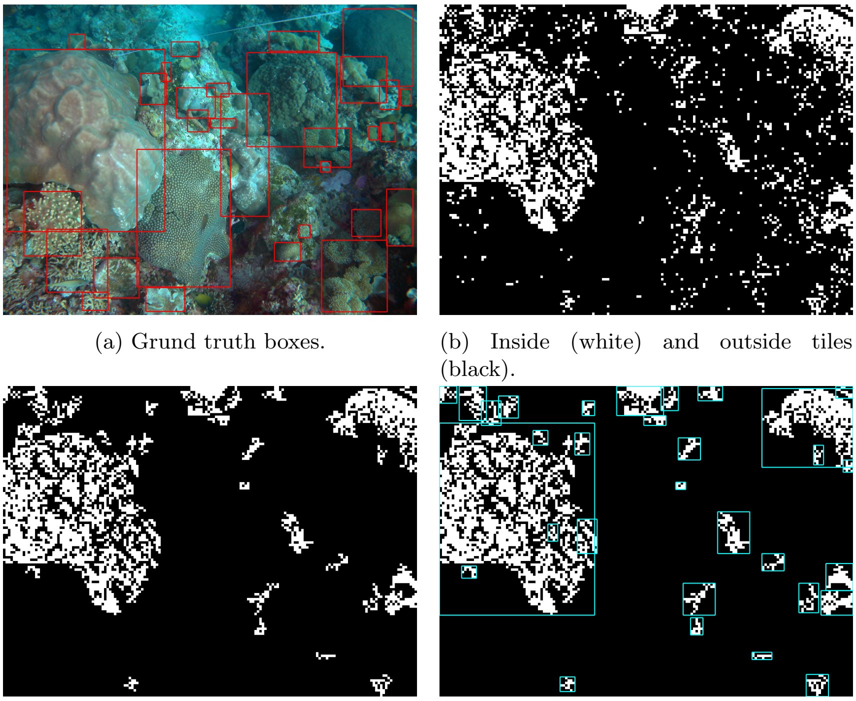
These features are used to train a binary classifier, which classifies whether a tile is a coral area or a non-coral area. An area of a training set image is considered a coral area, if more than 50% of its area intersects with a bounding box area of the ground truth. To classify areas of images that should be predicted, this image is also divided into tiles of the same previously defined size. Now the labels were obtained by feeding the features into the learned classifier. We decided to use K-Nearest-Neighbor with $k = 15$ to classify the tiles, because $k = 15$ performed best on our validation split.

After each tile of the grid was classified, we got an black and white image with 12×12 pixel large tiles. There are multiple strategies to extract boxes out of the resulting picture. We used a relatively naive approach with the application of connected-component labeling. Since we discovered a large amount of single, not connected tiles, we only kept components, that consisted of more than ten tiles. This counters a less beneficial performance of our classifier.

Each unique component is now bordered with a bounding box, that borders the outside tiles of the component. Figure 2 is showing the different stages of the location process (b - d), as well as the ground truth (a).

Labeling Found Boxes The found bounding boxes were classified on previously mentioned features 2.1 using a k-Nearest Neighbors Classifier. In addition to this already presented classification approaches, we studied whether the features can also be classified using a convolutional neural network. For the research, we subdivided the training data into a training and validation set in a ratio of 80:20 as previously. For comparability of the results we scaled the input data to the size of our grid. In consideration of the low amount of image data, we begun our work with a correspondingly small CNN, which we call baseline. The given CNN consists of one convolutional layer with maxpooling and rectified linear activation. We use dropout to prevent overfitting. The deactivation of neurons happens with a 20% probability. Subsequently, the data is handed to a flattening layer which serves as connection between convolutional and following dense layers. The result first enters a dense layer with RELU as the activation function and is then passed on to a density layer with softmax as activation function. This leads us to a confidence for each bounding box to belong to one of our 13 classes.

Considering that such a simple architecture may not be able to "remember" all relevant features of coral images, we extended our baseline architecture. Therefore we enlarged the existing architecture with two additional convolutional hid-



(a) Grund truth boxes.

(b) Inside (white) and outside tiles (black).

(c) Refined inside and outside areas.

(d) Bounding boxes of connected components.

Fig. 2: Visualized process of localization corals with our approach. The raw picture is taken from the ImageCLEFcoral dataset [2].

den layers.

Finally we looked for an extra deep architecture for comparison. All networks were trained with a batch size of 100 and with up to 1000 epochs. We decided to use VGG19 [12] and trained it on our data via transfer learning, since it has been proven to be gold standard in recent years.

4 Evaluation and Results

The following section discusses the submitted runs at ImageCLEF 2019. For a better understanding of the results of the approaches, we evaluated the localization and labeling separately. The results show that YOLO is considered state-of-the-art for a reason.

Besides presenting our results of the submissions, we also discuss the limitations and potentials of our approach as well.

In table 2 we present the results of our submissions. Our own approach is marked

with I, and YOLO based submissions with II. MAP_0.5 stands for the localised mean average precision for each submitted method with an IoU ≥ 0.5 of the ground truth and R_0.5 for the recall value, respectively MAP_0 represents the image annotation average without any localization. The results for I show, that CNNs and k-NN deliver comparable results. Sharpening has not led to better results, perhaps because it accentuates noise. YOLO combined with statistical probability distribution provides best results with an precision of 0.243 and a recall of 0.131.
Table 2 shows that we worked only on data set (A) and (C). We did not use data set (B) for our run submissions, since it did not lead to any kind of improvement.

Table 2: **Results of our submitted runs at ImageCLEF 2019.** Comparison of the results of the different approaches in our submissions. The methods used in I are our developed two-staged approach, whereas II are approaches using YOLO.

	Approach	Dataset	MAP_0.5	R_0.5	MAP_0
I	k-NN, k = 13	A	0.003	0.004	0.272
	Statistical labeling	A	0.002	0.003	0.203
	Baseline CNN	A	0.003	0.004	0.228
	Transfer Learning	A	0.003	0.004	0.291
	3-Layer CNN	A	0.003	0.004	0.205
II	YOLO + k-NN	A	0.229	0.131	0.500
	YOLO + Statistical	A	**0.243**	**0.131**	**0.488**
	YOLO + k-NN	C	0.210	0.122	0.455
	YOLO + Statistical	C	0.220	0.122	0.442

All approaches we used have some limitations and therefore leave space for improvement. Some of which we will describe in the following.

YOLO The weakness of YOLO is evident on rather smaller bounding boxes. Predictions on the validation data set showed that small coral substrates are either not found or subsequently labeled incorrectly. This results in the low recall value of 0.131. Corals that are found however, are mostly labeled as "c_soft_coral". Nevertheless, even on larger corals, YOLO shows rather moderate results. In a quarter of images it did not find boxes at all, that is why we used the found boxes from our other approach I to complete the results.

Our Approach Not only the performance (see Table 2) shows flaws in our approach, but also some obvious conclusions do. Since we got an accuracy of 0.534 on labeling boxes, which was evaluated on a 80 : 20 split of the training set, we assume that our approach fails to locate corals correctly. We also tested using SIFT features which had an accuracy of 0.4744.
One problem of the two-staged approach is the assumption, features of coral and

non-coral tiles are distinct enough. This leaves room for further evaluation and research, regarding the choice of features and labels. It might be beneficial to use more than two labels, i.e. more than just coral and non-coral. This could e.g. be water in the background, because we discovered that water in the background is often "false positive" classified, i.e. as coral area. An additional label would also need an additional annotation.

The question arises, whether 14 (13 substrate classes + background) labels could be used. This approach would only need one, instead of two classifiers.

Regarding the tile classification, the usage of CNNs to classify the tiles sounds promising because of the high number of tiles.

Another issue with our approach is the size of the tiles. Big tiles prevent small boxes from getting found and increase the chance of two corals in one tile. From a design perspective, boxes should be as small as possible to be as precise as possible. But if tiles are chosen relatively small, not only does the computational time extend, but features contain less information. This could lead e.g. to forms not getting recognized. We encountered the problem of an enormous computational time, because of that we increased the size to 24 × 24. Also we reduced the training set of tiles by 80%, which decreases the performance not significantly as seen in table 3. An approach of using a sliding window should also be considered in future work.

Table 3: Performance of 20% of the training set tiles compared to all tiles.

Amount of dataset	Precision		Recall		F_1 Score	
	Non-coral	Coral	Non-coral	Coral	Non-coral	Coral
1.0	0.6916	0.4950	0.7664	0.4013	0.7271	0.4433
0.2	0.6907	0.4900	0.7603	0.4034	0.7238	0.4425

Lastly using connected components as the method to extract boxes from the tile images, could be not sophisticated enough. Firstly with a perfect labeling of coral and non-coral tiles, it would not be able to recognize inlying boxes. And secondly it only considers two labels as features. The use of density-based clustering, working on more than just the predicted labels could lead to better results.

5 Conclusion

Overall, our approaches show moderate results. The idea to use neural networks proved to be promising. However, afterwards we can assert that YOLO was not the best choice. It completely fails to find smaller bounding boxes.

The concept of using feature engineering and searching for features or feature constellations, which are able to describe and represent different types of benthic substrate, still seems to be useful regarding the small amount of given data. But there is a lot of room for improvement.

Beside of that there are multiple images in the data set that show the same sea ground and contain for the most part the same corals. This kind of information can be used locally to improve the bounding boxes of corals, since their position can be tracked.

With regard to our approach 2b of labeling coral and non-coral areas, we can make the conclusion that the chosen features are not working properly. Probably we need a kind of back propagation to mark wrong labeled areas and process images multiple times. Additionally we could investigate the set of our features for a more performant subset using boosting. We would also stick to the deep learning approach and try another, maybe more time consuming but also more precise neural network, like an R-CNN.

Finally, the concept of combining deep learning and classic feature engineering is where we see the most potential.

Besides that, another point of potential improvement is the correction and balancing of the data set itself. Currently, seven of 13 coral type classes have a relative ratio of less than two percent, six out of them even less than one percent. The quality of the pictures is very variable too. Some of the images do not even seem to be completely annotated.

For future approaches, we would recommend publishing a larger and more balanced data set, in which each class has almost the same number of representatives.

To address the initial question whether an automatic localization and annotation of corals is feasible, we see good chances for future research.

References

1. Bloice, M.D., Stocker, C., Holzinger, A.: Augmentor: an image augmentation library for machine learning. arXiv preprint arXiv:1708.04680 (2017)
2. Chamberlain, J., Campello, A., Wright, J.P., Clift, L.G., Clark, A., García Seco de Herrera, A.: Overview of ImageCLEFcoral 2019 task. In: CLEF2019 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org (2019)
3. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark (2009)
4. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
5. Haralick, R.M., Shanmugam, K., et al.: Textural features for image classification. IEEE Transactions on systems, man, and cybernetics (6), 610–621 (1973)
6. Hoegh-Guldberg, O., Mumby, P.J., Hooten, A.J., Steneck, R.S., Greenfield, P., Gomez, E., Harvell, C.D., Sale, P.F., Edwards, A.J., Caldeira, K., et al.: Coral reefs under rapid climate change and ocean acidification. science **318**(5857), 1737–1742 (2007)
7. Hu, M.K.: Visual pattern recognition by moment invariants. IRE transactions on information theory **8**(2), 179–187 (1962)
8. Ionescu, B., Müller, H., Péteri, R., Dang-Nguyen, D.T., Piras, L., Riegler, M., Tran, M.T., Lux, M., Gurrin, C., Cid, Y.D., et al.: Imageclef 2019: Multimedia retrieval in lifelogging, medical, nature, and security applications. In: European Conference on Information Retrieval. pp. 301–308. Springer (2019)

9. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (Nov 2004)
10. Pass, G., Zabih, R., Miller, J.: Comparing images using color coherence vectors. In: ACM multimedia. vol. 96, pp. 65–73. Citeseer (1996)
11. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
12. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)

4.2 Enhanced Localization and Classification of Coral Reef Structures and Compositions

The results we have achieved during our investigation of the research question, **whether an automatic localization and annotation of corals in large scale images is feasible** outperformed those of the other top competitors of the ImageCLEFcoral challenge (Caridade and Marçal, 2019), (Jaisakthi et al., 2019) and (Steffens et al., 2019) but still showed a lot of space for improvement. Therefore, the proposed two-step architecture can not yet be considered a key concept for the automation of the annotation process or QA for monitoring of inventory control. For this reason, we want to benefit from the gained knowledge and deal with the same research question once again. Previously, in Chapter 4.1 we have discovered and discussed a series of hidden difficulties so far. To overcome these difficulties a closer look is necessary: on the one hand there are complications that can be summed up as special challenges in maritime imaging. These include certain quality issues of underwater images, such as color distribution, contrast, blueish cast and sharpness. At this point, we want to raise a more detailed question **whether an improvement of the image quality has a direct impact on the performance of OD**. Therefore, we apply and evaluate algorithms developed for the improvement of image quality of underwater images to coral data in the next publication. On the other hand, we saw issues concerning the given data set itself. Unbalanced class distribution in conjunction with a low number of images is a major challenge for most ML algorithms. Therefore we tackle the question **how a fair data split can be provided in terms of train and validation splits** on the same data. We consider in the next publication the split as fair if the amount of class represents in each subset is close to equal relative to the required percentage value. Although the common OD benchmark data sets such as COCO (T.-Y. Lin et al., 2014b) and Pascal VOC (Everingham et al., 2010c) do not benefit from the proposed method, because of their low complexity in regards to class distribution, our solution might be of a high significance for more complex data sets. In addition, we encountered the fact that small objects stay almost undetected. The reasons for this can be of different nature. Already Ferrari et al. (2007) drew attention to a correlation between loss of spatial information and weakness in localization performance, in general. In our opinion, this applies to both ML and DL. In terms of limitation of DL, Y. Zhang et al. (2019) found out that the CNN-based methods take mostly global features i.e. the fully connected for sizing and refinement of the bounding boxes, while local features being the more delicate input are more suitable of OD. Relating to One-Stage-Detectors such as YOLO, this insight can be expanded by the fact that the features from the bottom layers in high resolution are not used for detection because of a lack of semantic values (Xin Zhang et al., 2020). The missing ability to construct higher resolution layers and therefore benefit from the bottom as well as higher levels may be solved by Feature Pyramid Network (FPN) (T.-Y. Lin et al., 2017a), which we want to integrate in the further research. In the following paper, we rework both stages of our approach. Both DL and classical feature engineering are revised.

Enhanced Localization and Classification of Coral Reef Structures and Compositions

Kirill Bogomasov, Philipp Grawe, and Stefan Conrad

Heinrich Heine University, Universitätsstraße 1, 40225 Düsseldorf, Germany
http://dbs.cs.uni-duesseldorf.de
{bogomasov,grawe,stefan.conrad}@hhu.de

Abstract. The automatic annotation of coral images is important for researching the underwater ecosystem, which is the focus of the ImageCLEFcoral task. We participated by refining our approaches from the last years challenge for localization and classification of corals within images of sea floor. Underwater images bear multiple difficulties which we tackle with applying image enhancement algorithms. To locate and classify the corals we applied multiple deep learning approaches and also revisioned our two-staged algorithm. The results show that deep learning approaches are the most convincing. Still, the localization of corals is the most challenging part for us, but we managed to increase our models performance significantly.

Keywords: Image Segmentation · Image Classification · Object Localization

1 Introduction

Monitoring coral reefs and their health is an important component to understand effects of the climate change on maritime life [8]. Experts annotate underwater images, who not only have to deal with the complex morphology of the corals but also the large number of pictures. Computer vision based localization and classification of corals seems to be a reasonable solution. Unlike typical datasets for object detection tasks, underwater images hold more problems regarding the image quality and thus need very specific features and preprocessing.

In this paper we present the improvements of our approaches which are based on the last years ImageCLEFcoral [3][5] submission. Additionally we used another deep learning approach, namely RetinaNet [13], since this seemed to be the most promising. The classical machine learning is revisioned, but still not compatible with deep learning approaches regarding its performance. Lastly we implemented a popular suggestion to increase the image quality by preprocessing the images with algorithms made for underwater photography [7][1].

Overall we increased the performance of our approaches and can provide more insights, which we present in the following sections.

2 Related Work

The research of last year coral task can be divided into classical feature engineering and deep learning approaches. Caridade and Marçal [4] used random forest classification, based on a selected feature set, consisting of color and texture features to localize and classify the substrate types. Jaisakthi et al. [10] used a faster R-CNN to solve this task. Another solution proposal presented by Steffens at el. [22] is based on a DCNN architecture. Our approach differs from the mentioned research. Considering the different properties, distributions and sizes of the corals, we rely on a combination of both categories of image processing. The good results substantiate our approach and make it one of the most promising so far.

3 Data

For the purpose of the task [9][6], a training dataset with 440 images and 12077 annotated substrates, which are labeled with one of 13 substarte types, is provided. An additional dataset contains 400 raw images, that is used for testing the predictions while no further information about the images is given to the participants.

Table 1: **Substrate types** with their relative frequency in the training set.

Class label	Relative frequency
c_algae_macro_or_leaves	0.00761463
c_fire_coral_millepora	0.00157259
c_hard_coral_boulder	0.13590465
c_hard_coral_branching	0.09774872
c_hard_coral_encrusting	0.07829829
c_hard_coral_foliose	0.01464989
c_hard_coral_mushroom	0.01845721
c_hard_coral_submassive	0.01638802
c_hard_coral_table	0.00173812
c_soft_coral	0.46871379
c_soft_coral_gorgonian	0.0074491
c_sponge	0.13996027
c_sponge_barrel	0.01150472

The substrate representatives have a highly imbalanced distribution as shown in table 1. A random split of the data can lead to an even more disadvantageous class distribution, since it can exclude the representatives of the rare classes in one of the sets or increase the impact of frequent classes on the set. We give an example with a split into train and validation subsets with a ratio of 80 : 20 which distribution can be seen in table 2. To prevent the tendency of high imbalance, we propose the following procedure similar to [22].

May $P(D)$ be the relative distribution of classes of the complete image dataset D. Since both splits of the dataset should have the same distribution of classes, $P(D)$ is our target distribution. May A and B, with $A \cap B = \emptyset$, be the two sets after splitting D with the relative distributions $P(A)$ and $P(B)$. The key idea is to swap images between the two initial random splits A and B to make $P(A), P(B)$ and $P(D)$ as similar as possible. Let a, b with $a \in A$ and $b \in B$ be the two images to swap between A and B. We define $A^* = (A \setminus \{a\}) \cup \{b\}$ and respectively $B^* = (B \setminus \{b\}) \cup \{a\}$. If the similarity between $P(A^*)$ and $P(D)$ is smaller than the similarity between $P(A)$ and $P(D)$, we swap the items, so that $A = A^*$ and $B = B^*$. Since the similarity between $P(B^*)$ and $P(D)$ decreases when the similarity between $P(A^*)$ and $P(D)$ decreases, this approach works w.l.o.g. The loop over the images is running until there are no more swaps, i.e. no swap increases the similarity. To measure this similarity we use Jensen-Shannon divergence [11]. Since this procedure only converges to a local, but not global optimum, not every random split ends up in the same balanced split but consequently the optimal split is not found every time. Further research is needed to evaluate this approach, in regards of optimizations and metrics. The results of the balancing algorithm are shown in the table 3.

The Jensen-Shannon divergence between the train and validation set before swapping is 0.040313, whereas after swapping its divergence is 0.0061. This is an improvement by a factor of almost 8.

Table 2: **Substrate distribution** before balancing.

Class label	Relative frequency train	Relative frequency valid
c_algae_macro_or_leaves	0.00799747	0.005827505
c_fire_coral_millepora	0.00126276	0.00271950272
c_hard_coral_boulder	0.12722298	0.167443667
c_hard_coral_branching	0.0967063	0.101787102
c_hard_coral_encrusting	0.07965906	0.0730380730
c_hard_coral_foliose	0.01515311	0.012810513
c_hard_coral_mushroom	0.01894139	0.0167055167
c_hard_coral_submassive	0.01746817	0.01243201
c_hard_coral_table	0.0021046	0.000388500389
c_soft_coral	0.47606019	0.442113442
c_soft_coral_gorgonian	0.00683995	0.00971250971
c_sponge	0.13869304	0.144910645
c_sponge_barrel	0.01189098	0.101010101

4 Approaches

The "Coral reef image annotation and localisation task" can be divided into two tasks. Segmentation of various coral objects from images of sea ground and classification of those with their specific type of one of the 13 known types of

Table 3: **Substrate distribution** after balancing.

Class label	Relative frequency train	Relative frequency valid
c_algae_macro_or_leaves	0.00758972	0.00735809
c_fire_coral_millepora	0.00140952	0.00210231
c_hard_coral_boulder	0.13574759	0.13594954
c_hard_coral_branching	0.09779898	0.09775753
c_hard_coral_encrusting	0.07828255	0.07813595
c_hard_coral_foliose	0.01474574	0.0143658
c_hard_coral_mushroom	0.01843218	0.01857043
c_hard_coral_submassive	0.01637211	0.01646811
c_hard_coral_table	0.00173479	0.00175193
c_soft_coral	0.46882793	0.4688157
c_soft_coral_gorgonian	0.0074813	0.00735809
c_sponge	0.14008457	0.13980378
c_sponge_barrel	0.01149301	0.01156272

substrates.

Due to multiple difficulties that underwater images bear, strategies to enhance the image quality which should help to find better features are used. We applied two of the state-of-the-art deep learning approaches and additionally combined these with an improvement of our own development [3]. Those approaches are presented in the following subsections.

4.1 Image Enhancement

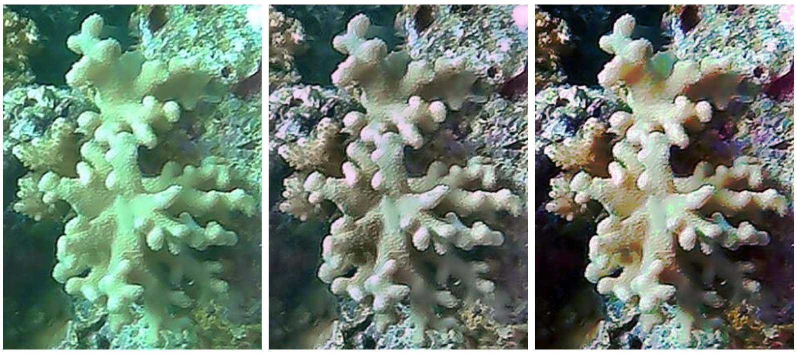
Underwater images inherit problems, like the attenuation of light or the suspension of particles reflecting the light. Those conditions distort colors and visibility, which affect the performance of machine learning algorithms. Therefore we applied and evaluated multiple enhancement algorithms, that are specialized on underwater images.

Ancuti et al. [1] use Fusion [24] and work without knowledge of a physical model of the lighting conditions. Two derivations of the image, improving the white balance and the contrast, are fused together using different weight measures to restore the image. The authors show that they retrieve more features using SIFT [15] through applying their image enhancement. Ghani and Isa [7] use Rayleigh-stretching, as well as stretching using the HSV color model, to first correct the contrast and then correct the color. Further on the processings are referred to as Fusion and RD. Since we do not have access to already corrected images of the instant dataset, we use three evaluation measures that predict how human would perceive the image, based on learned examples. Namely the measures are BRISQUE [16], NIQE [17] and PIQUE [23]. For all of them smaller values mean better image quality.

The evaluation [18] of the dataset of 2019 showed an improvement in image quality using the two enhancement algorithms, as seen in table 4 as well as in the subsections 4.4 and 5. Example images are shown in figure 1.

Table 4: **Image enhancement** algorithms evaluated on ImageCLEF Coral dataset 2019.

Algorithm	BRISQUE	NIQE	PIQUE
None	25.98	3.61	26.34
Fusion	22.66	3.43	30.99
RD	**20.92**	**2.94**	**25.06**



(a) Original (b) Fusion (c) RD

Fig. 1: Comparison of the image enhancements.

4.2 Yolo - Improvement

Neural networks are still state of the art in segmentation and classification tasks. Last year we used Yolov3 [20] as our main neural network approach. Especially areas with a particularly large denseness of smaller corals were challenging. The reason for this is on one hand Yolo's native ROI restraints which sets a natural limit on the regions considered within a certain area. On the other hand we were limited by the input data size with the largest resolution we could use with GPU of 608 × 608 pixels. Regarding the original resolution of images of 4032 × 3024, we preserve a scaling factor of at least of 5, i.e. each pixel represents an area of more than 25 pixels in the original image. Considering that the smallest corals consists of 12 × 12 pixels, the information loss is significant. We split the images into overlapping subimages of 608 × 608 pixels and trained the network on unscaled input. Consequently we got a large amount of images which also meant that the training time of our network was almost one month. Unfortunately, there was a mistake in the training data, therefore a revision of the results of the last years challenge was just recently announced. This left us with not enough time to retrain our working setup.

4.3 RetinaNET

For our second neural network approch we chose RetinaNet. The network showed impressive results on the COCO dataset and outperformed Yolov2 with a 17% higher $AP_{0.5}$ value. RetinaNet is well suited, since it is able to produce more predictions and is capable to work with less balanced data. At its core, the architecture consists of the following components: a feature pyramid network [12] (based on Resnet), a regressor for bounding box prediction and a classifier. Basically, it is a one-stage detector. The particular advantage of RetinaNet is the focal loss [14]. In case of end-to-end object detection, background predictions oftentimes dominate. The optimizer rates the prediction as correct and the loss of the positive background prediction forms the complete return loss. This mostly leads to an optimizer return value of zero for the background areas in case of cross entropy and thus reduce the loss. Focal loss weights the positive samples higher and ensures that the network performs better on unbalanced data.

Right suited anchor boxes are the key to quality of object detection for any architecture that works with a "regions of interest". If the anchors are not properly prepared, the network has in many cases no chance of finding particularly small, neither large objects. In our dataset, we experience a wide variety of sizes of corals. Starting from a box size of 18 × 9 to a size of 3966 × 2662 pixels, the standard deviation of the areas is 629 assuming a square size. That means that irregular or peripherally sized objects present a special challenge. To tackle this problem, we chose a solution that was originally used on medical data [25]. In our opinion, the potential for improvement can be easily transferred to coral context, since tumors and nodules as smaller objects are comparable to coral objects of small size.

4.4 Own Developments

Besides the deep learning we also increased the performance of our classic feature based approaches. We evaluated the use of principal component analysis (PCA) [19] to select the best features, which increased the performance slightly. Apart from the features the choice of the classifier is important. Last year we used k-NN, which is depended from the parameter k. To overcome the search for the right parameter we evaluated the use of naïve bayes [21] for locating and classifying substrates.

When classifying the coral areas and non-coral areas, the features along with the approach are the same as in [3], which is illustrated in figure 2. A problem that can be seen with k-NN is the low precision. This is due to labeling non-coral areas as coral areas, because most often water gets falsely classified. There are multiple ways to evaluate, as well as multiple things to evaluate. Beside the pure evaluation of the coral and non-coral tiles, we also evaluate the bounding boxes that enclose those coral tiles. In the end the evaluation of the found bounding boxes is more significant, but the pure evaluation of the coral and non-coral tiles helps with the assessment of the performance of finding bounding boxes in the generated black and white images (see figure 2b - 2d). Likewise this creates the opportunity to compare different image enhancement algorithms. All results are discussed in the following.

Table 5 holds the evaluation of the coral/non-coral grid that is compared with a grid representing the ground truth. The results show that the naïve bayes classifier has increased the accuracy, as well as the precision but greatly decreased the recall. PCA on the other hand did not have a big impact.

It could be shown that image enhancement increases the performance, even if just slightly. Table 6 showed that connected components works much better with the naïve bayes classifier that k-NN. The naïve bayes classifier has an insignificantly increased, whereas k-NN has significantly worse performance. We believe that this is caused by k-NN having to much false-positives which results in big boxes, that cover rather more than less area. Another indication for this assumption is the decrease of precision but increase of recall (compare figure 2b).

Table 5: **Evaluation** of the coral/non-coral classification, based on the tiles.

	Image enhancement								
	None			Fusion			RD		
Approach	Acc	Prec	Rec	Acc	Prec	Rec	Acc	Prec	Rec
k-NN	0.617	0.4496	0.5283	0.5823	0.4066	0.4813	0.6146	0.4355	0.4275
k-NN with PCA	0.6146	0.4471	**0.5343**	0.5858	0.4075	0.4637	0.6171	0.4371	0.4134
Bayes with PCA	0.6488	0.4545	0.1317	0.6570	0.3550	0.0031	**0.6575**	**0.4857**	0.0192

Table 6: **Evaluation** of the coral/non-coral classification, based on the bounding boxes.

	Image enhancement								
	None			Fusion			RD		
Approach	Acc	Prec	Rec	Acc	Prec	Rec	Acc	Prec	Rec
k-NN	0.4234	0.3920	0.8755	0.3919	0.4386	0.8543	0.4812	0.3973	0.8302
k-NN with PCA	0.4088	0.4088	**0.8927**	0.4097	0.4345	0.83106	0.4879	0.3888	0.8186
Bayes with PCA	0.6518	0.4833	0.2558	0.6571	0.3373	0.0025	**0.6606**	**0.5948**	0.0254

We also used the same two classifiers for classifying the bounding boxes. While using the same set of features, k-NN outperformed naïve bayes by far. Image enhancement shows its contribution again with tripling the accuracy of the naïve bayes classier, as seen in table 7. This substantiate that the enhancement proposed by Ghani and Isa [7] leads to increased performance with our dataset.

Table 7: **Evaluation** of substrate classification. The given values represent the accuracy.

	Image enhancement		
Approach	*None*	Fusion	RD
k-NN	0.4332	**0.4550**	0.4374
Bayes	0.1261	0.3328	0.3672

5 Evaluation of the Submitted runs

The evaluation was processed on test data that consists of images from four different geographical regions than the training set:

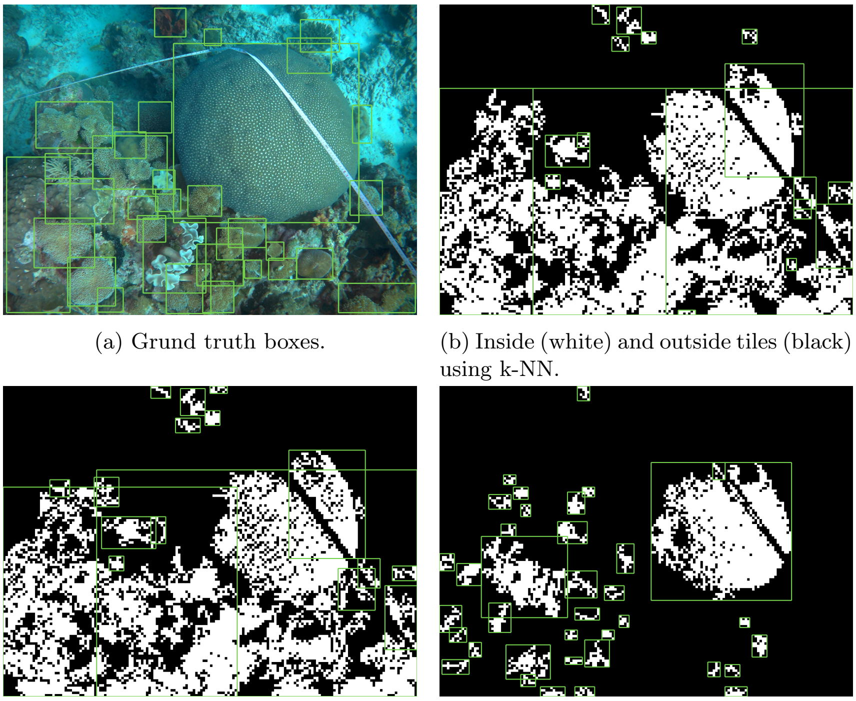
- same location
- similar location
- geographically distinct but ecologically connected
- geographically and ecologically distinct

In total the test dataset has 400 images, made by 100 images per subset. The results are examined in more detail below, while the interesting and informative values are discussed in the text. For the complete list of results, the reader is referred to the task overview working note [6]. Each submitted result was produced by one of our approaches, all of which were trained on the full training set.

The classification of the substrate types based on the classic features alone fails largely with an MAP_0 value of around 27.4. and an $\text{MAP}_{0.5}$ of 1. Although we improved the $\text{MAP}_{0.5}$ value compared to last year by 300 percent, the results are still not really useful due to the low absolute values. This applies to both experiments, for the classification by means of k-NN based on the chosen features boosted by PCA and to statistical label assignment as well.

Our neuronal Network based approaches show a significantly better performance.

Since we had a limitation in the number of submissions, we chose none linear composition of available options. Our pool of options consisted alongside to classical features of RetinaNet and Yolov3, which both were trained on the unpreprocessed data and also on enhanced images by RD and Fusion. In addition, we worked with a variation of threshold $\tau \in \{0.001, 0.1, 0.2, 0.5\}$, which limits the MAP.



(a) Grund truth boxes.

(b) Inside (white) and outside tiles (black) using k-NN.

(c) Inside (white) and outside tiles (black) using PCA and k-NN.

(d) Inside (white) and outside tiles (black) with PCA and naive Bayes.

Fig. 2: Visualized process of localization corals with our approach. The raw picture is taken from the ImageCLEFcoral dataset [6].

We achieve our best result with an ensemble of RetinaNet and Yolov3. Whereby the predictions of RetinaNet were extended by the predictions of Yolov3. Both Networks were trained on RD-preprocessed images, with a τ of 0.1. The combination of both systems results in a $\mathrm{MAP}_{0.5}$ of 39.2 % and MAP_0 of 80.6 %. It is noticeable that we predicted very few bounding boxes for a $\mathrm{MAP}_{0.5}$ both through Yolov3 and through RetinaNet, additionally the found predictions were far from being present in all of the images.

In case of reduction of the accepted overlap, we get significantly more bounding boxes which also leads to a higher chance of hitting the right coral within the test images on cost of our overall accuracy. However, if we increase the accuracy, the MAP value drops. The following are significant examples: RetinaNet ($\tau = 0.01$) combined with Yolov3 ($\tau = 0.01$) has an $\mathrm{MAP}_{0.5}$ of 0.303 and an MAP_0 of 0.727, but only an overall accuracy of 7 %. In contrast to it RetinaNet alone produces significantly fewer boxes with ($\tau = 0.2$), but achieves the best overall accuracy of 14.2 % with an $\text{MAP}_{0.5}$ of only 30.3 and a MAP_0 of 66.3 % while the accuracy is 10 % higher than in our best run.

A possible explanation is probably that RetinaNet has not finished training in 35 epochs. The large amount of predictions with a low overlap value could probably be boosted by non-maximum suppression. However, we were not aware of this problem until the results were published.

When provided with such a variety of possibilities, it can not be clearly determined which enhancement variant is the best choice. Certainly enhanced images improve the predictions, while we suspect that RD is superior to Fusion for coral images.

5.1 Transferability of the results within the test data subsets

One major question is the transferability of the results within the test dataset, considering that results, that form the average measure, vary widely. For our best run the $\text{MAP}_{0.5}$ for *same location* increases by 6 percent to 45.7 % and by 1 % to 81.5 % the MAP_0. For *similar location*, however, it drops to 28.3 % for $\text{MAP}_{0.5}$, but reaches a value of 86.4 % for MAP_0 and thus has the highest score among all presented submissions by all participants. Our best approach performs very well on *geographically similar* data. The $\text{MAP}_{0.5}$ is 42.6 % and the MAP_0 81.2 %. Overall, this part of the test dataset seems to be less complex, which is shown by the fact that the performance of almost all of our approaches have an increased performance on it.

It is also worth mentioning that we perform significantly worse on *geographically distinct* data with just an $\text{MAP}_{0.5}$ 0.125 and an MAP_0 0.362. This tendency is also evident in the other approaches we used. The data seems to differ significantly, we either generalize less or the classes that are harder to recognize are more present. Some additional, yet unknown substrate types may be more dominant in geographically and ecologically distinct rocky reef and lead to distorted results. A further investigation of the dataset is required.

6 Conclusion

Overall, our approaches show significantly better results than last year. A comparison of our best approaches between the two years shows that we have improved the $\text{MAP}_{0.5}$ by 13.4 %. The classification of the tiles into in- and outside boxes could also be improved by a Bayesian classifier, but is still far from being accurate.

Image enhancement techniques on the last years data were confirmed by the evaluation of the current test dataset, which leads us to the conclusion that the correction of blurry images in terms of contrast and sharpness is necessary. The RD algorithm makes the greatest contribution to improving the quality of the images. Less noteworthy results are made with the classical feature engineering approach. A deeper examination of the features and their information value is needed.

It can be assumed that the fair partitioning of images according to our balancing strategy, also has significantly contributed to the improved results. Deep learning strategies generalize quiet well and are superior when using for this task. Especially the performance of RetinaNet, since it is not only better on the coco dataset than the state of the art. So far, the complexity of the images can hardly be handled by a single approach. We still see the most potential in an ensemble of several architectures. The combination of advantages of different approaches is the key to a stable solution. Since the amount of data has grown, while remaining relatively small, we still cannot exclude the potential of classic machine learning. Usually neural networks show a much better performance with an increased amount of data. For this reason, the images should be split into overlapping images and thus increase the number of training samples.

For future approaches, we recommend the usage of more specific features that are suitable for corals, such as used in [2]. However, we would rather rely on an ensemble of neural networks of the gold standard, which we would reduce in depth to shorten the training time and increase in terms of input resolution to decrease the information loss.

References

1. Ancuti, C., Ancuti, C.O., Haber, T., Bekaert, P.: Enhancing underwater images and videos by fusion. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 81–88. IEEE (2012)
2. Ani Brown Mary, N., Dharma, D.: Coral reef image classification employing improved ldp for feature extraction. J. Vis. Comun. Image Represent. **49**(C), 225–242 (Nov 2017). https://doi.org/10.1016/j.jvcir.2017.09.008, https://doi.org/10.1016/j.jvcir.2017.09.008
3. Bogomasov, K., Grawe, P., Conrad, S.: A two-staged approach for localization and classification of coral reef structures and compositions. In: Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019 (2019), http://ceur-ws.org/Vol-2380/paper_106.pdf
4. Caridade, C.M.R., Marçal, A.R.S.: Automatic classification of coral images using color and textures. In: CLEF (2019)
5. Chamberlain, J., Campello, A., Wright, J.P., Clift, L.G., Clark, A., García Seco de Herrera, A.: Overview of ImageCLEFcoral 2019 task. In: CLEF2019 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org (2019)
6. Chamberlain, J., Campello, A., Wright, J.P., Clift, L.G., Clark, A., García Seco de Herrera, A.: Overview of the ImageCLEFcoral 2020 task: Automated coral reef image annotation. In: CLEF2020 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org (2020)
7. Ghani, A.S.A., Isa, N.A.M.: Underwater image quality enhancement through composition of dual-intensity images and rayleigh-stretching. SpringerPlus **3**(1), 757 (2014)
8. Hoegh-Guldberg, O., Mumby, P.J., Hooten, A.J., Steneck, R.S., Greenfield, P., Gomez, E., Harvell, C.D., Sale, P.F., Edwards, A.J., Caldeira, K., et al.: Coral reefs under rapid climate change and ocean acidification. science **318**(5857), 1737–1742 (2007)

9. Ionescu, B., Müller, H., Péteri, R., Abacha, A.B., Datla, V., Hasan, S.A., Demner-Fushman, D., Kozlovski, S., Liauchuk, V., Cid, Y.D., Kovalev, V., Pelka, O., Friedrich, C.M., de Herrera, A.G.S., Ninh, V.T., Le, T.K., Zhou, L., Piras, L., Riegler, M., l Halvorsen, P., Tran, M.T., Lux, M., Gurrin, C., Dang-Nguyen, D.T., Chamberlain, J., Clark, A., Campello, A., Fichou, D., Berari, R., Brie, P., Dogariu, M., Ştefan, L.D., Constantin, M.G.: Overview of the imageclef 2020: Multimedia retrieval in medical, lifelogging, nature, and internet applications. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the 11th International Conference of the CLEF Association (CLEF 2020), vol. 12260. LNCS Lecture Notes in Computer Science, Springer, Thessaloniki, Greece (September 22-25 2020)
10. Jaisakthi, S.M., Mirunalini, P., Aravindan, C.: Coral reef annotation and localization using faster r-cnn. In: CLEF (2019)
11. Lin, J.: Divergence measures based on the shannon entropy. IEEE Transactions on Information theory **37**(1), 145–151 (1991)
12. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. CoRR **abs/1612.03144** (2016), http://arxiv.org/abs/1612.03144
13. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. pp. 2980–2988 (2017)
14. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. 2017 IEEE International Conference on Computer Vision (ICCV) (Oct 2017). https://doi.org/10.1109/iccv.2017.324, http://dx.doi.org/10.1109/ICCV.2017.324
15. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
16. Mittal, A., Moorthy, A.K., Bovik, A.C.: No-reference image quality assessment in the spatial domain. IEEE Transactions on image processing **21**(12), 4695–4708 (2012)
17. Mittal, A., Soundararajan, R., Bovik, A.C.: Making a "completely blind" image quality analyzer. IEEE Signal processing letters **20**(3), 209–212 (2012)
18. Nguyen, H.C.: Comparison of methods for underwater image improvement (2020)
19. Pearson, K.: Liii. on lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**(11), 559–572 (1901)
20. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
21. Rish, I., et al.: An empirical study of the naive bayes classifier. In: IJCAI 2001 workshop on empirical methods in artificial intelligence. vol. 3, pp. 41–46 (2001)
22. Steffens, A., de A. Campello, A.C., Ravenscroft, J., Clark, A., Hagras, H.: Deep segmentation: using deep convolutional networks for coral reef pixel-wise parsing. In: CLEF (2019)
23. Venkatanath, N., Praneeth, D., Bh, M.C., Channappayya, S.S., Medasani, S.S.: Blind image quality evaluation using perception based features. In: 2015 Twenty First National Conference on Communications (NCC). pp. 1–6. IEEE (2015)
24. Xydeas, C., , Petrovic, V.: Objective image fusion performance measure. Electronics letters **36**(4), 308–309 (2000)
25. Zlocha, M., Dou, Q., Glocker, B.: Improving retinanet for ct lesion detection with dense masks from weak recist labels. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 402–410. Springer (2019)

5 Quality Assurance in Retail Applications

"The first rule of any technology used in a business is that automation applied to an efficient operation will magnify the efficiency. The second is that automation applied to an inefficient operation will magnify the inefficiency."
— (Furrer, 2019)

5.1 Efficient Fruit and Vegetable Classification and Counting for Retail Applications Using Deep Learning

In order to be able to answer the research question listed at the beginning of this thesis in Chapter 1.2 also for smaller images with a rather low resolution and correspondingly a lower information density, further data needs to be analyzed. This entails additional requirements, including efficiency and processing time issues.

Inspired by the success of the ImageCLEF coral challenge, we set the goal to make a transition of our findings and algorithms from object detection on natural images to business-related applications. Convinced of our intent and in accordance with our own development, REWE IT Solutions have agreed to cooperate with us. According to the agreement, the core research question, in this case, is **how to ensure the quality of the sales process by counting and classifying barcode-free goods, such as fruits and vegetables in local markets using CV**. The motivation behind this question is to make the cashier's work easier, make the process smoother while reducing errors, and additionally monitor the stocks for each market. Therefore, images of these products created by a webcam at the checkout in supermarkets and labeled by the corresponding class but without localization information were provided. The practical insights have shown that the process of image acquisition entails a wide variety of obstacles in images. For instance, customer hands, carrier bags or wrong camera placement, to a great extent, preclude a conventional feature-based ML approach. Likewise, a large variation in lighting conditions makes the data set even more irregular.

The first idea to solve this problem was to use object localization techniques to find objects and count these subsequently. Our initial approach to apply the methods that we primarily developed for complex images with high resolution and thus benefit from preliminary work did not succeed. Since only budget-friendly hardware was intended to be installed in markets, it was necessary to switch from powerful GPUs to CPUs. The tested inference time on CPU of our architecture presented in chapter 4 on the new data set was well over a second and did not meet the expectations of the project partner because the process took too long and was therefore not practicable. Starting from this finding, we realized that the baseline had to be built on an architecture that has a mobile application area as a background. Usually, the efficiency of such architectures is achieved through less depth, less computationally intensive 1-depth convolutions, and also less trainable parameters. Obviously, fewer trainable parameters might mean less accuracy. However, the decrease of performance usually takes place on rather tiny objects, which are unusual in this specific case.

In the following publication, we investigate the performance of different mobile networks for a counting and classification task on real-world data. In particular, the evaluation results are compared to an own innovative two-step architecture which includes the weight information for each image.

Efficient Fruit and Vegetable Classification and Counting for Retail Applications Using Deep Learning

Kirill Bogomasov
bogomasov@hhu.de
Department of Computer Science, Heinrich Heine
University Duesseldorf
Düsseldorf, Germany

Stefan Conrad
stefan.conrad@hhu.de
Department of Computer Science, Heinrich Heine
University Duesseldorf
Düsseldorf, Germany

ABSTRACT

The process of manual classification and counting of fruits and vegetables, from the moment the customer places items on the conveyor belt to their weighing by the cashier on the checkout scale is time consuming and may be burdensome for cashiers, who need to look up or remember the identification code for each product. Not any more: We built a real-life application, which is capable of doing both tasks simultaneously. The presented research is focused on a case that is attractive for its practical applications, in which data is expanded by product weight information. We approach the problem as that of estimating the object count as a classification task and evade the more resource consuming object detection. We introduce a new hybrid architecture which is an ensemble of EfficientNet [31] for image classification and a Decision Tree [3] for object counting based on weight and previous classification result. The trained architecture provides accurate object count and requires fewer resources and less time than current object detection architectures. The proposed architecture accomplishes a counting accuracy of around 80% and an inference time of 0.2 sec. per image. It is a good candidate for handling huge amount of visual information involving fast processing on a CPU.

CCS CONCEPTS

• **Computing methodologies → Neural networks**; **Classification and regression trees**; • **Applied computing → Service-oriented architectures**.

KEYWORDS

deep learning, object counting, quality assurance, hybrid architecture

ACM Reference Format:
Kirill Bogomasov and Stefan Conrad. 2021. Efficient Fruit and Vegetable Classification and Counting for Retail Applications Using Deep Learning. In *ICAAI 2021: The 5th International Conference on Advances in Artificial Intelligence (ICAAI)(ICAAI 2021), November 20–22, 2021, Virtual Event, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3505711.3505712

ICAAI 2021, November 20–22, 2021, Virtual Event, United Kingdom

ACM ISBN ISBN 978-1-4503-9069-9/21/08…$15.00
https://doi.org/10.1145/3505711.3505712

1 INTRODUCTION

In Germany, each citizen consumes on average 200 g vegetables and 250 g fruits every single day [10]. Selling such amounts is challenging for retail and all associated processes. Checkout scales in supermarkets are an essential part of the sales process. Incorrect or incomplete billing of sold products have serious effects on corporate balance sheets and on the inventories as well. Additionally, mistakes on a regular basis can lead to customer dissatisfaction [30] and mistrust of the retailer. Both potentially may result in a loss of customers.

Usage of such systems requires cashiers to scan the bar code of each product, which is then retrieved from the connected database. Products from the group of fruits and vegetables usually do not have a bar code due to time restrictions and effort in placing a sticker on each piece. Goods, that do not have a bar code have to be included manually by a trained cashier, which can cause a bottleneck. Speed and accuracy of registration of product identification codes depends on the level of experience of staff and also on the duration that the product was being part of the assortment. Especially seasonal goods are frequently exchanged and therefore their codes often have to be looked up manually. This results in long queues for customers and also suggests another source of error, leading in turn to customer dissatisfaction [35]. There is a large need for quality assurance systems that support cashiers in classification and counting of fresh fruits and vegetables. Several publications covered this topic in the last few years, mostly in terms of object detection [22, 24, 25]. None of them deal with the issue from a more pragmatic cost-benefit perspective.

The limited transferability of the concepts is a big challenge for companies, and the body of existing research is so small, it is barely worth mentioning. Cost-benefit analysis determines the course of action. Ventures often do not have enough resources to set up a computing cluster to monitor their production, processing or sales. Especially challenging are opulent hardware requirements for Deep Learning-based solutions such as a large storage GPU. These are high-priced, have to be cooled at great expense and, with a few exceptions, are mostly used in workstations, which is why the question of an easy-to-implement solution without special hardware (i.e. GPU) and architectural design requirements arises.

In this paper we present a hybrid model, which solves classification and counting on the basis of information retrieved from a scale supplemented with a simple webcam. To be specific, product image and corresponding weight are retrieved in ongoing operation. The model is able to steadily recognize products passing the scale without a bar code in a fraction of a second. This allows object classification on a single core CPU or even in real time on

Coral Ai Edge TPU[1]. This also renders usage of high-priced and maintenance-intensive GPU unnecessary. Additionally, our model is able to count classified objects simultaneously. Both model properties allow a novel quality assurance system, which speeds up the checkout processing, minimizes potential errors and enables monitoring of the remaining stocks in markets. The latter could subsequently be used to a timed parenthetical restocking.

To summarize, our key contributions in this paper are:

- Presenting an efficient and innovative solution for classification and counting of fruits and vegetables for retail.
- Showing that the introduced architecture does not require extensive manual annotation of examples for training, unlike current object detection architectures.
- Comparing our solution with leading technologies in the field of mobile object detection and providing that the proposed solution is faster and less expensive while still being accurate.

2 RELATED WORK

The goal of object counting is to count the number of object instances in a single image or within an image sequence, which is just part of the problem. Every counted object needs to be labeled correctly, otherwise an error will lead to a serious miscalculation. Therefore, we see counting as an interdisciplinary task. Consequently, there are several task-related research fields.

Object Detection The counting problem can be considered as the estimation of the number of objects in an image or a video frame. Therefore, object detection is ideal for this goal. Sai et al. [26] rely on Faster R-CNN for a two-class problem and count the objects found in the images. Zhang et al. [38] showed that an additional subitizing technique using end-to-end CNN models may increase the performance of object localization. Proposed subitizing method was based on global image features. Chattopadhyay et al. were another research group who used subitizing [5]. Hsieh et al. [11] used case adapted spatial layout information to improve the counting process of a regularized regional proposal network. Zhang et al. [39] proposed a neural counting component for R-CNN architecture, which was primarily used for Visual Question Answering (VQA). In general, counting is commonly used in visual question answering [34]. The attention-based model [1] was applied for transcribing house number sequences from Google Street View images. Some rather large deep learning-based architectures specifically designed for counting were recently presented: Few-shot detection [20] and Count-ception [6].

Non-deep learning methods for object quantification are less common and outdated. Probably the most interesting idea was presented by Lempitsky [14], who used image density whose integral over any image region gave the count of objects within a selected region.

However, none of these approaches took efficiency in any way into account. Real time calculability was also not considered. All of the proposed architectures were computationally expensive and thus not appliable for the given task.

Hardware Some research is focused on acceleration of Deep Convolutional Neural Networks. The largest resource-consuming and computation-intensive modules are Convolutional Layers, which constitute over 90% of total operations [18]. The problem is sometimes reduced to maximization of parallelism for computation and reducing required memory bandwidth (e.g. using FPGA) [15]. In some cases SVD decomposition can be used to lower the complexity of algorithms [15]. To overcome the problem of limited memory, bandwidth weights are stored in the on-chip memory to reduce data accesses, which sets another constraint, since sufficient memory is still required. Other research is focused on techniques to lower the power consumption on CNN methods. Yu et al. [37] give an extended overview on this. However, such approaches are not universal and often need to be adapted to particular hardware. For this, deep insights are required. Since the only limitation we want to make is the ability to run on a CPU, we no longer consider hardware specifics.

Fruits and Vegetables Another respective research field is the fruit and vegetable classification. A detailed overview on this topic can be found in [8]. Practice-inspired research is done on fruit detection. Bargotti et al. [2] presented Deep Fruit detection of 3 categories for robotic harvesting. Another application called "Deep Count" [22], which is based on a modified Inception-ResNet and is used for robotic agriculture. The application focuses only on tomato images from Google Images. A more specific application is presented in "DeepFruits" [25], which is a Faster Region-based CNN working on two types of input images RGB and Near-Infrared (NIR) (7 classes).

Scale A lot of research was done on checkout-support-systems. These systems are used in the same context as checkout scales. The key aspect was set to reduce costs and gain more flexibility. Several self-checkout systems are using Deep Learning for product detection. Katarzyna et al. [12] presented an approach based on Yolov3 [23]. They also pointed out a number of difficulties in the sales process of fresh products. The authors studied classification of apples and were successful with Yolov3 and a simple CNN architecture. Zhang et al. [4] used Feature Pyramid Network for the same task. Both publications dealt with packaged goods, which could be classified on packaging. Rojas et al. [24] was probably the first study to consider fruit classification inside or without plastic bags. Due to complexity, the dataset was limited to just three classes of fruits with only 1067 image in total. According to the chosen classes of apples, oranges and bananas the task seems more easy to handle. The best performance was achieved by slightly adapted MobileNetV2. By this reason, we use MobileNetV2 as part of our study.

Weight Little contribution was done on inclusion of product weight information. Wu et al. [36] used constant weight information for single packed products to combat fraud on self-checkout systems based on "ticket switching". The authors use visual features like SIFT, since only single packages are regarded. Hameed et al. used the weight information to change the coarse classification of 15 classes down to three, which were further classified on an AdaBoost-based CNN. The used weight was the average weight of each class [9]. Somehow, weight is discussed in several publications, but rarely successfully applied.

[1] https://coral.ai/

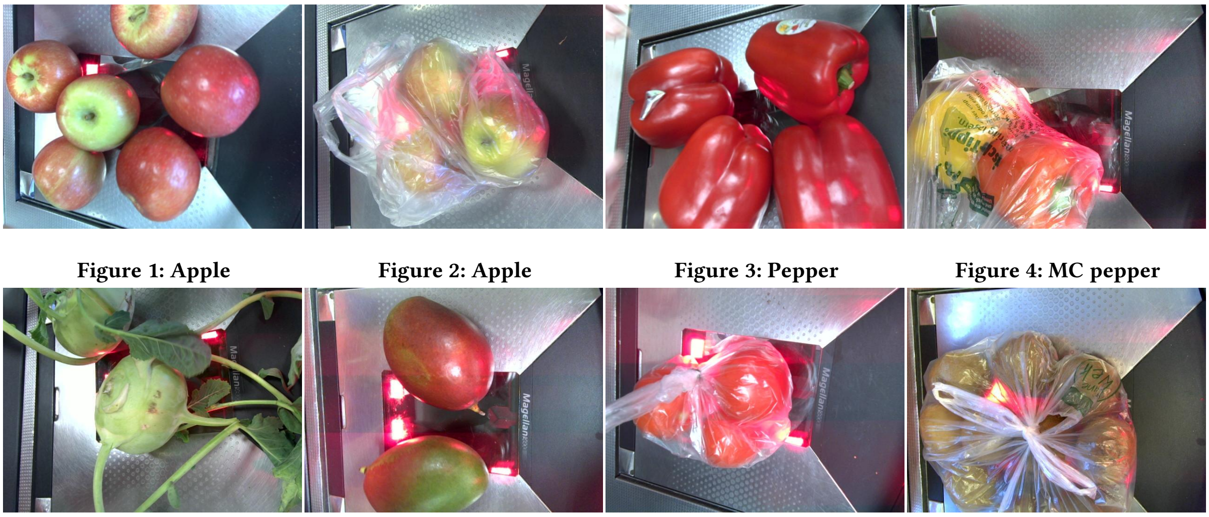


Figure 1: Apple **Figure 2: Apple** **Figure 3: Pepper** **Figure 4: MC pepper**

Figure 5: Kohlrabi **Figure 6: Mango** **Figure 7: Tomato** **Figure 8: Kiwi**

Figure 9: Example self-collected images from each class.

3 DATASET

Our dataset has been collected in ongoing selling processes in stores of a large German supermarket chain, so it is the most in step with actual practice. The dataset has been composed of 2380 images in total. Images with visually ambiguous quantity of objects were removed previously. All examples were divided into seven classes, with $c \in$ {tomato, mango, pepper, mixed color pepper, apple, kohlrabi}. There were 340 single class images i along with weight information w_i, with $w \in \mathbb{N}$ for each class. The classes were carefully selected as representatives of the full range of products, which covered convenient difficulties such as product similarity in color, shape and weight. The pictures were taken in different markets and naturally differ in brightness and camera orientation (viewing angle) due to different environmental conditions and settings. We manually annotated each product in each image by a labeled bounding box. Many pictures contained obstacles such as plastic bags and shopping nets and were not removed with the aim of retaining the complexity. However, the resolution of 640 × 480 pixels is the same for all images. For each experiment, we use the same random train and test subsets split by a ratio of 80:20. Namely, 268 images are used for training and 72 for testing. Example images from each class can be found in Fig. 9. The object quantity distribution is visualized in Fig. 12.

4 APPROACH

Our task is to accurately count the number of instances of correctly predicted classes in an image of fruits and vegetables. The most promising object detection architecture in terms of processing time and accuracy has been shown to be MobileNetV2 [24, 27]. In the meantime, a higher-performance successor model has been presented, namely EfficientDet [32]. We include both in our research for comparison. Additionally, one of the components of EfficientDet becomes part of our model, which we present in this section.

4.1 Baseline Architectures

The specified architectures described below naturally differ in input data resolution due to their technological properties. If not predetermined in original publications, we try to keep the size of the input images as close to the original one as possible. In terms of batch size, it varies due to limitations of GPU storage.

4.1.1 MobileNetV2. MobileNetV2 is one of the most promising mobile architectures and the gold standard for object detection. It is based on inverted residual structure. The intermediate expansion layer uses lightweight depth-wise convolutions to filter features and thus reduces non-linearity. These properties and a few more lead to a significant increase in speed [27]. We use an architecture that the authors call SSDLite. The settings are as follows: batch size = 2, Adam optimizer [13] with an initial learning rate = 0.001. Further settings are upon the recommendation of the authors.

4.1.2 EfficientDet. EfficientDet showed the ability to produce similar accuracy with at least 19 × fewer FLOPs than NAS-FPN[7], RetinaNet[16] and YOLOV3[23] [32]. The architecture uses EfficientNet as a backbone and can easily reuse ImageNet-pretrained checkpoints, which we used in all approaches, EfficientDet and our own Hybrid Architecture to hold the results comparable.

The reason for detection head efficiency is that a weighted bi-directional feature pyramid network allows a fast multi-scale feature fusion. From a practical point of view, this is quite interesting because the network fuses features at different resolution scales.

Each model is trained using Adam optimizer with a cosine restart learning rate with an initial learning rate of 0.001. Synchronized batch normalization is added after every convolution with batch norm decay of 0.99 and ϵ = 1e−3. We also use swish activation with an exponential moving average with decay of 0.9998, as proposed in [32].

We used a focal loss as common with $\alpha = 0.25$ and $\gamma = 1.5$, while the aspect ratio was 1/2, 1, 2 for both EfficientNet and EfficientDet and MobileNetV2 as well.

EfficientDet D0 is trained with a batch size of 32, EfficientDet D1 with size of 8 and EfficientDet D2 with size of 2.

4.2 Hybrid EfficientNet Architecture

Object localization is computationally an expensive task. While localization always requires calculation of numerous precise bounding boxes to fit the object, counting does not. Basically, the positional information is not relevant at all. Because of this we built a two step approach to handle the task and relinquish localization. This reduced the calculation complexity extensively. The developed hybrid architecture is shown in Fig. 10. The first step consists of a classification head (EfficientNet), which provides a confidence score s with $s \in [0, 1]$ provided by Softmax function for each class label c. Afterwards, Argmax is applied to get the index location of the maximum value inside the output tensor, which in fact is the most probable class c_i for the input image $i \in I$. The second step is classification too. The chosen classifier is a Decision Tree (DT) which takes a combination of the previously predicted object class c_i and additional weight information w_i as input and returns the object count q. Taking into account the output of EfficientNet, we are able to tell the class and the amount of items.

Due to limitations imposed by the hardware, we only consider models with the sizes B0 - B2 in case of EfficientNet and D0 - D2 in case of EfficientDet accordingly. The input size of D0, D1 and D2 is 224 × 224, 240 × 240 and 260 × 260 for B0, B1 and B2 accordingly.

4.2.1 *EfficientNet.*

EfficientNets indicate better efficiency than the previously widely used backbones [32]. Especially attractive is its advantage of different width/depth scaling coefficients, which allow a custom case dependent thread-off between accuracy and resource consumption. Therefore, we investigate the accuracy for three configurations: B0, B1 and B2. The settings are equivalent to the settings of EfficientDet as far as present. The batch size is increased to 32, due to a smaller input size of 224 × 224, 240 × 240 and 260 × 260 for B0, B1 and B2.

4.2.2 *DecisionTree.*

Decision tree algorithms are widely used in machine learning [21]. The input data is filtered down through the leafs to get the right output to the input pattern. Many algorithms have been proposed, we use one of its newer versions [33], which has been shown later on to have better performance than previous versions [17]. One of the most important advantages is the simplicity of the algorithm and the fact that in the chosen version, it has just a few parameters. We set the maximal depth to 5 to keep decisions traceable and prevent the classifier from over-fitting. The decision criterion for information gain is entropy [28], like in several other publications [29].

5 EVALUATION

All architectures are trained for 500 epochs on Tesla V100 GPU. Since transfer learning has proven to be useful in the context of comparable data, we use model weights trained on ImageNet data for each mentioned architecture. To increase the quantity of training samples augmentation techniques are applied on trainings data. We use rotation (f = 0.15), which results in an output rotation by a random amount in the range of $[-15\% * \pi, +15\% * \pi]$. We also use translation (height_f = 0.1, width_f = 0.1) which leads to an output height and width shift by a random amount in the range $[-10\%, +10\%]$, a contrast change (f = 0.1) randomly picked and adjusted to each x for each channel to $(x - mean) * f + mean$. Finally, a random flip is randomly applied on the input data.

In case of inference, each model is run on a single-thread Xeon CPU. For the purpose of counting, each image is processed separately. We consider only bounding boxes with a confidence score higher than 0.3 to prevent vice versa classification of similar classes like mixed color pepper ("MCPepper") and red pepper ("Pepper"). We report model accuracy, which is extended by RMSE, MAE since the count error is also relevant for stock monitoring. Additionally, we report standard deviation and variance for prediction results.

5.1 Performance Metrics

Mean Absolute Error (*MAE*) and Root Mean Squared Error (*RMSE*) are widely used for counting performance measurement [19, 40]. Since both metrics are used to measure the error for same class instances, they first need to be extended to be able to benchmark the counting error in case of misclassification. Let i be an image with $i \in I$, $c \in C$ the ground truth class, c' the predicted class. Furthermore, let q be the ground truth quantity of objects and q' the predicted quantity then ground truth g and prediction p are defined as:

$$g, p : i \in I \longrightarrow C \times \mathbb{N}$$

which means:

$$g(i) = (c, q)$$
$$p(i) = (c', q')$$

and

$$pr_1(c, q) = c$$
$$pr_2(c, q) = q$$

Accordingly, prediction $p\hat{(i)}$ is defined as:

$$\hat{p_i} := \begin{cases} pr_2(p(i)), & \text{if } pr_1(p(i)) = pr_1(g(i)) \\ 0, & \text{otherwise} \end{cases}$$

Finally *MAE* and *RMSE* are defined as following:

$$MAE = \frac{1}{|I|} \sum_{i \in I} |pr_2(g(i)) - \hat{p}(i)|$$

$$RMSE = \frac{1}{|I|} \sqrt{\sum_{i \in I} (pr_2(g(i)) - \hat{p}(i))^2}$$

where $|I|$ is the number of test set images, $\hat{g}$ and $\hat{p}$ are ground truth and predicted count for correctly classified images, otherwise $\hat{p} = 0$ because wrongly-classified objects should not be counted, as they can not be subtracted from the stocks.

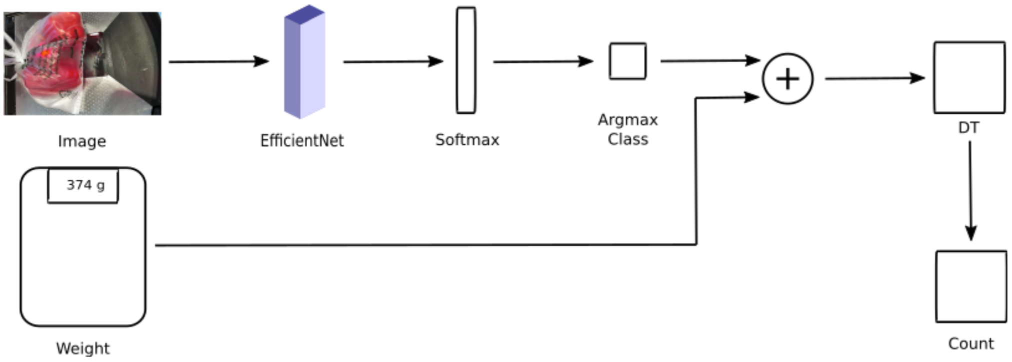


Figure 10: Flow-diagram of proposed hybrid architecture.

Table 1: Inference results.

Architecture	Inference Time (in sec.)	# Parameters
EfficientDet D0	0.41	4.3 M
EfficientDet D1	0.86	6.6 M
EfficientDet D2	1.43	8.1 M
MobileNetV2 (SSDLite)	0.36	-
Hybrid EfficientNet B0	**0.20**	**4.0 M**
Hybrid EfficientNet B1	0.28	6.5 M
Hybrid EfficientNet B2	0.36	7.8 M

Table 2: Evaluation results.

Architecture	*Acc*	*MAE*	*RMSE*	*Std*	*Variance*
EfficientDet D0	0.87	0.20	0.63	2.00	4.00
EfficientDet D1	0.86	0.20	0.57	2.04	4.18
EfficientDet D2	0.90	0.16	0.59	2.15	4.62
MobileNetV2 (SSDLite)	0.76	0.39	0.92	2.24	5.02
Hybrid EfficientNet B0	0.79	0.32	0.82	2.00	4.02
Hybrid EfficientNet B1	0.80	0.30	0.79	1.98	3.94
Hybrid EfficientNet B2	0.82	0.288	0.76	1.95	3.81

Accuracy (*Acc_counting*) is another important metric, which shows how many images are counted as well as classified correctly. We defined it as:

$$Acc\_counting = \frac{|\{i \in I | p(i) = g(i)\}|}{|I|}$$

where $|I|$ is the number of test set images and the counter is the number of prediction with correct class and count. The evaluation is extended by standard deviation and variance of predictions. Additionally, we report inference time on the CPU for processing of a single image.

Tab. 1 illustrates the comparison of inference time and model size (trainable parameters). The number of trainable parameters is according to original contribution in the case of EfficientDet. Each inference time calculation is a sum over all predictions divided by the number of all images, while processing each image separately. The inference time was measured as an average of ten calls. Tab. 2 shows an extensive performance accuracy. For fair comparison, only results on the same test data and machine settings are included. Standard deviation and variance are calculated over $\hat{p}(i)$.

6 DISCUSSION

The measurements show that the inference time on CPU ranges between 0.2 and 1.43 sec./image. We think that a processing time of over a second makes an application unusable in the context of sales. Because of this we can exclude EfficientDet D2.

Our proposed hybrid EfficientNet B0 is twice as fast as EfficientDet D0. This is due to the fact that no object detection is required, which we see as a huge advantage. Although the new methods cannot reach the same accuracy that comparable detectors do, we still

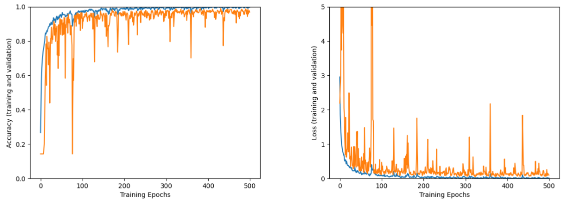
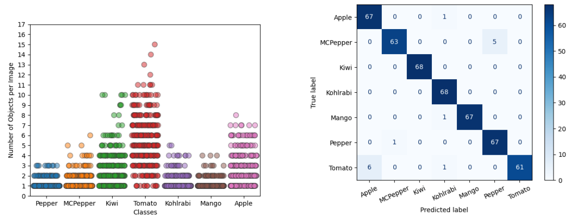
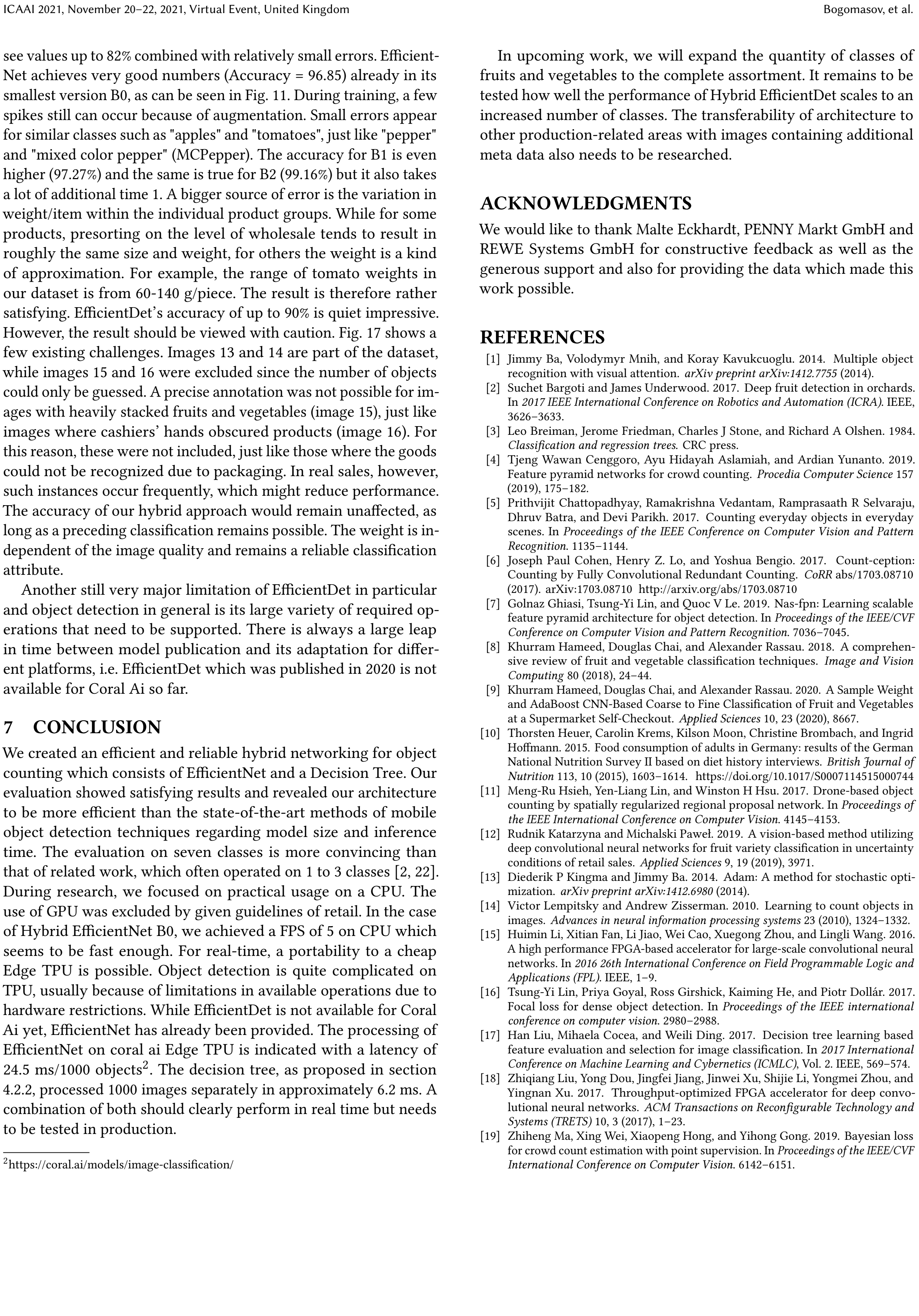


Figure 11: Training Accuracy and Training Loss of EfficientNet B0.



	Apple	MCPepper	Kiwi	Kohlrabi	Mango	Pepper	Tomato
Apple	67	0	0	1	0	0	0
MCPepper	0	63	0	0	0	5	0
Kiwi	0	0	68	0	0	0	0
Kohlrabi	0	0	0	68	0	0	0
Mango	0	0	0	1	67	0	0
Pepper	0	1	0	0	0	67	0
Tomato	6	0	0	1	0	0	61

Figure 12: Label distribution within the complete dataset and Confusion Matrix over EfficientNet B0 predictions.



see values up to 82% combined with relatively small errors. EfficientNet achieves very good numbers (Accuracy = 96.85) already in its smallest version B0, as can be seen in Fig. 11. During training, a few spikes still can occur because of augmentation. Small errors appear for similar classes such as "apples" and "tomatoes", just like "pepper" and "mixed color pepper" (MCPepper). The accuracy for B1 is even higher (97.27%) and the same is true for B2 (99.16%) but it also takes a lot of additional time 1. A bigger source of error is the variation in weight/item within the individual product groups. While for some products, presorting on the level of wholesale tends to result in roughly the same size and weight, for others the weight is a kind of approximation. For example, the range of tomato weights in our dataset is from 60-140 g/piece. The result is therefore rather satisfying. EfficientDet's accuracy of up to 90% is quiet impressive. However, the result should be viewed with caution. Fig. 17 shows a few existing challenges. Images 13 and 14 are part of the dataset, while images 15 and 16 were excluded since the number of objects could only be guessed. A precise annotation was not possible for images with heavily stacked fruits and vegetables (image 15), just like images where cashiers' hands obscured products (image 16). For this reason, these were not included, just like those where the goods could not be recognized due to packaging. In real sales, however, such instances occur frequently, which might reduce performance. The accuracy of our hybrid approach would remain unaffected, as long as a preceding classification remains possible. The weight is independent of the image quality and remains a reliable classification attribute.

Another still very major limitation of EfficientDet in particular and object detection in general is its large variety of required operations that need to be supported. There is always a large leap in time between model publication and its adaptation for different platforms, i.e. EfficientDet which was published in 2020 is not available for Coral Ai so far.

7 CONCLUSION

We created an efficient and reliable hybrid networking for object counting which consists of EfficientNet and a Decision Tree. Our evaluation showed satisfying results and revealed our architecture to be more efficient than the state-of-the-art methods of mobile object detection techniques regarding model size and inference time. The evaluation on seven classes is more convincing than that of related work, which often operated on 1 to 3 classes [2, 22]. During research, we focused on practical usage on a CPU. The use of GPU was excluded by given guidelines of retail. In the case of Hybrid EfficientNet B0, we achieved a FPS of 5 on CPU which seems to be fast enough. For real-time, a portability to a cheap Edge TPU is possible. Object detection is quite complicated on TPU, usually because of limitations in available operations due to hardware restrictions. While EfficientDet is not available for Coral Ai yet, EfficientNet has already been provided. The processing of EfficientNet on coral ai Edge TPU is indicated with a latency of 24.5 ms/1000 objects[2]. The decision tree, as proposed in section 4.2.2, processed 1000 images separately in approximately 6.2 ms. A combination of both should clearly perform in real time but needs to be tested in production.

In upcoming work, we will expand the quantity of classes of fruits and vegetables to the complete assortment. It remains to be tested how well the performance of Hybrid EfficientDet scales to an increased number of classes. The transferability of architecture to other production-related areas with images containing additional meta data also needs to be researched.

ACKNOWLEDGMENTS

We would like to thank Malte Eckhardt, PENNY Markt GmbH and REWE Systems GmbH for constructive feedback as well as the generous support and also for providing the data which made this work possible.

REFERENCES

[1] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. 2014. Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755* (2014).

[2] Suchet Bargoti and James Underwood. 2017. Deep fruit detection in orchards. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3626–3633.

[3] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. 1984. *Classification and regression trees.* CRC press.

[4] Tjeng Wawan Cenggoro, Ayu Hidayah Aslamiah, and Ardian Yunanto. 2019. Feature pyramid networks for crowd counting. *Procedia Computer Science* 157 (2019), 175–182.

[5] Prithvijit Chattopadhyay, Ramakrishna Vedantam, Ramprasaath R Selvaraju, Dhruv Batra, and Devi Parikh. 2017. Counting everyday objects in everyday scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 1135–1144.

[6] Joseph Paul Cohen, Henry Z. Lo, and Yoshua Bengio. 2017. Count-ception: Counting by Fully Convolutional Redundant Counting. *CoRR* abs/1703.08710 (2017). arXiv:1703.08710 http://arxiv.org/abs/1703.08710

[7] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. 2019. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 7036–7045.

[8] Khurram Hameed, Douglas Chai, and Alexander Rassau. 2018. A comprehensive review of fruit and vegetable classification techniques. *Image and Vision Computing* 80 (2018), 24–44.

[9] Khurram Hameed, Douglas Chai, and Alexander Rassau. 2020. A Sample Weight and AdaBoost CNN-Based Coarse to Fine Classification of Fruit and Vegetables at a Supermarket Self-Checkout. *Applied Sciences* 10, 23 (2020), 8667.

[10] Thorsten Heuer, Carolin Krems, Kilson Moon, Christine Brombach, and Ingrid Hoffmann. 2015. Food consumption of adults in Germany: results of the German National Nutrition Survey II based on diet history interviews. *British Journal of Nutrition* 113, 10 (2015), 1603–1614. https://doi.org/10.1017/S0007114515000744

[11] Meng-Ru Hsieh, Yen-Liang Lin, and Winston H Hsu. 2017. Drone-based object counting by spatially regularized regional proposal network. In *Proceedings of the IEEE International Conference on Computer Vision.* 4145–4153.

[12] Rudnik Katarzyna and Michalski Paweł. 2019. A vision-based method utilizing deep convolutional neural networks for fruit variety classification in uncertainty conditions of retail sales. *Applied Sciences* 9, 19 (2019), 3971.

[13] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[14] Victor Lempitsky and Andrew Zisserman. 2010. Learning to count objects in images. *Advances in neural information processing systems* 23 (2010), 1324–1332.

[15] Huimin Li, Xitian Fan, Li Jiao, Wei Cao, Xuegong Zhou, and Lingli Wang. 2016. A high performance FPGA-based accelerator for large-scale convolutional neural networks. In *2016 26th International Conference on Field Programmable Logic and Applications (FPL).* IEEE, 1–9.

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision.* 2980–2988.

[17] Han Liu, Mihaela Cocea, and Weili Ding. 2017. Decision tree learning based feature evaluation and selection for image classification. In *2017 International Conference on Machine Learning and Cybernetics (ICMLC)*, Vol. 2. IEEE, 569–574.

[18] Zhiqiang Liu, Yong Dou, Jingfei Jiang, Jinwei Xu, Shijie Li, Yongmei Zhou, and Yingnan Xu. 2017. Throughput-optimized FPGA accelerator for deep convolutional neural networks. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)* 10, 3 (2017), 1–23.

[19] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. 2019. Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 6142–6151.

[2] https://coral.ai/models/image-classification/

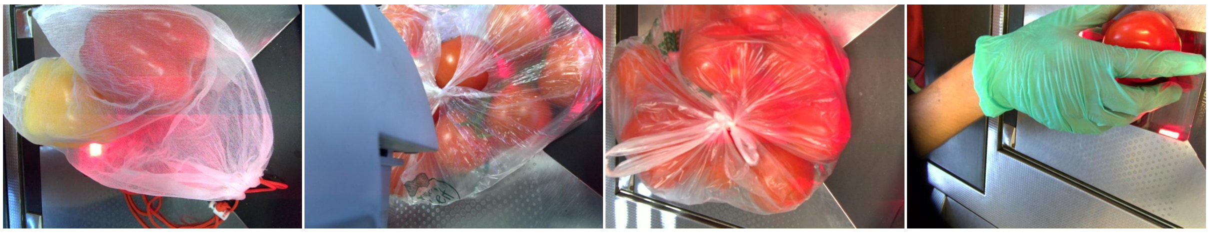


Figure 13: Unusual packaging **Figure 14: Misplaced camera** **Figure 15: Stacked products** **Figure 16: Hands**

Figure 17: Additional challenges

[20] T Nathan Mundhenk, Goran Konjevod, Wesam A Sakla, and Kofi Boakye. 2016. A large contextual dataset for classification, detection and counting of cars with deep learning. In *European Conference on Computer Vision*. Springer, 785–800.

[21] Arundhati Navada, Aamir Nizam Ansari, Siddharth Patil, and Balwant A Sonkamble. 2011. Overview of use of decision tree algorithms in machine learning. In *2011 IEEE control and system graduate research colloquium*. IEEE, 37–42.

[22] Maryam Rahnemoonfar and Clay Sheppard. 2017. Deep count: fruit counting based on deep simulated learning. *Sensors* 17, 4 (2017), 905.

[23] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[24] Jose Luis Rojas-Aranda, Jose Ignacio Nunez-Varela, Juan C Cuevas-Tello, and Gabriela Rangel-Ramirez. 2020. Fruit Classification for Retail Stores Using Deep Learning. In *Mexican Conference on Pattern Recognition*. Springer, 3–13.

[25] Inkyu Sa, Zongyuan Ge, Feras Dayoub, Ben Upcroft, Tristan Perez, and Chris McCool. 2016. Deepfruits: A fruit detection system using deep neural networks. *Sensors* 16, 8 (2016), 1222.

[26] BN Krishna Sai and T Sasikala. 2019. Object Detection and Count of Objects in Image using Tensor Flow Object Detection API. In *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*. IEEE, 542–546.

[27] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4510–4520.

[28] Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal* 27, 3 (1948), 379–423.

[29] B. A. Shepherd. 1983. An Appraisal of a Decision Tree Approach to Image Classification. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence - Volume 1* (Karlsruhe, West Germany) *(IJCAI'83)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 473–475.

[30] Sarah Steenhaut, Patrick Van Kenhove, et al. 2003. *Consumers' Reactions to" Receiving Too Much Change at the Checkout"*. Faculteit Economie en Bedrijfskunde.

[31] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*. PMLR, 6105–6114.

[32] Mingxing Tan, Ruoming Pang, and Quoc V Le. 2020. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10781–10790.

[33] R. Timofeev. 2004. Classification and Regression Trees(CART)Theory and Applications.

[34] Alexander Trott, Caiming Xiong, and Richard Socher. 2017. Interpretable counting for visual question answering. *arXiv preprint arXiv:1712.08697* (2017).

[35] Allard CR Van Riel, Janjaap Semeijn, Dina Ribbink, and Yvette Bomert-Peters. 2012. Waiting for service at the checkout. *Journal of Service Management* (2012).

[36] Bing-Fei Wu, Wan-Ju Tseng, Yung-Shin Chen, Shih-Jhe Yao, and Po-Ju Chang. 2016. An intelligent self-checkout system for smart retail. In *2016 International Conference on System Science and Engineering (ICSSE)*. IEEE, 1–4.

[37] Jincheng Yu, Kaiyuan Guo, Yiming Hu, Xuefei Ning, Jiantao Qiu, Huizi Mao, Song Yao, Tianqi Tang, Boxun Li, Yu Wang, et al. 2018. Real-time object detection towards high power efficiency. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 704–708.

[38] Jianming Zhang, Stan Sclaroff, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. 2015. Minimum Barrier Salient Object Detection at 80 FPS. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[39] Yan Zhang, Jonathon Hare, and Adam Prügel-Bennett. 2018. Learning to count objects in natural images for visual question answering. *arXiv preprint arXiv:1802.05766* (2018).

[40] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. 2016. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 589–597.

5.2 Supplement: Expansion of the product range

Subsequent to Bogomasov and Conrad (2021), the cooperation partners made the decision to expand the investments for markets with a large range of fruits and vegetables and equip the stores with GPUs. At this point, a distinction regarding the computational capabilities has to be made between a discount food retailer and a full-range market. As seen in the previous chapter, object detection promises a more precise detection than a stochastic approximation using weight information, presupposed that the objects to be detected are not widely covered by obstacles. A precise object detection, for all currently common architectures, presupposes that the entire range of known classes is present and labeled within the training data. Due to the nature of deep learning, only known classes can be recognized. The prerequisite for training a neural network is always labeling, a work which is mostly handled manually. Typical for food markets is a seasonal rotation of the assortment in the stores, as well as a regular expansion of offered products. While the product range that a discount food retailer offers has barely more than 70 basic products that keep coming back every year, the number of fresh fruits and vegetables in a full-range market contains up to 250 entities and is extended yearly. Concerning this matter, a continuing expansion by new classes is indispensable, which in terms entails a completely reapplied training routine, labeling work included. This is a huge drawback for a business that has to be agile.

Therefore, in the case of object localization, each sample of the training data set requires the corresponding bounding box information for each object containing its position. The amount of required manual annotation work is huge. In practice, however, it is not only the amount of annotation work that needs to be done that limits the applicability of object detection for the given task, but also the period of time until an updated model is accessible. Thus, the question arises **how to reduce the annotation effort and make object counting using object localization relevant for business**. The answer to this question would allow both, increasing flexibility in usage while simultaneously reducing costs. For a deeper investigation, an extended data set was made available. This set contains 36 classes from currently available products of a discount supermarket. Each class has 300 representatives. This results in 10800 images in total. Due to the nature of image acquisition during the sales process, class labeling is known for each image, since cashiers already labeled the products at the checkout system. This is an important benefit that we want to take advantage of. However, neither the number of objects nor their position are known and had to be labeled manually for a proper evaluation as well as the initial subset. The concept of the initial subset will be explained later in this chapter. The following set C of 36 classes is available: {apple, kiwi, orange, radish, aubergine, banana, sweet potato, blue grapes, pomegranate, lemon, celery, pickle, green apple, leek, mango, fennel, nectarine, vine tomato, garden leek, chiquita banana, white cabbage, bio banana, light grapes, cauliflower, lettuce, lychee, carrot, bio hokkaido pumkin, mandarin with leaves, bio pepper mix, pear, zucchini, avocado, ginger, grapefruit, kohlrabi}. A visualization of the listing can be seen in Fig. 5.2 and Fig. 5.3.

To tackle this problem and reduce the overhead of annotation work of the full set of classes without human interaction, we had the idea to find a subset of products which consists of distinctive objects. These objects need to be annotated manually. In our imagination, the semantical similarity among the selected classes in each case is

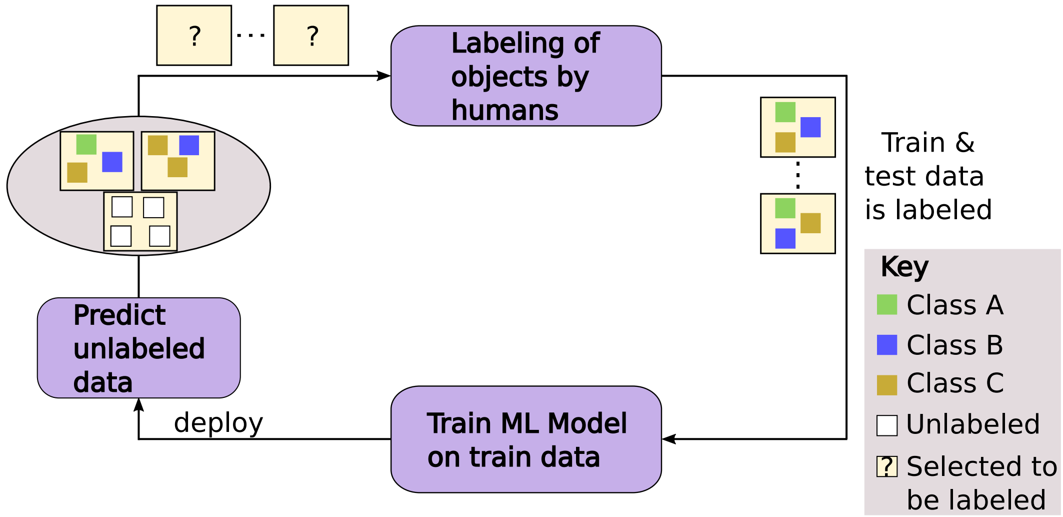


Figure 5.1: Overview of a simple human-in-the-loop machine learning system. Adapted from (Munro, 2020) to OL

less present than that among the classes that are left out. The hypothesis we stated implies that some of the classes share similar characteristics such as shape, color and texture which the implemented network learns to be able to separate objects from the background. This idea can be summarized as semi-supervised *active learning* (AL). Usually, AL is seen as the process of selecting the right data for human review (Munro, 2020). The reviewer then decides if the prediction quality is satisfying or adjusts it if necessary. In contrast, the proposed algorithm does the adjustments independently (Step 2). There are three main active learning strategies: diversity (selecting the most diverse), uncertainty and random. Probably the most conventional is sampling for uncertainty, which takes the difference between the maximally possible percentage and the highest predicted label confidence for each object. A popular methodology to semi-automated annotation is to use the model itself for predictions annotation. This approach is called semi-supervised learning (Munro, 2020). A basic architecture including human as well as ML model as annotators is visualized in Fig. 5.1. The so-called human-in-the-loop machine learning system iteratively provides data that is labeled by humans and the ML model for human review or labeling. On this basis, a couple of AL-publications for OD have been introduced (Choi et al., 2021), (Haussmann et al., 2020), (Kao et al., 2018), (Brust et al., 2018), (Yuan et al., 2021). However, the main differences to our approach are that on the one hand we do not need to evaluate the uncertainty or score images for informativeness and on the other that there is no need in removing noisy images. Additionally, these related systems still rely on manual effort while learning procedure and is therefore not limited to the one-time work at the beginning in terms of annotation effort. An additional advantage of our approach is that we do not have a composition of different models but use a single one. In the following we explain our contribution in detail:

The idea can be divided into three logical steps: In **Step 1** an initial subset is built. It contains only selected classes and is used for the training of a mobile object detection network. As EfficientDet D0 (M. Tan et al., 2020) has proven great performance in previous works, it is selected as the core OD module. In **Step 2** the model is trained on the initial subset. Afterwards, an inference on the left-out classes of the training set is calculated. The resulting output is inaccurate and contains false labels only, therefore it needs to be post-processed. The post-processing procedure is based on filtering, non-maximum suppression, relabeling and augmentation. Finally, in **Step 3** the training is resumed on the union of the initial subset and in Step 2 generated pseudo-labels. It is continued as long as the evaluation on the test data, consisting of the full set of classes, shows any improvement. All experiments were based on a subdivision of 80:20, while 80% of data was used for training, 20% was left out for testing. The next paragraph describes the algorithm in detail.

- **Step 1: Building initial subsets**

  Possibly the simplest way in finding the greatest subset is a grid search, building subsets over all combinations of its elements. In the case of the given 36 elements $c \in C$ the quantity of possible subsets is $2^{36} - 2 = 68719476734$, empty set excluded. This is an absolutely gigantic number for practical experiments. Diverse strategies may reduce the number of subsets as initial candidates. The most promising because of its efficiency is a genetic algorithm, as described in Loussaief and Abdelkrim (2018). Further on, each of the subsets $C_{initial} \subset C$ is used as input for the second step.

- **Step 2: Iterative semi-supervised object detection training**

  The starting point for each supervised and semi-supervised learning is providing labeled examples. Based on this input data the training procedure is continued at best as long as something new is being learned and stopped as soon as the evaluation results stop improving. Training stopped at the maximum is a great way to avoid overfitting and reduction of generalization appropriately. Therefore the initial subset that was defined in Step 1 is provided to Algorithm 1 as input for training with the main intention to learn $C_{initial}$ classes. Meanwhile, the set of labels for images of the target classes $\mathcal{L}_{target}$ remains empty. Having a fully trained model, predictions are calculated on the remaining classes C_{target}. All images I_{target} belonging to new classes are unlabeled as unseen at this point. Therefore it is to be assumed that the prediction results are subject to large fluctuations. To counteract such inaccuracy, in **filterOnConfidence** only bounding boxes with particularly high confidence τ are extracted. We use a dynamic value $\tau = 0.7$ as a starting point. This value is reduced in constant steps of 0.1 until the returned result list of predictions contains at least one object. What is important to note here is that, in the case of the presence of multiple objects of the same class, a similar confidence score is assigned to all objects of the same appearance. This leads naturally to the fact that there is a great chance to extract all found objects at once, not just the first that has been found.

  In the next step, called **nonMaxSupression** (NMS), all redundant bounding boxes are deleted by applying the name-giving algorithm (Bodla et al., 2017). Finally, the remaining predictions in **relabelBoundingBox** are set to the correct class. In spite of filtering on confidence, some predictions may be empty. In this case, **registerMissing** is called to make a record of an image without detected objects for each class $c \in C_{target}$. An important aspect to the most ML algorithms is the overall balance of classes in a data set. Therefore the amount of images with empty predictions needs to be filled up. For this purpose, we use augmentation methods such as rotation, flipping, blurring and adding noise. In terms of source images for the augmentation function, the successfully created pseudo-labeled images (**pLD**) are passed. Finally, the set of previously used training images is expanded by the newly annotated set and used in the following step 3 for subsequent training steps.

- **Step 3: Resumption of the training**

  The algorithm 2 shows the continuation of the learning processes on the complete data set, including the artificial pseudo bounding boxes including true class labels. The training continues as long as the performance of the model improves. Our previously defined evaluation metrics (Bogomasov and Conrad, 2021) are used for performance measurements. In case the inference for a single image contains more than one class label, a minority rule is applied to stabilize the output as part of post-processing in **evaluate**.

Algorithm 1: First step of the iterative semi-supervised object detection training

```
Input  : a bag of images of the train split $I_c$ for each class of products
         $c \in C = C_{initial} \cup C_{target}$ and $C_{initial} \nsubseteq C_{target}$, labeling for images of
         the initial classes $\mathcal{L}_{initial}$, confidence threshold $\rho$, number of required
         images $n_{train}$, object detection model $M$, number of epochs $n_{epochs}$,
         initial confidence $\tau$, pretrained weights $W$
Output : Predicted labeling $\hat{\mathcal{L}}_{target}$ for images of the target domain, model
         weights $\hat{W}$
$M, \hat{W} \leftarrow$ trainModel($M, I_c, \mathcal{L}_{initial}, C_{initial}, n_{epochs}, W$)
save($\hat{W}$)
$\hat{\mathcal{L}}_{target} \leftarrow$ predict($M, C_{target}$)
 foreach $c_i \in C_{target}$ do
    foreach $l_j \in \hat{\mathcal{L}}_{target}$ do
        if $l_j \neq \emptyset$ then
            $l_j \leftarrow$ filterOnConfidence($l_j, \tau$)
            $\hat{l_j} \leftarrow$ nonMaxSupression($l_j$))
            $\hat{l_j} \leftarrow$ relabelBoundingBoxes($\hat{l_j}, c_i$)
            $pLD \leftarrow$ append($i_j, \hat{l_j}$)
            appendToInitialTrainSet($i_j, \hat{l_j}$)
        end
        else
            $missing_{ij} \leftarrow$ registerMissing($c_i, i$)
        end
    end
end
$\hat{I}_c$, $\hat{L}_c \leftarrow$ augment($missing, pLD, n_{train}$)

appendToInitialTrainSet($\hat{I}_c, \hat{L}_c$)
save($I_c \leftarrow I_c \cup \hat{I}_c$)
save($\mathcal{L}_{target} \leftarrow \mathcal{L}_{initial} \cup \hat{\mathcal{L}}_{target}$)
```

Algorithm 2: Resumption of the iterative semi-supervised object detection training

```
Input  : a bag of images of the train and test splits $I_{train \cup test}$ for each class of
         products $c \in C = C_{train} \cup C_{test}$ and $C_{train} = C_{test}$, labeling for images
         $\mathcal{L}_{train \cup test}$ a sequence of evaluation metrics $\Psi = (f_1, f_2, \ldots, f_n)$, object
         detection model $M$, maximum number of epochs $n_{epochs}$, starting
         learning rate $\tau$
Output : model weights $\hat{W}$, metrics $\hat{\Phi}$
foreach $f_i \in \Psi$ do
    $f_i \leftarrow 0$
end
while $n_{epochs} > 0$ do
    $M, \hat{W} \leftarrow$ trainModel($M, I_{train}, \mathcal{L}_{train}, C_{train}, n_{epochs}, \hat{W}, \tau$)
    $n_{epochs} \leftarrow n_{epochs} - 1$
    if $n_{epochs}$ mod 10 then
        $\hat{\Phi} \leftarrow$ evaluate($M, I_{test}, \mathcal{L}_{test}, C_{test}, \hat{W}$)
        if $\exists Nin\mathbb{N} : (N \leq n \land \phi_n < \hat{\phi_n})$ then
            $\Phi \leftarrow \hat{\Phi}$;
        else
            break
        end
    end
end
save($\hat{W}$)
save($\Phi$)
```

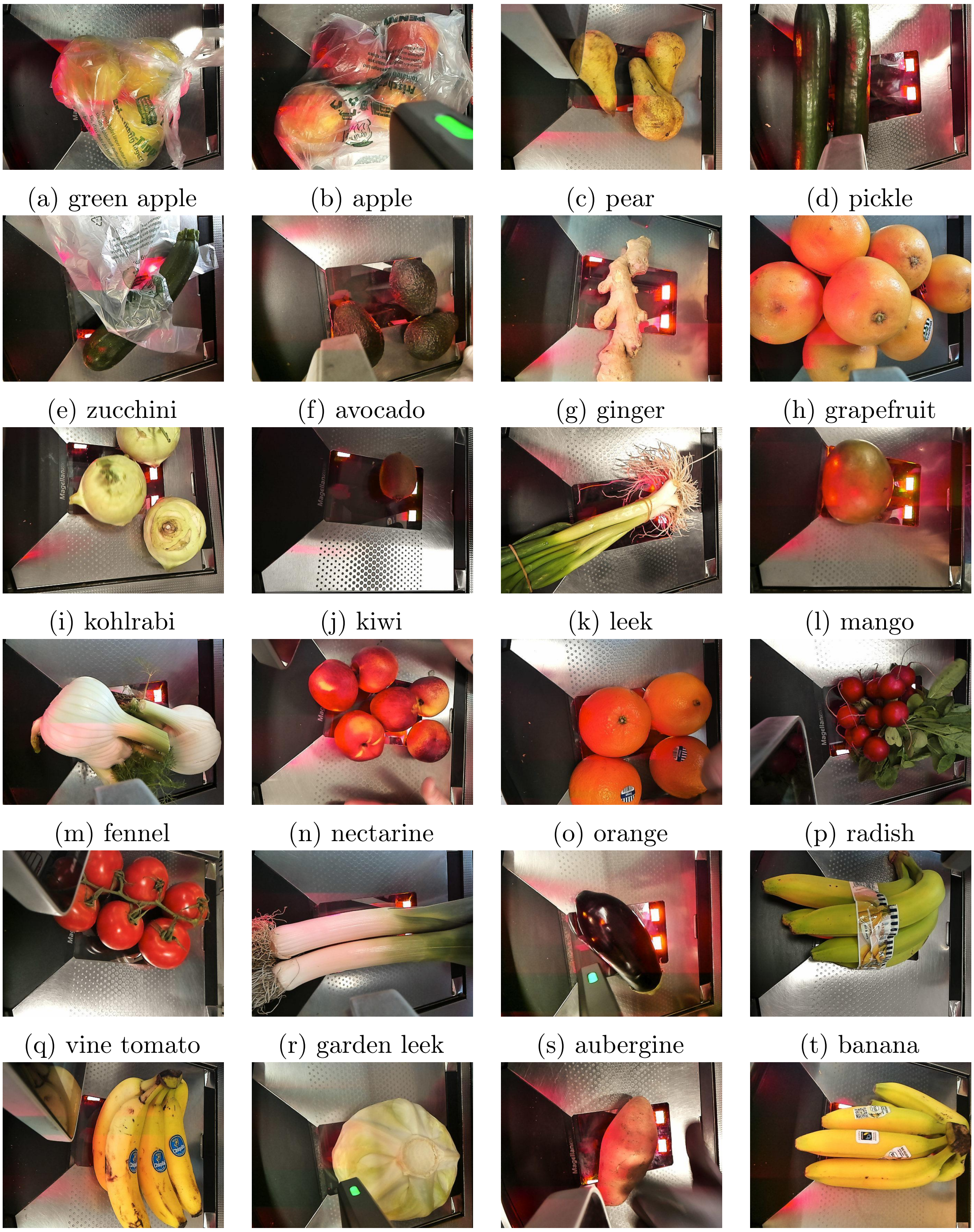
Set	Acc	$RMSE$	Std	$Variance$
Best subset 1	0.83	0.70	0.76	0.58
Full set	0.86	0.67	1.11	1.24

Table 5.1: Counting results on 36 classes

Among the examined subsets $c \in C$, the following subset $C_{initial}$ has shown the best results {apple, aubergine, banana, blue grapes, celery, green apple, kiwi, lemon, orange, pickle, pomegranate, radish, sweet potato}. Tab. 5.1 shows a comparison between the training on the entire data set and the chosen subset as initial using the proposed algorithm.

The evaluation results do not show that the final subset is objectively the most suitable and provides the greatest disparity over the selected classes. Much more can be seen from the fact that the developed solution already achieves great results and reduces the annotation overhead at this point. Based on this insight, we examined whether reducing the subsets by individual classes would lead to equivalent results. It is noticeable that the reduction of the final proposed set by any of the included classes leads to the fact that the overall performance error grows. Obviously, the choice of classes is an important decision. Due to the computational effort, while forming the initial classes, we could not try out all classes C using grid searches. In light of the computing effort of an average of four days for one iteration on a Tesla GPU, even including a genetic search algorithm, the total computing time was not reasonable. For this reason, we choose a random selection for all compilations. Furthermore, the results show that on the one hand, the manual annotation of a small proper subset is sufficient for a precise annotation as the result of the full algorithm cycle, while further classes are being learned in an iterative learning step, assuming a single class multi label task. Moreover, we suspect that while training with a range of 36 known classes we already cover the semantic diversity of foreground objects to a great extent. In turn, this assertion means that any future classes, while being semantically comparable to the known classes, may be separated from the background by the currently trained model. This effectively allows adding new products and keeping the architecture up-to-date without manual annotation effort. Already a few sample images of the new product class are sufficient for retraining the network starting at Step 2 with a prediction. Afterwards, an updated network can be used again for QA in retail. While for most of the products the expected prediction performance of the model is promising, in some cases the counting accuracy may drop. Although we could not observe such behavior in the currently available data, it cannot be fully excluded. A possible reason for this deviation may be a strong visual dissimilarity including differences in shape, color and texture to previously known data. These particular cases would still require manual annotation. However, the expected benefit of the proposed approach could reduce the manual annotation effort significantly. Since we use only 13 out of 36 classes for annotation, the overall saving is around 64%. The evaluation on the best set found shows that our algorithm achieves results comparable to the training on the complete data. In terms of counting accuracy, we achieve values of 83% to 86%.

Object detection proved great results for the task of counting fresh fruits and vegetables using GPU power. However, the limitation of object detection as part of CV is always that objects that are not visible, are impossible to detect. In this situation, we can rely on the previously introduced Hybrid EfficientNet. A combination of both algorithms is even more promising. In order to get the maximum out of both systems in business applications, instead of images the entire video sequence could be analyzed in real-time. For optimisation purposes, the sales process for each product or set of products can be logically divided into several steps. In the first step, the field of vision should be verified by means of completeness and excluding stationary obstacles and thus allowing object tracking. In the next step, products could be classified and counted over the entire period of their movement over the scale surface. Afterwards, while no longer being moved, a prediction result would appear on the screen. Accordingly, the inference could be calculated as an arithmetic mean of inferences, comparable to 3.2, made over the entire period of each particular set of products passing the scale. This would lead to additional stability without restricting the noticeable performance in the perception of the cashier. However, this is not part of this work, but an outlook for future developments.



(a) green apple (b) apple (c) pear (d) pickle

(e) zucchini (f) avocado (g) ginger (h) grapefruit

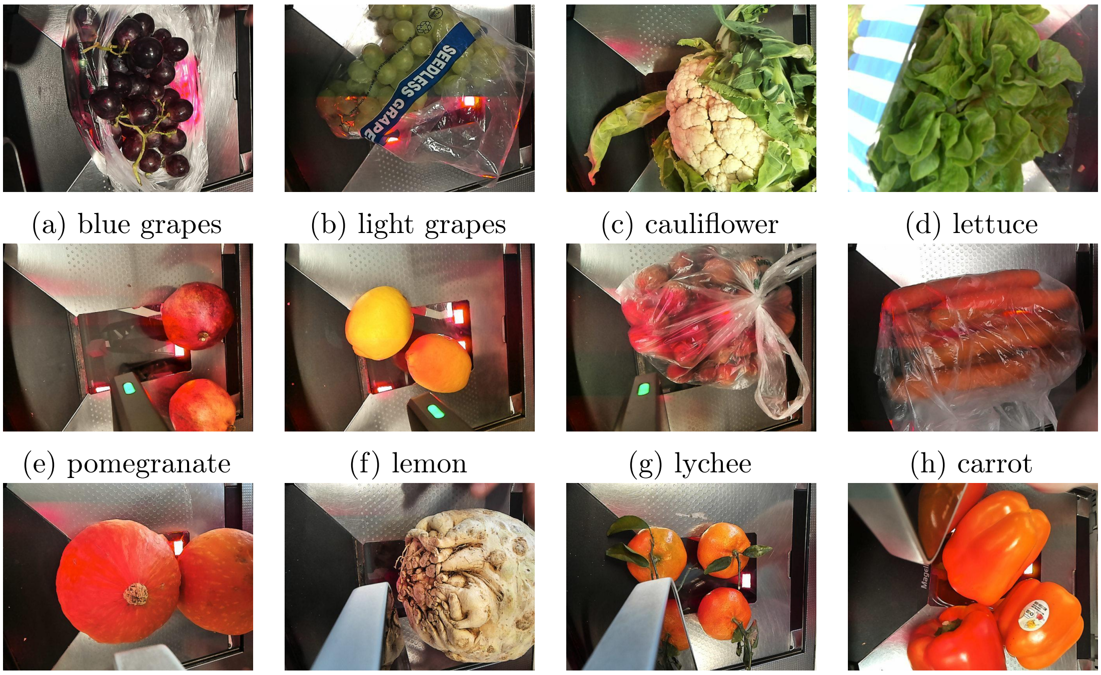
(i) kohlrabi (j) kiwi (k) leek (l) mango

(m) fennel (n) nectarine (o) orange (p) radish

(q) vine tomato (r) garden leek (s) aubergine (t) banana

(u) banana chiquita (v) white cabbage (w) sweet potato (x) banana bio

Figure 5.2: Examples of new classes



(a) blue grapes (b) light grapes (c) cauliflower (d) lettuce

(e) pomegranate (f) lemon (g) lychee (h) carrot

(i) pumpkin h. bio (j) celery (k) mandarin (leaves) (l) pepper mix bio

Figure 5.3: Examples of new classes

6

Conclusion

This chapter contains the conclusion of the thesis. In Section 6.1, we summarize the contributions of our research and describe the results of the addressed problems. In Section 6.2, we discuss the results we achieved and deal with insights accomplished. Finally, Section 6.3 presents starting ideas for future research, further subdivided into corresponding research areas.

6.1 Overall Summary

In this thesis, the focus has been laid on extracting useful knowledge from image data with the goal of enabling automation and quality assurance of visual processes. In order to diversify the research and contribute more general results, we decided to include real world data from several thematically independent application fields. Therefore we analyzed four data sets that share the data type, to be specific — digital image and also the common goal — automation and QA of processes using CV. For this purpose, in Chapter 2, we gave a short introduction into basic concepts of ML and CV that are relevant to this work. In Chapter 3, we addressed the task of QA for two medical applications. Therefore, we focused on the analysis of 3D images. Firstly, we have provided convincing algorithmic solutions for severity scoring and automatic report generation of lung tuberculosis. Secondly, we introduced in Chapter 3.3 the idea of QA for prostate MRI with the purpose of controlling the orientation of recordings and presented a reliable solution as well. Both algorithmic solutions aim to reduce inaccuracies and thus avoid diagnostic errors. Furthermore, in Chapter 4, we have changed the data to large scale images, which in turn have their own peculiarities, and researched the possibility of detecting different types of corals in underwater images. The result offers, for the first time, the possibility of computer-aided monitoring of maritime inventory, which is necessary for offering species-appropriate conservation as well as countermeasures in terms of the effects of global warming. Finally, we set the goal to make the transition of CV support systems to business applications in Chapter 5. As it has turned out, the previously introduced algorithms were not suitable for applications with time and resource constraints. Because of that, we focused on efficiency and presented another architecture for classification and counting, specialised in products passing a conveyors belt in retail. In particular, the presented approach is capable of reducing manual effort. Furthermore, it reduces costs and errors. All in all, this thesis provides a compilation of ML solutions on different real world data sets and reveals the benefits and limitations of CV for automation and QA of visual processes.

6.2 Discussion

After accomplishing extensive research on different kinds of images with feature-based models, we can conclude that while they are often useful and constructive, the future of automation and QA for visual processes lies more in the area of deep learning. This is due to the fact that while self designed features are limited to defined formulations, deep learning methods are not. They share the ability to learn properties that have been classified as relevant by the system itself, a great advantage for data with complex texture, structure or color distribution, or even a combination of these properties. Especially in real world data, these factors frequently come together. Overall, it can be stated that DL will be an indispensable part of QA systems for both image data as well as visual data consigned with meta information. Another insight that should not go unmentioned, is that based on the chosen heterogeneous data we can conclude that the state-of-the art architectures, irrespective of their theoretical background, are rarely the ne plus ultra for other, even comparable, application fields. These are usually optimized for more or less context-similar benchmark data sets, such as the most common COCO or Pascal VOC in case of object recognition for instance. The similarities among e.g. image resolution, object size, class distribution and object complexity are greater than the differences. The reality, however, looks quite the opposite. We often encounter data that differs significantly from the benchmark data sets, but also data that is clearly superior to the benchmark data sets in terms of complexity. In such quite specific, but regularly occurring real world scenarios, a case study as well as the analysis of the particular demands are required. To put it in detail: A DL architecture that showed great performance on benchmark data, such as Pascal VOC, probably will not show the same results on underwater large scale images. On the contrary, a solution tailored to the application is more likely to meet expectations using the data conditions. Likewise, an architecture designed for high-resolution images is not very suitable for doing a corresponding job on medical data.

We have elaborated various AI strategies. These strategies have their fundamentals in either DL or conventional features engineering in connection with established ML. In both cases, the prime aim is the accomplishment of a stable generalization. Along with a fitting model design, generalization is a matter of the amount of sample data. Although DL proved to be a better choice in most cases, some exceptions exist, as seen in Chapter 4.1. Any time the data may be considered complex, in terms of its internal diversity and in relation to the low quantity of labeled samples, a conventional feature engineering solution may still be preferred. Having a large number of samples, DL might be a better choice. The features that a DL architecture learns in order to be able to solve the mapping from input to a pre-specified output are often unknown and therefore more difficult to control, a property which can be both — a disadvantage and an advantage as well. Detailed insights on meaningful areas may be provided by analyzing the heat maps. Decisive region identification i.e., explainable AI, is often considered helpful. Indeed, these technologies share the limitation that statements are made on the basis of selected examples, while in the case of feature-based methods, the specified rules provide full control over the results. Therefore, if observations are made, the confidence is more comprehensible compared to DL and, thereby, more preferred. Concerning less complex images, the NN is also able to converge on a small number of examples and accordingly make a choice confidently, similar to feature-based methods. It follows that there is a trade-off between the complexity of the image associated with the task and the number of examples required for generalization. Hence, ML is always about the evaluation of the striven generalizability.

The presented research work focuses on the technical point of view of QA and process automation providing massive benefits to the respective target group. Apart from all the advantages, the work might entail an ethical and social component. However, predicting the social impact is an open question on its own and requires a separate expertise. It is therefore not part of this work. For a first insight into this topic, we recommend the work of Helbing (2019).

Referring to the main research question on how the image data can be computationally processed, analyzed and interpreted to allow automation of a conventionally manual process and assure its quality, and summing up the results, CV proved to offer an enormous potential. The presented research work shows which possibilities for automation and quality assurance of processes in the various application areas of the real world exist using examples of representatively selected application scenarios in relation to the developed algorithms. In this context, the origin of the image data hardly seems to matter. After finishing the research work on four use cases, presented in this thesis, we can say that the available technological tools, regardless of the processing unit i.e. GPU, CPU or even TPU, already allow the development of stable and reliable solutions. These solutions can both, reduce the manual effort of various processes and thus save time and costs, as well as decrease the error rate of manual processes. Crucial to successful automation and QA of processes that rely on visual information is that visual data provides the information necessary to answer a specific question or to ensure automation in particular. Since this might not always be the case, supplementary meta-information, if available, can be used as shown by the example of weight information in Chapter 5. However, the image characteristics, such as resolution and quality, along with the number of available samples are still crucial factors in finding a reliable solution. As is so often the case, the CV algorithms too become little by little more accurate and stable over time. Thus, in the future, numerous application fields for QA and automation will arise.

6.3 Outlook and Future Research

In this thesis, we have presented multiple successful CV approaches for a number of different application fields focusing on the task of automation and QA. The imposed requirements in terms of accuracy and, if needed, processing time constraints while adhering to hardware limitations were met to a great extent. However, we faced several technical limitations. These were often due to the availability of essential hardware such as high memory GPUs but also to the quality and size of the data sets. Truly, the aphorism that “any data are better than no data” (Shahian et al., 2016) is wrong since it can produce serious unintended consequences. While the last two aspects, i.e. the quality and the size, are often improvable as described in respective chapters of this work, the hardware issues are beyond our influence. For example, in the case of the ImageCLEF coral challenge, the reduction of the input size of data for processing with CNN architectures was inevitable because of the Video Direct Access Memory (VRAM) limitation of currently accessible GPUs. Nevertheless, it is not entirely unreasonable that an NN with the ability to use unscaled images in its input layer would lead to more accurate results. In this context, however, an input layer with greater capacity can only be handled at the expense of network depth, which in turn is also crucial for reliable performance. A way out could be found by dividing the source images into overlapping sub-images using a sliding window approach and subsequently using them as input data unscaled. In this particular case, the images for predictions would have to be split in the same manner and compounded afterwards to the complete image. Under these circumstances, this approach requires a multiple of training time while using the unchanged strategy as presented in Chapter 4, and can be estimated to take months on current hardware. Fortunately, the technological development of GPUs is progressing rapidly, so that processing of unscaled images will be possible in the foreseeable future and thus, greater performance on large scale images can be assumed leaving the same architectures unchanged. In the field of QA and automation of prostate cancer diagnosis, we suggest to follow the previously mentioned plan of developing a system based on the guidance of the Prostate Imaging Reporting and Data System (PI-RADS v2.1). In this thesis, we already presented the first step of the planned system in Chapter 3.3. The second step could consist of image quality evaluation such as formulated in PI-RADS v2.1. However, this step could not be developed yet, since the preliminary annotation work was not provided by the cooperating physicians so far. The final vision on the QA system can be designed according to a “traffic light” principle, which during the recording of MRI, based on the preceding steps, classifies the image quality and provides a supportive indication. Negative feedback, as expressed by the “red light”, can in turn serve as an indication of poor recording quality and lead to an instruction to automatically repeat the recording process.

Abbreviations

ACC accuracy

ACR video direct access memory

AI artificial intelligence

AP average precision

AUC area under the roc curve

BB minimum bounding box

BITMAP BMP file format

CAR Association of Radiologists

CH convex hull

CNN convolutional neural network

COCO common objects in context

CS computer science

CT computed tomography

CV computer vision

DI digital image

DIA digital image analysis

DICOM Digital Imaging and Communications in Medicine

DIP digital image processing

DL deep learning

DM data mining

DS data science

DT decision tree

EPS encapsulated Postscript

FL focal loss

FN false negative

FP false positive

FPN feature pyramid network

GIF graphics Interchange Format

HU Hounsfield units

JPEG joint Photographic Experts Group

KNN k-nearest-neighbour

LR linear regression

MAE mean absolute error

MBC minimum bounding circle

MBE minimum bounding ellipse

MI medical imaging

ML machine learning

MM millions

MRI magnetic resonance imaging

MTB mycobacterium tuberculosis

NC minimum bounding n-corner

NIfTI Neuroimaging Informatics Technology Initiative

OD object detection

OL object localization

OR object recognition

PCA principal component analysis

PNG portable Network Graphics

PR precision

QA quality assurance

QC quality control

QI quality improvement

REC recall

RF random forest

RGB RGB color space

RL reinforcement learning

RMBB rotated minimum bounding box

RMSE root mean square error

RQ research question

SD standard deviation

SGD stochastic gradient descent

SL supervised learning

SVM support vector machine

TIFF tagged Image File Format

TN true negative

TP true positive

UL unsupervised learning

VA visual assurance

VAR variance

VI visual inspection

VOC pascal visual object classes challenge

VRAM video direct access memory

References

Apicella, Andrea, Donnarumma, Francesco, Isgrò, Francesco, and Prevete, Roberto (2021). “A survey on modern trainable activation functions”. In: *Neural Networks* 138, pp. 14–32.

Association, Canadian Medical et al. (1992). “Quality of care: 1. What is quality and how can it be measured? Health Services Research Group”. In: *CMAJ* 146.12, pp. 2153–2158.

Bashir, Daniel, Montañez, George D, Sehra, Sonia, Segura, Pedro Sandoval, and Lauw, Julius (2020). “An information-theoretic perspective on overfitting and underfitting”. In: *Australasian Joint Conference on Artificial Intelligence*. Springer, pp. 347–358.

Batchelor, Bruce and Waltz, Frederick (2001). “Machine vision for industrial applications”. In: *Intelligent Machine Vision*. Springer, pp. 1–29.

Bera, Somenath and Shrivastava, Vimal K (2020). “Effect of pooling strategy on convolutional neural network for classification of hyperspectral remote sensing images”. In: *IET Image Processing* 14.3, pp. 480–486.

Beyerer, Jürgen, León, Fernando Puente, and Frese, Christian (2015). *Machine vision: Automated visual inspection: Theory, practice and applications*. Springer.

Bingham, Garrett, Macke, William, and Miikkulainen, Risto (2020). “Evolutionary optimization of deep learning activation functions”. In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pp. 289–296.

Bisiani, Roberto (1987). *Fundamentals in computer understanding: speech and vision*. CUP Archive.

Bodla, Navaneeth, Singh, Bharat, Chellappa, Rama, and Davis, Larry S (2017). “Soft-NMS–improving object detection with one line of code”. In: *Proceedings of the IEEE international conference on computer vision*, pp. 5561–5569.

Bogomasov, Kirill (2016). “FSC-FloodFill Sky Classification”. In: *Proceedings of the 28th GI-Workshop Grundlagen von Datenbanken, Nörten Hardenberg, Germany, May 24-27*, pp. 27–32. URL: `http://ceur-ws.org/Vol-1594/paper6.pdf`.

Bogomasov, Kirill, Braun, Daniel, Burbach, Andreas, Himmelspach, Ludmila, and Conrad, Stefan (2019a). “Feature and Deep Learning Based Approaches for Automatic Report Generation and Severity Scoring of Lung Tuberculosis from CT Images.” In: *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 09-12*. CEUR-WS.org.

Bogomasov, Kirill and Conrad, Stefan (2021). “Efficient Fruit and Vegetable Classification and Counting for Retail Applications Using Deep Learning”. In: *Proceedings of the 5th International Conference on Advances in Artificial Intelligence*.

Bogomasov, Kirill, Geuer, Tim, and Conrad, Stefan (2023). “InfEval: Application for Object Detection Analysis”. In: *European Conference on Information Retrieval.* Springer.

Bogomasov, Kirill, Grawe, Philipp, and Conrad, Stefan (2019b). “A two-staged Approach for Localization and Classification of Coral Reef Structures and Compositions.” In: *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 09-12.* CEUR-WS.org.

Bogomasov, Kirill, Grawe, Philipp, and Conrad, Stefan (2020). “Enhanced Localization and Classification of Coral Reef Structures and Compositions”. In: *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25.* CEUR-WS.org.

Bogomasov, Kirill, Himmelspach, Ludmila, Klassen, Gerhard, Tatusch, Martha, and Conrad, Stefan (2018). “Feature-Based Approach for Severity Scoring of Lung Tuberculosis from CT Images.” In: *Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14.* CEUR-WS.org.

Bogomasov, Kirill, Thomas, Grings, Rubbert, Christian, Schimmöller, Lars, and Stefan, Conrad (2021). “Orientation Estimation in MRI of Prostate Cancer Patients: When Simple Models Perform Better”. In: *2021 IEEE International Conference on Imaging Systems and Techniques (IST).* IEEE, pp. 1–6.

Bouma, Gary D, Ling, Rod, and Wilkinson, Lori (2004). “The research process”. In.

Boureau, Y-Lan, Ponce, Jean, and LeCun, Yann (2010). “A theoretical analysis of feature pooling in visual recognition”. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 111–118.

Brazdil, Pavel, Carrier, Christophe Giraud, Soares, Carlos, and Vilalta, Ricardo (2008). *Metalearning: Applications to data mining.* Springer Science & Business Media.

Brinkhoff, Thomas, Kriegel, H-P, and Schneider, Ralf (1993). “Comparison of approximations of complex objects used for approximation-based query processing in spatial database systems”. In: *Proceedings of IEEE 9th International Conference on Data Engineering.* IEEE, pp. 40–49.

Brust, Clemens-Alexander, Käding, Christoph, and Denzler, Joachim (2018). “Active learning for deep object detection”. In: *arXiv:1809.09875.*

Cadík, Martin (2008). “Perceptual Evaluation of Color-to-Grayscale Image Conversions”. In: vol. 27. 7. Wiley Online Library, pp. 1745–1754.

Campbell, Stephen M, Roland, Martin O, and Buetow, Stephen A (2000). “Defining quality of care”. In: *Social science & medicine* 51.11, pp. 1611–1625.

Cao, YuShe, Niu, Xin, and Dou, Yong (2016). “Region-based convolutional neural networks for object detection in very high resolution remote sensing images”. In: *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD).* IEEE, pp. 548–554.

Caridade, Cristina MR and Marçal, André RS (2019). “Automatic Classification of Coral Images using Color and Textures.” In: *CLEF (Working Notes).*

Castleman, Kenneth R (1996). *Digital image processing.* Prentice Hall Press.

Che, Shuai, Boyer, Michael, Meng, Jiayuan, Tarjan, David, Sheaffer, Jeremy W, and Skadron, Kevin (2008). “A performance study of general-purpose applications on graphics processors using CUDA”. In: *Journal of parallel and distributed computing* 68.10, pp. 1370–1380.

Chen, Wei, Liu, Yang, Wang, Weiping, Bakker, Erwin M, Georgiou, TK, Fieguth, Paul, Liu, Li, and Lew, MSK (2021). "Deep image retrieval: A survey". In: *arXiv:2101.11282*.

Chen, Yucheng, Tian, Yingli, and He, Mingyi (2020). "Monocular human pose estimation: A survey of deep learning-based methods". In: *Computer Vision and Image Understanding* 192, p. 102897.

Chen, Yuxin and Krause, Andreas (2013). "Near-optimal batch mode active learning and adaptive submodular optimization". In: *International Conference on Machine Learning*. PMLR, pp. 160–168.

Choi, Jiwoong, Elezi, Ismail, Lee, Hyuk-Jae, Farabet, Clement, and Alvarez, Jose M (2021). "Active learning for deep object detection via probabilistic modeling". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10264–10273.

Chollet, François (2017). "Xception: Deep learning with depthwise separable convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258.

COCO evaluatation metrics, (2020). URL: `https://cocodataset.org/#detection-eval` (visited on 03/20/2021).

Conrad, Stefan, Tatusch, Martha, Bogomasov, Kirill, and Klassen, Gerhard (2020). "Anwendungsgebiete für die automatisierte Informationsgewinnung aus Bildern". ger. In: Harrassowitz Verlag, Wiesbaden, pp. 85–99.

Cooper, David, Turinsky, Andrei, Sensen, Christoph, and Hallgrimsson, Benedikt (2007). "Effect of voxel size on 3D micro-CT analysis of cortical bone porosity". In: *Calcified tissue international* 80.3, pp. 211–219.

Crombie, Donald L (1963). "Diagnostic process". In: *The Journal of the College of General Practitioners* 6.4, p. 579.

Dalenogare, Lucas Santos, Benitez, Guilherme Brittes, Ayala, Néstor Fabián, and Frank, Alejandro Germán (2018). "The expected contribution of Industry 4.0 technologies for industrial performance". In: *International Journal of production economics* 204, pp. 383–394.

Davenport, Thomas and Kalakota, Ravi (2019). "The potential for artificial intelligence in healthcare". In: *Future healthcare journal* 6.2, p. 94.

Davies, E Roy (2012). *Computer and machine vision: theory, algorithms, practicalities*. Academic Press.

De Mauro, Andrea, Greco, Marco, and Grimaldi, Michele (2016). "A formal definition of Big Data based on its essential features". In: *Library Review* 3, pp. 122–135.

Donabedian, Avedis (1966). "Evaluating the quality of medical care". In: *The Milbank memorial fund quarterly* 44.3, pp. 166–206.

Donaldson, Molla S, Corrigan, Janet M, Kohn, Linda T, et al. (2000). *To err is human: building a safer health system*. National Academies Press. ISBN: 0309068371.

Dong, Liju and Yu, Ge (2004). "An optimization-based approach to image binarization". In: *The Fourth International Conference onComputer and Information Technology, 2004. CIT'04.* IEEE, pp. 165–170.

Dube, Simant (2021). *An Intuitive Exploration of Artificial Intelligence: Theory and Applications of Deep Learning*. Springer Nature. ISBN: 9783030686239.

Dumoulin, Vincent and Visin, Francesco (2016). "A guide to convolution arithmetic for deep learning". In: *arXiv:1603.07285*. eprint: `1603.07285`.

Everingham, Mark, Van Gool, Luc, Williams, Christopher KI, Winn, John, and Zisserman, Andrew (2010a). "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88.2, pp. 303–338.

Everingham, Mark, Van Gool, Luc, Williams, Christopher KI, Winn, John, and Zisserman, Andrew (2010b). "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88.2, pp. 303–338.

Everingham, Mark, Van Gool, Luc, Williams, Christopher KI, Winn, John, and Zisserman, Andrew (2010c). "The pascal visual object classes (voc) challenge". In: *International journal of computer vision* 88.2, pp. 303–338.

Fabricius, Katharina E and De'Ath, Glenn (2004). "Identifying ecological change and its causes: a case study on coral reefs". In: *Ecological Applications* 14.5, pp. 1448–1465.

Farahnakian, Fahimeh, Haghbayan, Mohammad-Hashem, Poikonen, Jonne, Laurinen, Markus, Nevalainen, Paavo, and Heikkonen, Jukka (2018). "Object detection based on multi-sensor proposal fusion in maritime environment". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, pp. 971–976.

Feng, Xin, Jiang, Youni, Yang, Xuejiao, Du, Ming, and Li, Xin (2019). "Computer vision algorithms and hardware implementations: A survey". In: *Integration* 69, pp. 309–320.

Ferrari, Vittorio, Fevrier, Loic, Jurie, Frederic, and Schmid, Cordelia (2007). "Groups of adjacent contour segments for object detection". In: *IEEE transactions on pattern analysis and machine intelligence* 30.1, pp. 36–51.

Ferri, César, Hernández-Orallo, José, and Modroiu, R (2009). "An experimental comparison of performance measures for classification". In: *Pattern Recognition Letters* 30.1, pp. 27–38.

Fisher, Robert B, Breckon, Toby P, Dawson-Howe, Kenneth, Fitzgibbon, Andrew, Robertson, Craig, Trucco, Emanuele, and Williams, Christopher KI (2013). *Dictionary of computer vision and image processing*. John Wiley & Sons. ISBN: 1119941865.

Forsyth, David and Ponce, Jean (2011). *Computer vision: A modern approach*. Prentice hall.

Furrer, Frank J (2019). "Future-Proof Software-Systems". In: *Future-Proof Software-Systems*. Springer, pp. 45–55.

Galceran, Enric, Djapic, Vladimir, Carreras, Marc, and Williams, David P (2012). "A real-time underwater object detection algorithm for multi-beam forward looking sonar". In: *IFAC Proceedings Volumes* 45.5, pp. 306–311.

Garcia-Garcia, Alberto, Orts-Escolano, Sergio, Oprea, Sergiu, Villena-Martinez, Victor, Martinez-Gonzalez, Pablo, and Garcia-Rodriguez, Jose (2018). "A survey on deep learning techniques for image and video semantic segmentation". In: *Applied Soft Computing* 70, pp. 41–65.

Géron, A. (2020). *Deep Learning avec Keras et TensorFlow: Mise en oeuvre et cas concrets*. Dunod. ISBN: 9782100805020. URL: https://books.google.de/books?id=45%5C_qDwAAQBAJ.

Gholamalinezhad, Hossein and Khosravi, Hossein (2020). "Pooling methods in deep neural networks, a review". In: *arXiv preprint arXiv:2009.07485*.

Girshick, Ross (2015). "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448.

Girshick, Ross, Donahue, Jeff, Darrell, Trevor, and Malik, Jitendra (2014). “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Glorot, Xavier, Bordes, Antoine, and Bengio, Yoshua (2011). “Deep sparse rectifier neural networks”. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, pp. 315–323.

Golmant, Noah, Vemuri, Nikita, Yao, Zhewei, Feinberg, Vladimir, Gholami, Amir, Rothauge, Kai, Mahoney, Michael W, and Gonzalez, Joseph (2018). “On the computational inefficiency of large batch sizes for stochastic gradient descent”. In: *arXiv preprint arXiv:1811.12941*.

Gonzalez, RC and Woods, RE (2008). *Digital Image Processing 3rd Edition 461–520*.

Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron (2016). *Deep learning*. MIT press.

Guo, Wei, Yang, Wen, Zhang, Haijian, and Hua, Guang (2018). “Geospatial object detection in high resolution satellite images based on multi-scale convolutional neural network”. In: *Remote Sensing* 10.1, p. 131.

Guo, Yanming, Liu, Yu, Georgiou, Theodoros, and Lew, Michael S (2018). “A review of semantic segmentation using deep neural networks”. In: *International journal of multimedia information retrieval* 7.2, pp. 87–93.

Guo, Yanming, Liu, Yu, Oerlemans, Ard, Lao, Songyang, Wu, Song, and Lew, Michael S (2016). “Deep learning for visual understanding: A review”. In: *Neurocomputing* 187, pp. 27–48.

Haussmann, Elmar, Fenzi, Michele, Chitta, Kashyap, Ivanecky, Jan, Xu, Hanson, Roy, Donna, Mittel, Akshita, Koumchatzky, Nicolas, Farabet, Clement, and Alvarez, Jose M. (2020). “Scalable Active Learning for Object Detection”. In: *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1430–1435. DOI: 10.1109/IV47402.2020.9304793.

He, Fengxiang, Liu, Tongliang, and Tao, Dacheng (2019). “Control batch size and learning rate to generalize well: Theoretical and empirical evidence”. In: *Advances in Neural Information Processing Systems* 32.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian (2016). “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Helbing, Dirk (2019). “Societal, economic, ethical and legal challenges of the digital revolution: from big data to deep learning, artificial intelligence, and manipulative technologies”. In: *Towards digital enlightenment*. Springer, pp. 47–72.

Hounsfield, Godfrey N (1973). “Computerized transverse axial scanning (tomography): Part 1. Description of system”. In: *The British journal of radiology* 46.552, pp. 1016–1022.

Huang, Gao, Sun, Yu, Liu, Zhuang, Sedra, Daniel, and Weinberger, Kilian Q (2016). “Deep networks with stochastic depth”. In: *European conference on computer vision*. Springer, pp. 646–661.

Ioffe, Sergey and Szegedy, Christian (2015). “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International conference on machine learning*. PMLR, pp. 448–456.

Jack, K. (2011). *Video Demystified: A Handbook for the Digital Engineer*. Video Demystified Series. Elsevier Science. ISBN: 9780080553955. URL: `https://books.google.de/books?id=6dgWB3-rChYC`.

Jaisakthi, SM, Mirunalini, Palaniappan, and Aravindan, Chandrabose (2019). "Coral Reef Annotation and Localization using Faster R-CNN." In: *CLEF (Working Notes)*.

Jiao, Licheng, Zhang, Fan, Liu, Fang, Yang, Shuyuan, Li, Lingling, Feng, Zhixi, and Qu, Rong (2019a). "A survey of deep learning-based object detection". In: *IEEE access* 7, pp. 128837–128868.

Jiao, Licheng, Zhang, Fan, Liu, Fang, Yang, Shuyuan, Li, Lingling, Feng, Zhixi, and Qu, Rong (2019b). "A survey of deep learning-based object detection". In: *IEEE access* 7, pp. 128837–128868.

Jin, Jonghoon, Dundar, Aysegul, and Culurciello, Eugenio (2014). "Flattened convolutional neural networks for feedforward acceleration". In: *arXiv preprint arXiv:1412.5474*.

Joblove, George H and Greenberg, Donald (1978). "Color spaces for computer graphics". In: *Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, pp. 20–25.

Kanan, Christopher and Cottrell, Garrison W (2012). "Color-to-grayscale: does the method matter in image recognition?" In: *PloS one* 7.1, e29740.

Kandel, Ibrahem and Castelli, Mauro (2020). "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset". In: *ICT express* 6.4, pp. 312–315.

Kao, Chieh-Chi, Lee, Teng-Yok, Sen, Pradeep, and Liu, Ming-Yu (2018). "Localization-aware active learning for object detection". In: *Asian Conference on Computer Vision*. Springer, pp. 506–522.

Kerlin, Felix, Bogomasov, Kirill, and Conrad, Stefan (2022). "Monitoring Coral Reefs Using Faster R-CNN". In: *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 04-08*. CEUR-WS.org.

Khirirat, Sarit, Feyzmahdavian, Hamid Reza, and Johansson, Mikael (2017). "Mini-batch gradient descent: Faster convergence under data sparsity". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, pp. 2880–2887.

Kiefer, Jack and Wolfowitz, Jacob (1952). "Stochastic estimation of the maximum of a regression function". In: *The Annals of Mathematical Statistics*, pp. 462–466.

Kim, Byeongjin and Yu, Son-Cheol (2017). "Imaging sonar based real-time underwater object detection utilizing AdaBoost method". In: *2017 IEEE Underwater Technology (UT)*. IEEE, pp. 1–5.

Koff, David A and Shulman, Harry (2006). "An overview of digital compression of medical images: can we use lossy image compression in radiology?" In: *Canadian association of radiologists journal* 57.4, p. 211.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E (2012). "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems* 25.

Larobina, Michele and Murino, Loredana (2014). "Medical image file formats". In: *Journal of digital imaging* 27.2, pp. 200–206.

Lasi, Heiner, Fettke, Peter, Kemper, Hans-Georg, Feld, Thomas, and Hoffmann, Michael (2014). "Industry 4.0". In: *Business & information systems engineering* 6.4, pp. 239–242.

Lateef, Fahad and Ruichek, Yassine (2019). "Survey on semantic segmentation using deep learning techniques". In: *Neurocomputing* 338, pp. 321–348.

LeCun, Yann, Boser, Bernhard, Denker, John S, Henderson, Donnie, Howard, Richard E, Hubbard, Wayne, and Jackel, Lawrence D (1989). "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4, pp. 541–551.

Li, Xiangrui, Morgan, Paul S, Ashburner, John, Smith, Jolinda, and Rorden, Christopher (2016). "The first step for neuroimaging data analysis: DICOM to NIfTI conversion". In: *Journal of neuroscience methods* 264, pp. 47–56.

Li, Yuxi (2017). "Deep reinforcement learning: An overview". In: *arXiv preprint arXiv:1701.07274*.

Lin, Min, Chen, Qiang, and Yan, Shuicheng (2013). "Network in network". In: *arXiv preprint arXiv:1312.4400*.

Lin, Tsung-Yi, Dollár, Piotr, Girshick, Ross, He, Kaiming, Hariharan, Bharath, and Belongie, Serge (2017a). "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125.

Lin, Tsung-Yi, Goyal, Priya, Girshick, Ross, He, Kaiming, and Dollár, Piotr (2017b). "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988.

Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence (2014a). "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer, pp. 740–755.

Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence (2014b). "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer, pp. 740–755.

Liu, Li, Ouyang, Wanli, Wang, Xiaogang, Fieguth, Paul, Chen, Jie, Liu, Xinwang, and Pietikäinen, Matti (2020). "Deep learning for generic object detection: A survey". In: *International journal of computer vision* 128.2, pp. 261–318.

Loussaief, Sehla and Abdelkrim, Afef (2018). "Convolutional neural network hyper-parameters optimization based on genetic algorithms". In: *International Journal of Advanced Computer Science and Applications* 9.10, pp. 252–266.

Lowbridge, Chris and Ralph, Anna P (2020). "Tuberculosis: yesterday, today and tomorrow". In: *Microbiology Australia* 41.4, pp. 192–195.

Maas, Andrew L, Hannun, Awni Y, Ng, Andrew Y, et al. (2013). "Rectifier nonlinearities improve neural network acoustic models". In: *Proc. icml*. Vol. 30. 1. Citeseer, p. 3.

Makantasis, Konstantinos, Doulamis, Anastasios, and Doulamis, Nikolaos (2013). "Vision-based maritime surveillance system using fused visual attention maps and online adaptable tracker". In: *2013 14th international workshop on image analysis for multimedia interactive services (WIAMIS)*. IEEE, pp. 1–4.

Mandhouj, Imen, Amiri, Hamid, Maussang, Frederic, Solaiman, Basel, et al. (2012). "Sonar image processing for underwater object detection based on high resolution system". In: *Proc. Second Workshop on Signal and Document Processing (SIDOP 2012)*. Citeseer, pp. 5–10.

Mariano, Vladimir Y, Min, Junghye, Park, Jin-Hyeong, Kasturi, Rangachar, Mihalcik, David, Li, Huiping, Doermann, David, and Drayer, Thomas (2002). "Performance evaluation of object detection algorithms". In: *Object recognition supported by user interaction for service robots*. Vol. 3. IEEE, pp. 965–969.

Maxwell, Robert J (1992). "Dimensions of quality revisited: from thought to action." In: *Quality in health care* 1.3, p. 171.

Miano, John (1999). *Compressed image file formats: Jpeg, png, gif, xbm, bmp*. Addison-Wesley Professional.

Mildenberger, Peter, Eichelberg, Marco, and Martin, Eric (2002). "Introduction to the DICOM standard". In: *European radiology* 12.4, pp. 920–927.

Min, Erxue, Guo, Xifeng, Liu, Qiang, Zhang, Gen, Cui, Jianjing, and Long, Jun (2018). "A survey of clustering with deep learning: From the perspective of network architecture". In: *IEEE Access* 6, pp. 39501–39514.

MINC File Format (2022). `https://www.mcgill.ca/bic/software/minc`. Accessed: 2022-04-07.

Moallem, Payman and Razmjooy, Navid (2012). "Optimal threshold computing in automatic image thresholding using adaptive particle swarm optimization". In: *Journal of applied research and technology* 10.5, pp. 703–712.

Mohammed, M., Khan, M.B., and Bashier, E.B.M. (2016). *Machine Learning: Algorithms and Applications*. CRC Press. ISBN: 9781498705394. URL: `https://books.google.de/books?id=X8LBDAAAQBAJ`.

Munro, Robert (2020). "Human-in-the-loop machine learning". In: *Sl: O'REILLY MEDIA*.

Nendaz, Mathieu and Perrier, Arnaud (2012). "Diagnostic errors and flaws in clinical reasoning: mechanisms and prevention in practice". In: *Swiss medical weekly* 43.

Ng, Andrew (2017). *Why AI Is the New Electricity*. `https://www.gsb.stanford.edu/insights/andrew-ng-why-ai-new-electricity`. Accessed: 2021-10-10.

NIfTI File Format (2022). `https://nifti.nimh.nih.gov/nifti-1`. Accessed: 2022-04-07.

Nurdin, Nurjannah, Komatsu, Teruhisa, Akbar, AS, Djalil, Abdul Rasyid, Amri, Khairul, et al. (2015). "Multisensor and multitemporal data from Landsat images to detect damage to coral reefs, small islands in the Spermonde archipelago, Indonesia". In: *Ocean Science Journal* 50.2, pp. 317–325.

Nwankpa, Chigozie, Ijomah, Winifred, Gachagan, Anthony, and Marshall, Stephen (2018). "Activation functions: Comparison of trends in practice and research for deep learning". In: *arXiv preprint arXiv:1811.03378*.

O'Leary, Dennis S and O'Leary, Margaret R (1992). "From quality assurance to quality improvement: the Joint Commission on Accreditation of Healthcare Organizations and emergency care". In: *Emergency medicine clinics of North America* 10.3, pp. 477–492.

Oakes, Guy (1990). "The sales process and the paradoxes of trust". In: *Journal of Business Ethics* 9.8, pp. 671–679.

Ojala, Timo, Pietikainen, Matti, and Maenpaa, Topi (2002). "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns". In: *IEEE Transactions on pattern analysis and machine intelligence* 24.7, pp. 971–987.

Otsu, Nobuyuki (1979). "A threshold selection method from gray-level histograms". In: *IEEE transactions on systems, man, and cybernetics* 9.1, pp. 62–66.

Otter, Daniel W, Medina, Julian R, and Kalita, Jugal K (2020). “A survey of the usages of deep learning for natural language processing”. In: *IEEE transactions on neural networks and learning systems* 32.2, pp. 604–624.

Parker, Jim R (2010). *Algorithms for image processing and computer vision.* John Wiley & Sons. ISBN: 9781118021880.

Pitas, Ioannis (2000). *Digital image processing algorithms and applications.* John Wiley & Sons. ISBN: 0471377392.

Process definition, Cambridge (2022). `https://dictionary.cambridge.org/de/worterbuch/englisch/process`. Accessed: 2022-04-29.

QA Definition (2022). `https://searchsoftwarequality.techtarget.com/definition/quality-assurance`. Accessed: 2022-04-29.

Qian, Ning (1999). “On the momentum term in gradient descent learning algorithms”. In: *Neural networks* 12.1, pp. 145–151.

Rabbani, Majid and Joshi, Rajan (2002). “An overview of the JPEG 2000 still image compression standard”. In: *Signal processing: Image communication* 17.1, pp. 3–48.

Ranzato, Marc’Aurelio, Huang, Fu Jie, Boureau, Y-Lan, and LeCun, Yann (2007). “Unsupervised learning of invariant feature hierarchies with applications to object recognition”. In: *2007 IEEE conference on computer vision and pattern recognition.* IEEE, pp. 1–8.

Rawat, Waseem and Wang, Zenghui (2017). “Deep convolutional neural networks for image classification: A comprehensive review”. In: *Neural computation* 29.9, pp. 2352–2449.

Redmon, Joseph, Divvala, Santosh, Girshick, Ross, and Farhadi, Ali (2016). “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Ren, Shaoqing, He, Kaiming, Girshick, Ross, and Sun, Jian (2015). “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems* 28.

Rizzini, Dario Lodi, Kallasi, Fabjan, Oleari, Fabio, and Caselli, Stefano (2015). “Investigation of vision-based underwater object detection with multiple datasets”. In: *International Journal of Advanced Robotic Systems* 12.6, p. 77.

Robb, Richard Arlin, Hanson, Dennis P, Karwoski, RA, Larson, AG, Workman, EL, and Stacy, MC (1989). “Analyze: a comprehensive, operator-interactive software package for multidimensional medical image display and analysis”. In: *Computerized Medical Imaging and Graphics* 13.6, pp. 433–454.

Robbins, Herbert and Monro, Sutton (1951). “A stochastic approximation method”. In: *The annals of mathematical statistics*, pp. 400–407.

Rosenblatt, Frank (1958). “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6, p. 386.

Ruder, Sebastian (2016). “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747.*

Rumelhart, David E, Hinton, Geoffrey E, and Williams, Ronald J (1986). “Learning representations by back-propagating errors”. In: *nature* 323.6088, pp. 533–536.

Schulz-Mirbach, Hanns (1995). “Invariant features for gray scale images”. In: *Mustererkennung 1995.* Springer, pp. 1–14.

Schyve, Paul M and Prevost, James A (1990). “From quality assurance to quality improvement”. In: *Psychiatric Clinics of North America* 13.1, pp. 61–71.

Ševo, Igor and Avramović, Aleksej (2016). "Convolutional neural network based automatic object detection on aerial images". In: *IEEE geoscience and remote sensing letters* 13.5, pp. 740–744.

Shafiq-ul-Hassan, Muhammad, Zhang, Geoffrey G, Latifi, Kujtim, Ullah, Ghanim, Hunt, Dylan C, Balagurunathan, Yoganand, Abdalah, Mahmoud Abrahem, Schabath, Matthew B, Goldgof, Dmitry G, Mackin, Dennis, et al. (2017). "Intrinsic dependencies of CT radiomic features on voxel size and number of gray levels". In: *Medical physics* 44.3, pp. 1050–1062.

Shahian, David M, Normand, Sharon-Lise T, Friedberg, Mark W, Hutter, Matthew M, and Pronovost, Peter J (2016). "Rating the raters: the inconsistent quality of health care performance measurement". In: *Annals of surgery* 264.1, pp. 36–38.

Shewfelt, Robert L (1999). "What is quality?" In: *Postharvest biology and technology* 15.3, pp. 197–200.

Sidlauskas, Darius, Chester, Sean, Zacharatou, Eleni Tzirita, and Ailamaki, Anastasia (2018). "Improving spatial data processing by clipping minimum bounding boxes". In: *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, pp. 425–436.

Simonyan, Karen and Zisserman, Andrew (2014). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.

Smith, Samuel L, Kindermans, Pieter-Jan, Ying, Chris, and Le, Quoc V (2017). "Don't decay the learning rate, increase the batch size". In: *arXiv preprint arXiv:1711.00489*.

Spalding, Mark, Spalding, Mark D, Ravilious, Corinna, Green, Edmund Peter, et al. (2001). *World atlas of coral reefs*. Univ of California Press.

Steffens, Aljoscha, Campello, Antonio, Ravenscroft, James, Clark, Adrian, and Hagras, Hani (2019). "Deep Segmentation: using Deep Convolutional Networks for Coral Reef pixel-wise Parsing." In: *CLEF (Working Notes)*.

Suárez, Isabelle, Fünger, Sarah Maria, Kröger, Stefan, Rademacher, Jessica, Fätkenheuer, Gerd, and Rybniker, Jan (2019). "The Diagnosis and Treatment of Tuberculosis". In: *Deutsches Aerzteblatt International* 116.43.

Sutton, Richard S and Barto, Andrew G (2018). *Reinforcement learning: An introduction*. MIT press.

Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, and Rabinovich, Andrew (2015). "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9.

Tan, Mingxing, Pang, Ruoming, and Le, Quoc V (2020). "Efficientdet: Scalable and efficient object detection". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10781–10790.

Tan, Pang-Ning, Steinbach, Michael, and Kumar, Vipin (2016). *Introduction to data mining*. Pearson Education India.

Tayara, Hilal and Chong, Kil To (2018). "Object detection in very high-resolution aerial images using one-stage densely connected feature pyramid network". In: *Sensors* 18.10, p. 3341.

Technology Usage Statistics (2022). `https://w3techs.com/technologies`. Accessed: 2022-04-07.

Tuytelaars, Tinne and Mikolajczyk, Krystian (2008). *Local invariant feature detectors: a survey*. Now Publishers Inc.

Uijlings, Jasper RR, Van De Sande, Koen EA, Gevers, Theo, and Smeulders, Arnold WM (2013). “Selective search for object recognition”. In: *International journal of computer vision* 104.2, pp. 154–171.

Van Den Bos, Jill, Rustagi, Karan, Gray, Travis, Halford, Michael, Ziemkiewicz, Eva, and Shreve, Jonathan (2011). “The $17.1 billion problem: the annual cost of measurable medical errors”. In: *Health Affairs* 30.4, pp. 596–603.

Voulodimos, Athanasios, Doulamis, Nikolaos, Doulamis, Anastasios, and Protopapadakis, Eftychios (2018). “Deep learning for computer vision: A brief review”. In: *Computational intelligence and neuroscience* 2018.

Wang, Nan, Zheng, Bing, Zheng, Haiyong, and Yu, Zhibin (2017). “Feeble object detection of underwater images through LSR with delay loop”. In: *Optics express* 25.19, pp. 22490–22498.

Wang, Tong, Zhu, Yousong, Zhao, Chaoyang, Zeng, Wei, Wang, Yaowei, Wang, Jinqiao, and Tang, Ming (2020). “Large Batch Optimization for Object Detection: Training COCO in 12 minutes”. In: *European Conference on Computer Vision*. Springer, pp. 481–496.

Wang, Xiaoyu, Yang, Ming, Zhu, Shenghuo, and Lin, Yuanqing (2013). “Regionlets for generic object detection”. In: *Proceedings of the IEEE international conference on computer vision*, pp. 17–24.

Weiss, Karl, Khoshgoftaar, Taghi M, and Wang, DingDing (2016). “A survey of transfer learning”. In: *Journal of Big data* 3.1, pp. 1–40.

Whitcher, Brandon, Schmid, Volker J, and Thorton, Andrew (2011). “Working with the DICOM and NIfTI Data Standards in R”. In: *Journal of Statistical Software* 44, pp. 1–29.

Wiggins, Richard H, Davidson, H Christian, Harnsberger, H Ric, Lauman, Jason R, and Goede, Patricia A (2001). “Image file formats: past, present, and future”. In: *Radiographics* 21.3, pp. 789–798.

Williams, David P (2011). “On adaptive underwater object detection”. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 4741–4748.

Xu, Bing, Wang, Naiyan, Chen, Tianqi, and Li, Mu (2015). “Empirical evaluation of rectified activations in convolutional network”. In: *arXiv preprint arXiv:1505.00853*.

Yu, Fisher and Koltun, Vladlen (2015). “Multi-scale context aggregation by dilated convolutions”. In: *arXiv preprint arXiv:1511.07122*.

Yu, Kun-Hsing, Beam, Andrew L, and Kohane, Isaac S (2018). “Artificial intelligence in healthcare”. In: *Nature biomedical engineering* 2.10, pp. 719–731.

Yuan, Tianning, Wan, Fang, Fu, Mengying, Liu, Jianzhuang, Xu, Songcen, Ji, Xiangyang, and Ye, Qixiang (2021). “Multiple instance active learning for object detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5330–5339.

Zeiler, Matthew D (2012). “Adadelta: an adaptive learning rate method”. In: *arXiv preprint arXiv:1212.5701*.

Zhang, Guodong, Li, Lala, Nado, Zachary, Martens, James, Sachdeva, Sushant, Dahl, George, Shallue, Chris, and Grosse, Roger B (2019). “Which algorithmic choices matter at which batch sizes? insights from a noisy quadratic model”. In: *Advances in neural information processing systems* 32.

Zhang, Xiangyu, Zhou, Xinyu, Lin, Mengxiao, and Sun, Jian (2018). “Shufflenet: An extremely efficient convolutional neural network for mobile devices”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856.

Zhang, Xin, Han, Liangxiu, Han, Lianghao, and Zhu, Liang (2020). “How well do deep learning-based methods for land cover classification and object detection perform on high resolution remote sensing imagery?” In: *Remote Sensing* 12.3, p. 417.

Zhang, Yuanlin, Yuan, Yuan, Feng, Yachuang, and Lu, Xiaoqiang (2019). “Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection”. In: *IEEE Transactions on Geoscience and Remote Sensing* 57.8, pp. 5535–5548.

Zheng, Ce, Wu, Wenhan, Yang, Taojiannan, Zhu, Sijie, Chen, Chen, Liu, Ruixu, Shen, Ju, Kehtarnavaz, Nasser, and Shah, Mubarak (2020). “Deep learning-based human pose estimation: A survey”. In: *arXiv preprint arXiv:2012.13392*.

Zhou, Zhenjin, Ma, Lei, Fu, Tengyu, Zhang, Ge, Yao, Mengru, and Li, Manchun (2018). “Change detection in coral reef environment using high-resolution images: comparison of object-based and pixel-based paradigms”. In: *ISPRS International Journal of Geo-Information* 7.11, p. 441.

Zhu, Yafei, Chang, Lin, Dai, Jialun, Zheng, Haiyong, and Zheng, Bing (2016). “Automatic object detection and segmentation from underwater images via saliency-based region merging”. In: *OCEANS 2016-Shanghai*. IEEE, pp. 1–4.

Zou, Zhengxia, Shi, Zhenwei, Guo, Yuhong, and Ye, Jieping (2019a). *Object Detection in 20 Years: A Survey*. arXiv: `1905.05055 [cs.CV]`.

Zou, Zhengxia, Shi, Zhenwei, Guo, Yuhong, and Ye, Jieping (2019b). “Object detection in 20 years: A survey”. In: *arXiv preprint arXiv:1905.05055*.

List of Figures

List of Tables

A

Publications

A.1 Feature-Based Approach for Severity Scoring of Lung Tuberculosis from CT Images

Bogomasov, Kirill, Himmelspach, Ludmila, Klassen, Gerhard, Tatusch, Martha, and Conrad, Stefan (2018). “Feature-Based Approach for Severity Scoring of Lung Tuberculosis from CT Images.” In: *Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14.* CEUR-WS.org.
Kirill Bogomasov contributed with the feature engineering and the implementation of features for Lung Calcification, Lung Wateriness as well as features based on the Hounsfield Histograms. Pulmonary Cavities and Infection Ratio was contributed by Ludmila Himmelspach. Lung shape comparison was contributed by Martha Tatusch and Gerhard Klassen. The regression and classification methods were contributed by Ludmila Himmelspach. The preparation of the manuscript was done accordingly to the contribution by all authors in a variable extent under the supervision of Stefan Conrad.

Status: Published.

A.2 Feature and Deep Learning Based Approaches for Automatic Report Generation and Severity Scoring of Lung Tuberculosis from CT Images

Bogomasov, Kirill, Braun, Daniel, Burbach, Andreas, Himmelspach, Ludmila, and Conrad, Stefan (2019a). “Feature and Deep Learning Based Approaches for Automatic Report Generation and Severity Scoring of Lung Tuberculosis from CT Images.” In: *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 09-12.* CEUR-WS.org.
Kirill Bogomasov and Daniel Braun contributed equally with the Deep Learning architecture including pre- and post-processing, as well as the decision rule. Ludmila Himmelspach and Andreas Burbach contributed with the feature-based approach. The preparation of the manuscript was done accordingly to the contribution by all authors in a variable extent under the supervision of Stefan Conrad.

Status: Published.

A.3 Orientation Estimation in MRI of Prostate Cancer Patients: When Simple Models Perform Better

Bogomasov, Kirill, Thomas, Grings, Rubbert, Christian, Schimmöller, Lars, and Stefan, Conrad (2021). “Orientation Estimation in MRI of Prostate Cancer Patients: When Simple Models Perform Better”. In: *2021 IEEE International Conference on Imaging Systems and Techniques (IST).* IEEE, pp. 1–6.

The ideas and methods for research were contributed by Kirill Bogomasov. The implementation was done mostly by Thomas Grings. The preparation of the manuscript was done entirely by Kirill Bogomasov under the supervision of Stefan Conrad, Christian Rubbert and Lars Schimmoeller.

Status: Published.

A.4 A two-staged Approach for Localization and Classification of Coral Reef Structures and Compositions

Bogomasov, Kirill, Grawe, Philipp, and Conrad, Stefan (2019b). "A two-staged Approach for Localization and Classification of Coral Reef Structures and Compositions." In: *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 09-12.* CEUR-WS.org.
The research was done by Kirill Bogomasov with support of Philipp Grawe. While Kirill Bogomasov provided the methology, Philipp Grawe focused on preparing the data. The preparation of the manuscript was done entirely by Kirill Bogomasov under the supervision of Stefan Conrad.

Status: Published.

A.5 Enhanced Localization and Classification of Coral Reef Structures and Compositions

Bogomasov, Kirill, Grawe, Philipp, and Conrad, Stefan (2020). "Enhanced Localization and Classification of Coral Reef Structures and Compositions". In: *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25.* CEUR-WS.org.
The research was done by Kirill Bogomasov with the support of Philipp Grawe. While Kirill Bogomasov provided the methodology, Philipp Grawe focused on preparing the data. The preparation of the manuscript was done entirely by Kirill Bogomasov under the supervision of Stefan Conrad.

Status: Published.

A.6 Efficient Fruit and Vegetable Classification and Counting for Retail Applications Using Deep Learning

Bogomasov, Kirill and Conrad, Stefan (2021). "Efficient Fruit and Vegetable Classification and Counting for Retail Applications Using Deep Learning". In: *Proceedings of the 5th International Conference on Advances in Artificial Intelligence.*

The research and the preparation of the manuscript were done entirely by Kirill Bogomasov under the supervision of Stefan Conrad.

Status: Published.

A.7 FSC-FloodFill Sky Classification

Bogomasov, Kirill (2016). "FSC-FloodFill Sky Classification". In: *Proceedings of the 28th GI-Workshop Grundlagen von Datenbanken, Nörten Hardenberg, Germany, May 24-27*, pp. 27–32. URL: `http://ceur-ws.org/Vol-1594/paper6.pdf`.
The research and the preparation of the manuscript were done entirely by Kirill Bogomasov under the supervision of Stefan Conrad.

Status: Published.

A.8 Monitoring Coral Reefs Using Faster R-CNN

Kerlin, Felix, Bogomasov, Kirill, and Conrad, Stefan (2022). "Monitoring Coral Reefs Using Faster R-CNN". in: *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 04-08*. CEUR-WS.org.
The ideas and methods for research are contributed by Kirill Bogomasov and Felix Kerlin. The implementation is done by Felix Kerlin under the supervision of Kirill Bogomasov. The preparation of the manuscript was done by Kirill Bogomasov and Felix Kerlin under the supervision of Stefan Conrad.

Status: Published.

A.9 InfEval: Application for Object Detection Analysis

Bogomasov, Kirill, Geuer, Tim, and Conrad, Stefan (2023). "InfEval: Application for Object Detection Analysis". In: *European Conference on Information Retrieval*. Springer.
The ideas and methods for research are contributed by Kirill Bogomasov and Tim Geuer. The implementation is done by Tim Geuer under the supervision of Kirill Bogomasov. The preparation of the manuscript was done by Kirill Bogomasov and Tim Geuer under the supervision of Stefan Conrad.

Status: Accepted.

A.10 Bilddaten in den Digitalen Geisteswissenschaften

Conrad, Stefan, Tatusch, Martha, Bogomasov, Kirill, and Klassen, Gerhard (2020). "Anwendungsgebiete für die automatisierte Informationsgewinnung aus Bildern". ger.

In: Harrassowitz Verlag, Wiesbaden, pp. 85–99.
The research on sky classification was done entirely by Kirill Bogomasov. The preparation of the manuscript was done by Kirill Bogomasov, Martha Tatusch and Gerhard Klassen under the supervision of Stefan Conrad.

Status: Published.