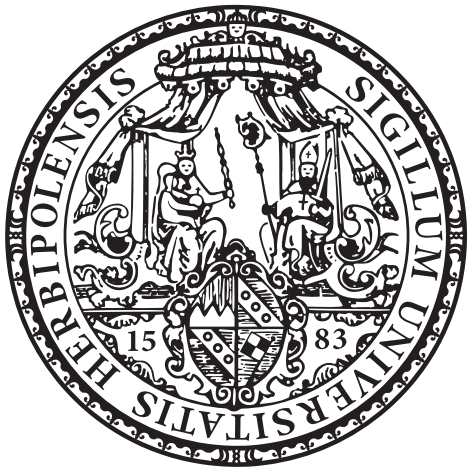
Julius-Maximilians-Universität Würzburg
Wirtschaftswissenschaftliche Fakultät



Applied Deep Learning: from Data to Deployment

Inauguraldissertation
zur Erlangung des akademischen Grades
Doctor rerum politicarum (Dr. rer. pol.)

vorgelegt von
Matthias Griebel, M.Sc.
geboren in Wilhelmshaven

Name und Anschrift:	Matthias Griebel Gertraud-Rostosky-Str. 54 97082 Würzburg
Erstgutachter:	Prof. Dr. Christoph M. Flath
Zweitgutachter:	Prof. Dr. Frédéric Thiesse
Datum der Einreichung:	19. Januar 2021

Abstract

Novel deep learning (DL) architectures, better data availability, and a significant increase in computing power have enabled scientists to solve problems that were considered unassailable for many years. A case in point is the "protein folding problem", a 50-year-old grand challenge in biology that the DL-system AlphaFold recently solved. Other examples comprise the development of large DL-based language models that, for instance, generate newspaper articles that hardly differ from those written by humans. However, developing unbiased, reliable, and accurate DL models remains a major challenge for various practical applications – and many promising DL projects never advance beyond the piloting stage. In light of these observations, this dissertation investigates the practical challenges encountered throughout the life cycle of DL projects and proposes solutions to develop and deploy rigorous DL models.

The first part of the dissertation is concerned with prototyping DL solutions in different domains. First, I conceptualize guidelines for applied image recognition and showcase their application in a biomedical research project. Next, I illustrate the bottom-up development of a DL backend for an augmented intelligence system in the manufacturing sector. Turning to the fashion domain, I present an artificial curation system for individual fashion outfit recommendations that leverages DL techniques and unstructured data from social media and fashion blogs. After that, I showcase how DL solutions can assist fashion designers in the creative process. Finally, I present my award-winning DL solution for the segmentation of glomeruli in human kidney tissue images that was developed for the Kaggle data science competition *HuBMAP – Hacking the Kidney*.

The second part continues the development path of the biomedical research project beyond the prototyping stage. Using data from five laboratories, I show that ground truth estimation from multiple human annotators and

training of DL model ensembles help to establish objectivity, reliability, and validity in DL-based bioimage analyses.

In the third part, I present *deepflash2*, a DL solution that addresses the typical challenges encountered during training, evaluation, and application of DL models in bioimaging. The tool facilitates the objective and reliable segmentation of ambiguous bioimages through multi-expert annotations and integrated quality assurance. It is embedded in an easy-to-use graphical user interface and offers best-in-class predictive performance for semantic and instance segmentation under economical usage of computational resources.

Kurzzusammenfassung

Die Entwicklung neuer Deep Learning (DL) Architekturen, flankiert durch eine bessere Datenverfügbarkeit und eine enorme Steigerung der Rechenleistung, ermöglicht Wissenschaftler:innen die Lösung von Problemen, die lange Zeit als unlösbar galten. Ein Paradebeispiel hierfür ist das 50 Jahre alte “Proteinfaltungsproblem” in der Biologie, das vor Kurzem duch das DL-System AlphaFold gelöst wurde. Andere Beispiele sind moderne, DL-basierte Sprachmodelle. Diese können unter anderem Zeitungsartikel verfassen, die nur schwer von Artikeln menschlicher Autoren:innen unterscheidbar sind. Die Entwicklung unvoreingenommener, zuverlässiger und präziser DL-Modelle für die praktische Anwendung bleibt jedoch eine große Herausforderung. Dies wird an zahlreichen vielversprechenden DL-Projekten sichtbar, die nicht über die Pilotphase herauskommen. Vor diesem Hintergrund untersuche ich in dieser Dissertation die Herausforderungen, die während des Lebenszyklus von DL-Projekten auftreten, und schlage Lösungen für die Entwicklung und den Einsatz verlässlicher DL-Modelle vor.

Der erste Teil der Arbeit befasst sich mit dem Prototyping von DL-Lösungen für verschiedene Anwendungsgebiete. Zunächst werden Richtlinien für die angewandte Bilderkennung konzipiert und deren Anwendung in einem biomedizinischen Forschungsprojekt gezeigt. Dem folgt die Darstellung einer Bottom-up-Entwicklung eines DL-Backends für ein Augmented-Intelligence-System im Fertigungssektor. Im Anschluss wird der Entwurf eines künstlichen Fashion-Curation-Systems für individuelle Outfit-Empfehlungen vorgestellt, das DL-Techniken und unstrukturierte Daten aus sozialen Medien und Modeblogs nutzt. Es folgt ein Abschnitt darüber, wie DL-Lösungen Modedesigner:innen im kreativen Prozess unterstützen können. Schließlich stelle ich meine prämierte DL-Lösung für die Segmentierung von Glomeruli in menschlichen Nierengewebe-Bildern vor, die für den Kaggle Data Science-Wettbewerb *HuBMAP - Hacking the Kidney* entwickelt wurde.

Im zweiten Teil wird der Entwicklungspfad des biomedizinischen Forschungsprojekts über das Prototyping-Stadium hinaus fortgesetzt. Anhand von Daten aus fünf Laboren wird gezeigt, dass die Schätzung einer *Ground-Truth* durch die Annotationen mehrerer Experten:innen und das Training von DL-Modell-Ensembles dazu beiträgt, Objektivität, Zuverlässigkeit und Validität in DL-basierten Analysen von Mikroskopie-Bildern zu manifestieren.

Im dritten Teil der Dissertation stelle ich die DL-Lösung *deepflash2* vor, welche die typischen Herausforderungen beim Training, der Evaluation und der Anwendung von DL-Modellen in der biologischen Bildgebung adressiert. Das Tool erleichtert die objektive und zuverlässige Segmentierung von mehrdeutigen Mikroskopie-Bildern durch die Integration von Annotationen mehrerer Experten:innen und integrierte Qualitätssicherung.

Contents

1 Introduction

The COVID-19 pandemic has become an unprecedented public health emergency. Beyond the spread of the disease itself, its far-reaching consequences affect society, economy, culture, ecology, politics, and science on a global scale. To mitigate the impact of the pandemic, thousands of scientists – virologists, epidemiologists, public health scholars, statisticians, and others – are working toward understanding the SARS-CoV-2 virus, the disease, and its evolution.

Fueled by the recent success of machine learning (ML) and deep learning (DL) methods in particular, the computer science community and associated organizations have rushed to contribute to the fight against the pandemic. The COVID-19 High-Performance Computing (HPC) Consortium[1], for instance, provides researchers access to the world's most powerful computing resources. It enables ML projects that aim to predict the spread of COVID-19 or analyze large numbers of chemical compounds to develop a therapy. The COVID Moonshot – a non-profit, open-science consortium – combines crowdsourcing with ML and robotic experiments to develop a globally accessible antiviral pill against COVID-19 (Chodera et al. 2020). Other initiatives involve the organization of ML competitions, e.g., to detect fake news related to COVID-19 (Patwa et al. 2021) or to identify informative COVID-19 tweets (Nguyen et al. 2020).

However, ML solutions do not always live up to their lofty promises. Wynants et al. (2020) analyze predictive ML models that are supposed to support medical decision making. This comprises, among other things, the detection of COVID-19 (based on medical imaging) and the prediction of mortality risk, progression to severe disease, or intensive care unit admission. Their review of 232 prediction models is a scathing verdict on the usefulness of such approaches, as most "proposed models are poorly reported and at high risk of bias such that their reported predictive performance is probably optimistic" (Wynants et al. 2020, p.1) and they "cannot recommend any model for use in

[1] https://covid19-hpc-consortium.org/

practice at this point" (Wynants et al. 2020, p.10). The majority of the reviewed models are diagnostic DL models trained on medical images (CT images or chest radiographs) to support the diagnosis of pneumonia. These models frequently lack external validation and are repeatedly criticized in subsequent studies (Roberts et al. 2021; DeGrave, Janizek, and Lee 2021).

While these findings may seem disillusioning, they are by no means specific to COVID-19 research. According to the 2020's McKinsey Global Survey on artificial intelligence (AI) by Balakrishnan et al. (2020), 50 percent of the responding companies report having adopted AI in at least one business function. However, a more detailed questioning revealed that only 16 percent had adopted DL models in a business function, as most projects got stuck in the piloting stage. In addition, the adoption of DL capabilities across different technology sectors is highly imbalanced, with 30 percent of DL adoption occurring in high-tech and telecom companies.

Conducting research at the intersection of computer science and domain sciences (Figure 1.1), information systems (IS) scholars can play a key role in bringing DL solutions into practice.

1.1 Research Focus

In the forthcoming editorial of the Management Information Systems Quarterly journal, Padmanabhan, Fang, and Sahoo (2022)[2] discuss the opportunities for ML research in IS and propose three different categories of contributions: (i) ML methods development, (ii) understanding phenomena using ML, and (iii) ML in complex systems.

The contributions in this thesis can best be attributed to the first category, ML methods development in IS. As a subset of ML, DL "allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction" (LeCun, Bengio, and Hinton 2015, p.436). Complementing this, computational design science research (DSR) in IS is "concerned with solving business and societal problems by developing computational models and algorithms" (Rai 2017, p.iii). A methodological contribution in the context of DL in IS includes the development of novel DL mod-

[2]Information taken from the MIS Quarterly Master class on ML in IS Research

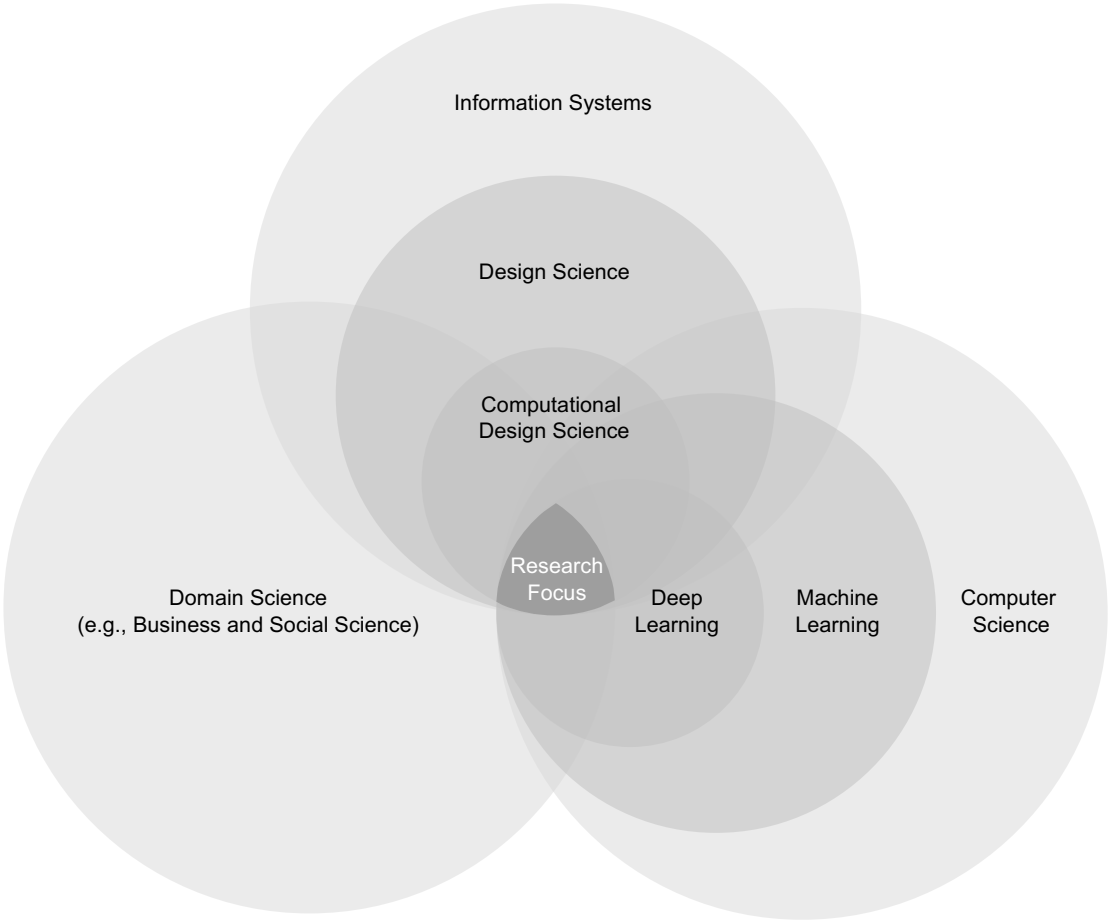


Figure 1.1: Research focus, inspired by Ullman (2021). Circles not to scale.

els and algorithms as well as the nontrivial extension of an existing DL method (Padmanabhan, Fang, and Sahoo 2022). That is, for instance, the development of a DL model for understanding medication nonadherence (Xie et al. 2017).

As illustrated in Figure 1.1, this thesis' research is located at the intersection of DL (computer science), computational DSR (IS), and different domain sciences. The guiding research objective can be summarized as:

> *Solving business and societal problems by developing rigorous deep learning models.*

To this end, I will follow a broad definition of business and societal problems, which may imply considerable overlap with other domains. For example, developing an objective and reliable DL-based model for diagnosing COVID-19 pneumonia may originate in the medical domain, however, it could eventually also solve a societal problem by mitigating the impacts of the pandemic.

This thesis combines the evidence given in several research articles published in outlets of different research domains.[3] While sections based on papers published in the IS domain explicitly follow a DSR approach (e.g., Sections 2.3 and 2.2), other parts that are based on papers published in other areas (e.g., life sciences in Chapter 3 and life science methods in Chapter 4) follow the methodology of the respective area (but may implicitly contain DSR elements).

From a technical perspective, the *Machine Learning Lifecycle* concept presents a suitable approach to establish a common denominator for all research presented in this thesis.

1.2 Machine Learning Life Cycle

Over the past 20 years, the Cross-Industry Standard Process for Data Mining (CRISP-DM, Chapman et al. 2000) has become a well-established methodology with solid industry support (Studer et al. 2021). CRISP-DM is a manifestation of best practices for conducting data mining and knowledge discovery. In addition, it is commonly applied in dedicated ML projects. To overcome hurdles typically encountered in ML projects, CRISP-DM has been extended several times to better address tasks such as model testing and monitoring (Breck et al. 2017), as well as quality assurance (Studer et al. 2021). Due to their iterative nature, these end-to-end ML workflows are frequently termed *ML life cycle* models.

Large organizations have also published their internal ML workflows. For instance, Microsoft has presented its machine learning workflow (Amershi et al. 2019) and the *Team Data Science Process lifecycle*[4]. Uber has introduced its Machine Learning Platform, *Michelangelo*, which implements an ML workflow to build, deploy, and operate machine learning solutions at scale (Hermann and Del Balso 2018). The workflow has been tried and tested in practice and is well suited to conceptually organize the status of ML or DL-based research projects. Thus, I have adapted the ML life cycle of Uber's *Michelangelo* for this thesis (Figure 1.2). Similar to the *Business Understanding* phase in CRISP-

[3] See Appendix A for a comprehensive list of articles.

[4] `https://docs.microsoft.com/bs-latn-ba/azure/architecture/data-science-process/lifecycle`

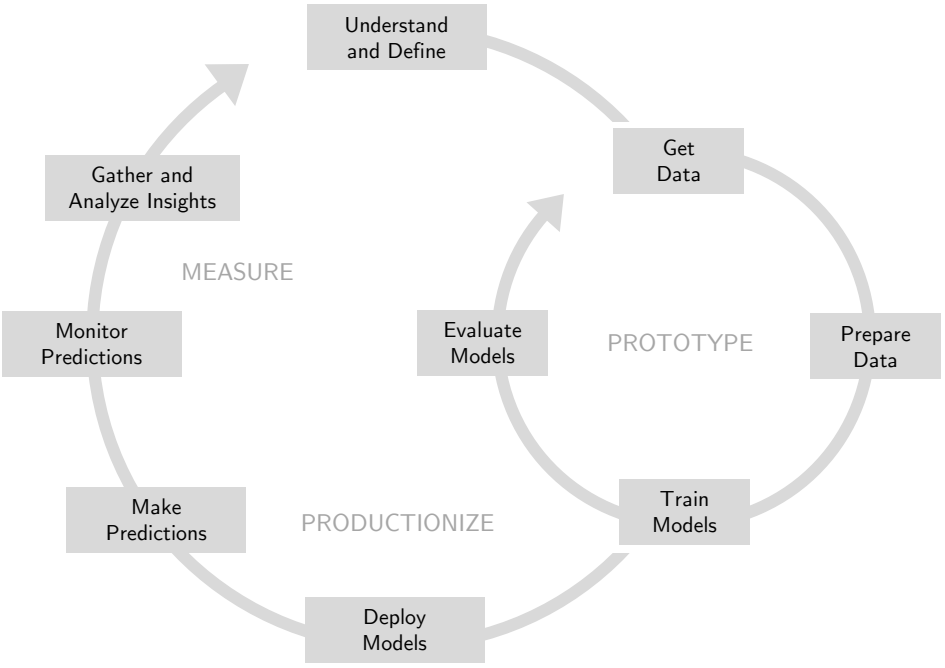


Figure 1.2: Machine learning life cycle adapted from Uber's *Michelangelo* (Hermann and Del Balso 2018).

DM, this ML life cycle starts with understanding the (business) needs and a definition of project requirements and objectives. The subsequent steps can be organized into three groups: prototyping a solution, productionizing the solution, and measuring the impact of the solution (Hermann and Del Balso 2018):

- **Prototype:** Prototyping a solution comprises most CRISP-DM phases. A prototyping cycle typically starts with *get data* (data collection), followed by *prepare data* to consolidate, clean, and transform the data. The *train models* step describes the *Modeling* phase when various modeling techniques are selected and applied. Finally, *evaluate models* covers the *Evaluation* phase to test the models for predictive accuracy, speed, robustness, or scalability.
- **Productionize:** Productionizing the solution means to *deploy models* and data pipelines into a staging or production environment. Once the models are successfully deployed, they are prepared to *make predictions* for new data (on demand).

- **Measure:** Measuring the impact of the solution requires to *monitor predictions* in the first place. This step is critical in real-world environments as models are typically trained and evaluated on a limited data set. Therefore, it is not guaranteed that they will work correctly for new data. The data collected during model monitoring should be *gathered and analyzed* to provide insights.

The three main concepts of the ML life cycle – Prototype, Productionize, Measure – form the backbone of this thesis. In this context, it should be noted that some of the presented steps in the ML life cycle (e.g., deploy models or monitor predictions) are typically attributed to the fields of ML engineering or MLOps. This thesis, however, emphasizes the theoretical and conceptual aspects of these steps.

1.3 Structure

This work addresses a variety of challenges that are encountered along the life cycle of DL models. Chapter 2 is primarily concerned with prototyping DL solutions for problems from different domains. In Section 2.1, I conceptualize guidelines for applied image recognition spanning task definition, deep neural network configuration, and training procedures. I showcase the guidelines by means of a biomedical research project for image recognition. Section 2.2 illustrates the bottom-up development of a DL backend for an augmented intelligence system in the manufacturing sector. A wearable device equipped with highly sensitive sensors is paired with a deep convolutional neural network to monitor connector system assembly processes in real-time. Turning to the fashion domain in Section 2.3, I present an artificial curation system for individual outfit recommendations that leverages DL techniques and unstructured data from social media and fashion blogs. Here, I lay out the artifact design and provide a comprehensive evaluation strategy to assess the system's utility. Section 2.4 provides a perspective on the possible role of AI in the creative sphere. Here, I explore how different DL algorithms can contribute to the creative process, and I specifically investigate the domain of fashion design to showcase how AI can assist designers. Finally, Section 2.5 presents my award-winning solution for the segmentation of glomeruli in human kidney tissue

images. The DL prototype was developed for the Kaggle data science competition *HuBMAP - Hacking the Kidney* hosted by the HuBMAP consortium.

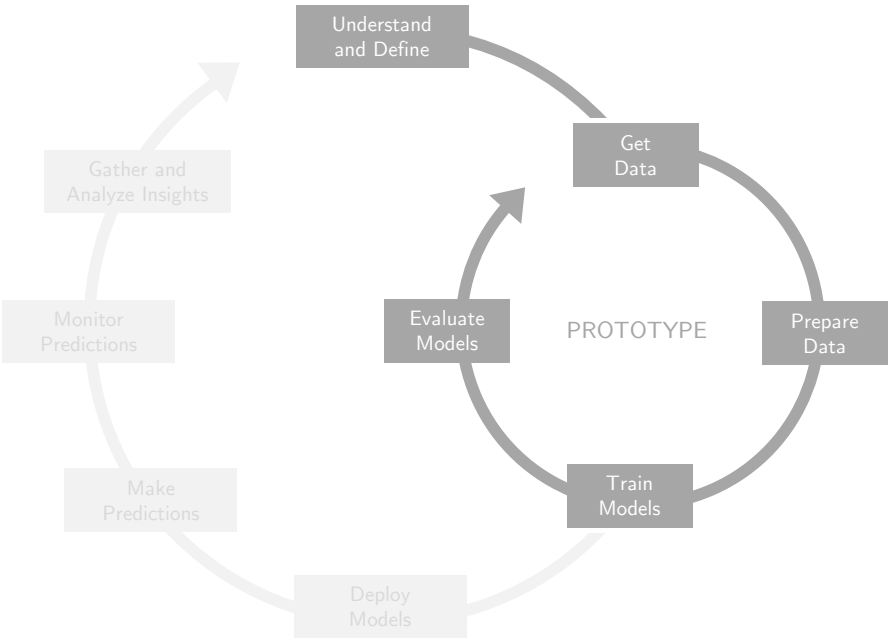
Chapter 3 continues the development path of the biomedical research project of Section 2.1 beyond the prototyping phase. Here, I investigate how different data annotation and training strategies affect the objectivity, reliability, and validity of DL-based bioimage analysis. I evaluate the results with data from two model organisms and five laboratories and provide guidelines for reproducible DL-based bioimage analyses.

Chapter 4 is concerned with all phases of the ML life cycle. Here, I present an easy-to-use DL tool that facilitates the objective and reliable segmentation of ambiguous bioimages through multi-expert annotations and integrated quality assurance (monitoring predictions and out-of-distribution detection). Thereby, the tool addresses typical challenges that may arise during training, evaluation, and application of DL models in bioimaging.

Lastly, Chapter 5 summarizes the findings and outlines future research directions for DL-based bioimage analyses and the potential role of DL solutions in the creative process.

2 Deep Learning Prototypes

This chapter is primarily concerned with prototyping DL solutions for problems from various domains. The presented DL solutions have passed one or more prototyping iterations, from data collection over data preparation and model training to model evaluation. Thus, they exhibit different levels of maturity.



Machine learning life cycle stages covered in Chapter 2.

Summary. *Applied Image Recognition* (Section 2.1), adapted from Griebel, Dürr, and Stein (2019): In recent years, novel DL techniques, greater data availability, and significant growth in computing power have enabled AI researchers to tackle problems that had remained unassailable for many years. Furthermore, the advent of comprehensive AI frameworks offers a unique opportunity for adopting these new tools in applied fields. Information systems research can play a vital role in bridging the gap to practice. To this end, we con-

ceptualize guidelines for applied image recognition spanning task definition, deep neural network configuration, and training procedures. We showcase our guidelines by means of a biomedical research project for image recognition.

Augmented Intelligence for Industrial Assembly Processes (Section 2.2), adapted from Krenzer et al. (2019): Empowered by machine learning and artificial intelligence innovations, IoT devices have become a leading driver of digital transformation. A promising approach are augmented intelligence solutions that seek to enhance human performance in complex tasks. However, there are no turn-key solutions for developing and implementing such systems. One possible avenue is to complement multi-purpose hardware with flexible AI solutions which are adapted to a given task. We illustrate the bottom-up development of a machine learning backend for an augmented intelligence system in the manufacturing sector. A wearable device equipped with highly sensitive sensors is paired with a deep convolutional neural network to monitor connector system assembly processes in real-time. Our initial study yields promising results in an experimental environment. While this establishes the feasibility of the suggested approach, further evaluations in more complex test cases and ultimately, in a real-world assembly process have to be performed.

Designing a Fashion Curation System (Section 2.3), adapted from Griebel et al. (2019): Online retailing has been experiencing explosive growth for years and is dramatically reshaping the way people shop. Given the lack of personal interactions fashion retailers have to establish compelling service and information offerings to sustain this growth trajectory. A recent manifestation of this is the emergence of shopping curation as a service. For this purpose, experts manually craft individual outfits based on customer information from questionnaires. For the retailers as well as for the customers, this process entails severe weaknesses, particularly regarding immediateness, scalability, and perceived financial risks. To overcome these limitations, we present an artificial fashion curation system for individual outfit recommendations that leverages DL techniques and unstructured data from social media and fashion blogs. Here, we lay out the artifact design and provide a comprehensive evaluation strategy to assess the system's utility.

Idea Generation in the Creative Sphere (Section 2.4), adapted from Griebel, Flath, and Friesike (2020): The continuing attention that artificial intelligence has received in recent years has given rise to a debate about its possible role in

creativity. Positions differ widely. Some argue that algorithms will replace human creativity altogether, while others claim that creativity is genuinely human and thus cannot be replaced. With this work, we want to leave this philosophical debate behind by looking at the specific challenges of creative processes. We distinguish between two essential aspects of creative action: divergent and convergent thinking. We investigate how different artificial intelligence algorithms can contribute to these two aspects of the creative process and unite creative AI with the existing IS design theory. We specifically investigate the domain of fashion design to showcase how artificial intelligence can assist designers. In divergent thinking, this would mean that artificial intelligence develops a large number of possible solutions/designs and therefore supports creatives in idea generation. In convergent thinking, this would mean help in idea selection. The application of artificial intelligence in creative processes could have far-reaching consequences for practices in creative domains. We are currently at the beginning of a possibly fundamental change in what constitutes a creative profession.

Kaggle Competition: Hacking the Kidney (Section 2.5): We present our award-winning solution for the segmentation of glomeruli in human kidney tissue images. The DL prototype was developed for the Kaggle data science competition *HuBMAP - Hacking the Kidney* hosted by the HuBMAP consortium. Our approach allows super-fast model training and achieves top results on the leaderboard. Our final submission (10^{th} place, gold medal rank) consists of a DL model ensemble trained on data with different zoom scales. These results were facilitated by our *efficient sampling* strategy that focuses on relevant regions (e.g., tiles that contain glomeruli and cortex). Our training procedure uses best practices for training schedules, DL architectures, augmentations, and inference leveraging the *PyTorch* ecosystem. Moreover, our solution implements capabilities for energy-based uncertainty estimation to enable human-in-the-loop refinement of difficult specimens. To test the generalizability of our approach, we tested our pipeline on image data of another tissue type. Our work is fully open source and reproducible.

2.1 Applied Image Recognition

This section is adapted from the article of Griebel, M., Dürr, A., & Stein, N. *Applied image recognition: guidelines for using deep learning models in practice* published in the proceedings of the 14th International Conference on business informatics (WI) 2019.

In recent years, novel deep learning techniques, greater data availability, and a significant growth in computing power have enabled AI researchers to tackle problems that had remained unassailable for many years. This holds especially true for voice or image recognition tasks where deep learning has demonstrated its remarkable capability of revealing structures in unstructured high-dimensional data. Given the wide availability of such data, deep learning applications can be used in many areas of science, business and administration (LeCun, Bengio, and Hinton 2015). At this point, a McKinsey study estimates the potential of AI applications to create between $3.5 trillion and $5.8 trillion in value annually across nine business functions in 19 industries (Parker et al. 2018). A case in point for image recognition applications is the health care sector where deep learning in conjunction may offer a critical complement to the gold standard of randomized controlled trials by supporting massive observational studies that were not feasible before (Agarwal and Dhar 2014). While there are already many successful biomedical applications enabled by deep learning applications, there is still a great need for innovative solutions. Grand Challenge[5] lists 167 data science competitions for biomedical image analysis over the last decade. These challenges comprise a wide range of applications, from ultrasound nerve segmentation, determination of skeletal age, and multiple sclerosis segmentation to different sorts of cancer detection and classification. A recent example is the Kaggle Data Science Bowl 2018 (Caicedo et al. 2019) that aims to develop algorithms to speed up research for almost every disease, from lung cancer and heart disease to rare disorders or to the common cold. While the IS community actively engages in various healthcare-oriented fields such as health care management (Wager, Lee, and Glaser 2017), health care services (Yaraghi et al. 2015) or mental health therapy programs

[5] `https://grand-challenge.org/challenges/`, accessed 10.10.2018

(Lederman et al. 2014), there has been little activity towards supporting researchers with cutting-edge tools such as advanced image recognition. Yet, our community should assume a more active role in this field as it is "uniquely positioned to provide the appropriate mix of rigor along with humanistic and instrumental relevance" (Abbasi, Sarker, and Chiang 2016).

In recent years, comprehensive new AI frameworks such as Keras (Chollet et al. 2015) have emerged. They focus on fast experimentation and prototyping through user-friendliness, modularity, and extensibility. The corresponding democratization of AI allows non-AI researches to easily access powerful deep learning applications. This shifts the focus of attention from the technology to the use case. We feel that this development offers a unique opportunity for information systems researchers in facilitating the use of these tools in practical applications. Alongside this development, the availability of unstructured data, notably image data, is increasing dramatically. Images are not only present on social media platforms (Instagram, Facebook), video platforms (YouTube), satellite images (such as Planet.com), but also a growing constituent in scientific research (Chen, Chiang, and Storey 2012). As the volume of image data has vastly exceeded the capacity of manual analysis, AI is henceforth a key component for automated evaluation (Provost and Fawcett 2013). For research purposes, AI applications, as with traditional machine learning applications, are typically embedded in data mining pipelines. Existing data mining frameworks such as the guidelines put forward by Müller et al. (2016) or CRISP-DM (Chapman et al. 2000) only vaguely describe machine learning applications as part of the modeling phase, whereas they focus on tasks such as feature engineering in the data preparation phase and the data mining process itself. However, modeling is a critical and extremely complex task for the distinctive nature of deep learning (AI) methods. To this end, we seek to outline the current state of advanced image recognition and contribute to the literature by providing tangible guidelines for non-AI researchers on how to incorporate state-of-the-art AI algorithms into data mining pipelines. Thereby, we follow up on the call for embracing the value of unstructured data in the design of analytical information systems put forward by Müller et al. (2016).

2.1.1 Building Blocks for Image Recognition Applications

Supervised learning for image recognition requires a data set of labeled images (e.g., magnetic resonance or microscopy images labeled healthy or infected). To facilitate the usage by researchers outside the AI world, we want to establish general guidelines for setting up computer vision projects. To this end, we break down the image recognition into its primary building blocks – task definition, the design of the neural network, and finally, the training approach. The design task features several sub-tasks (choice of architecture, loss function, evaluation metric). To offer concise recommendations for these highly technical sub-tasks, we link the design of the neural net to the initial task definition.

Defining the Task

To effectively address the abundance of image recognition applications, it is imperative to understand the underlying problem set. Consequently, any applied computer vision project must ultimately start with a proper definition of the image recognition task at hand. The majority of applications are captured by the following main task categories:

1. *Image classification* assigns the whole image to a particular class (Haralick, Shanmugam, and Dinstein 1973).

2. *Semantic segmentation* (also referred to as pixel-wise classification) identifies every pixel that is part of a specific class while neglecting distinct instances (Provost and Fawcett 2013; Noh, Hong, and Han 2015).

3. *Object classification* (also referred to as object detection) distinguishes between different objects (instances) of classes in a picture, returning their approximate location using a bounding box (Girshick et al. 2014).

4. *Instance segmentation* localizes objects on a pixel basis (He et al. 2017a).

We want to illustrate these categories by means of a histology image containing cancer cells (Figure 2.1.1). The histology images were adapted from “The GlaS Challenge Contest” data set (Sirinukunwattana et al. 2017). Depending on the focus of the study the following questions can be addressed using image recognition:

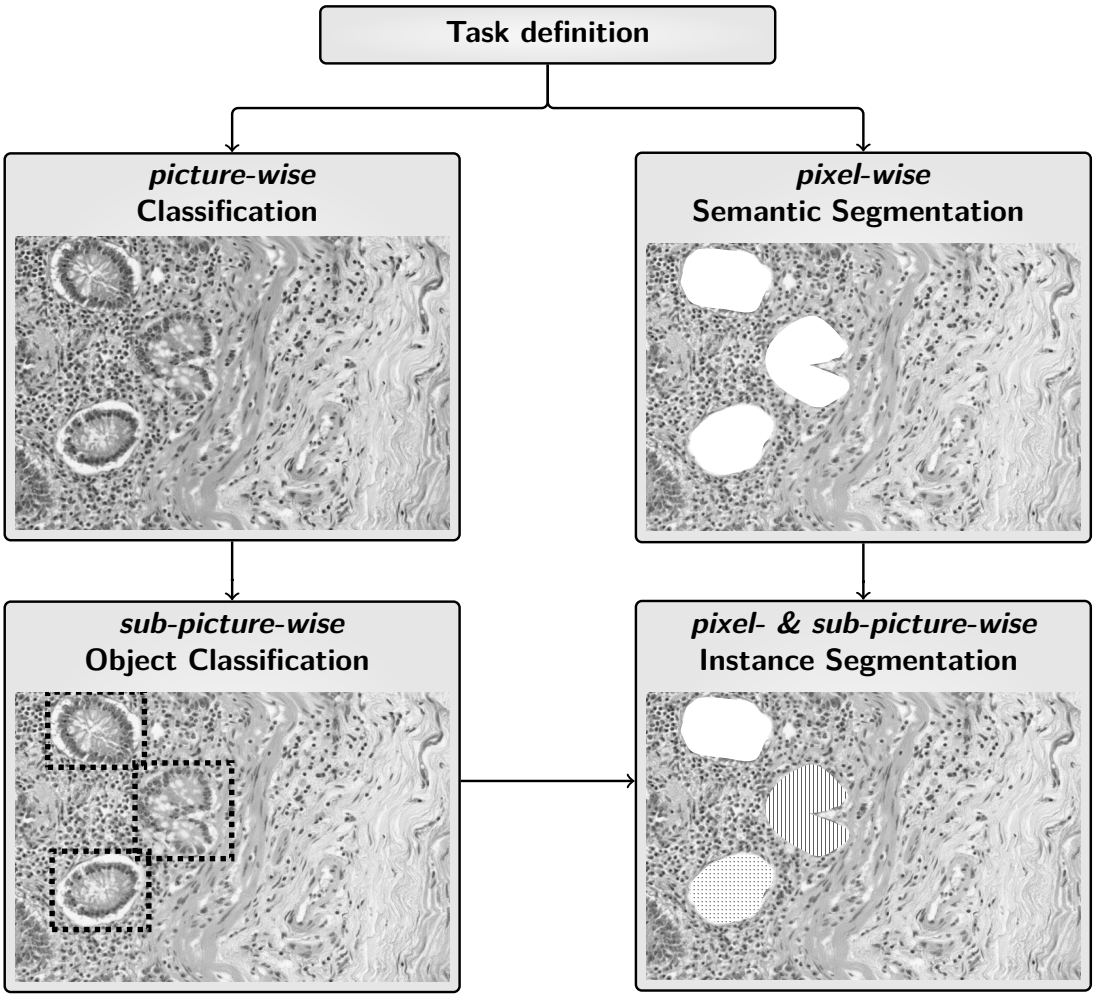


Figure 2.1.1: Taxonomy of image recognition tasks using the example of histology images that show cancer cells.

1. Classification: Does this image contain any cancer cells? If yes, assign this image to the class “cancer”.
2. Semantic Segmentation: What pixels belong to the class “cancer”?
3. Object Classification: How many cancer cells are in the image, and what is their approximate location?
4. Instance segmentation: How many cancer cells are in the image, and what is the exact (pixel) position?

Composing the Neural Network

Having identified the image recognition task, the underlying neural network for image analysis must be set up. Unlike other classification or regression techniques, this is a highly non-trivial task and requires interacting with frequently cryptic concepts and an overwhelming number of design options.

While artificial neural networks, i.e., multilayer perceptrons, have been successfully applied to various tasks since the 1980s, convolutional neural networks (CNN) have emerged as the standard for image recognition in the last decade (LeCun, Bengio, and Hinton 2015). Consequently, we focus on explaining the essential building blocks of this class of neural networks and establish best practices for each task category.

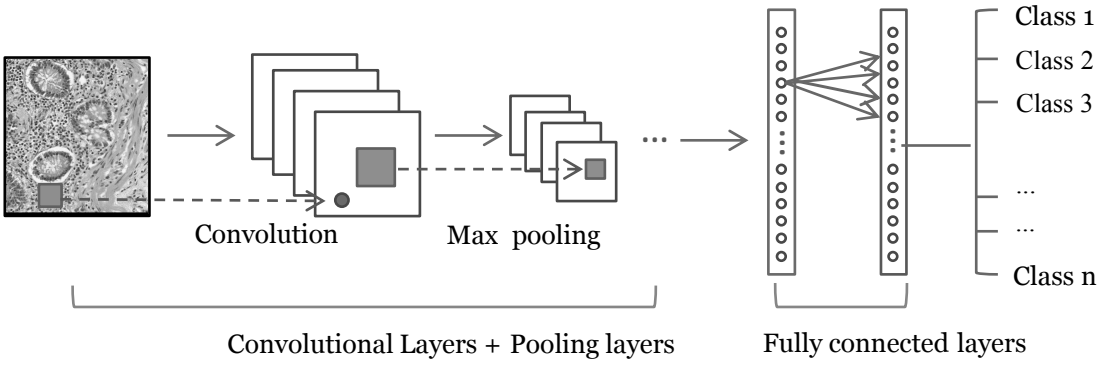
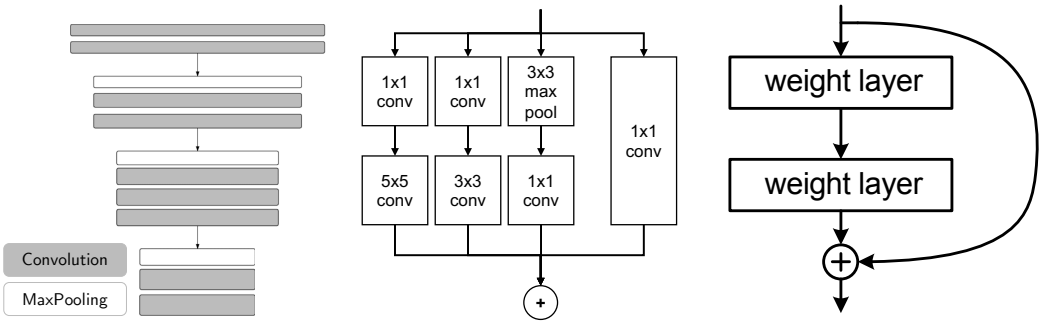


Figure 2.1.2: Example of a CNN architecture

Architectures for convolutional neural networks. The general CNN architecture is composed of three main neural layers, namely convolutional, pooling, and fully connected layers as shown in Figure 2.1.2 (LeCun, Bengio, and Hinton 2015). Convolutional layers consist of filters ("neurons") and feature maps to discover conspicuous local pattern-like edges, lines, and other visual elements. Pooling layers are typically considered as a technique to compress or generalize feature representations and reduce the overfitting on the training data by the model (Krizhevsky, Sutskever, and Hinton 2012). Fully connected layers are used at the end of the network after feature extraction and consolidation by the convolutional and pooling layers. They integrate all feature responses and provide the final classification results (Krizhevsky, Sutskever, and Hinton 2012).

The overwhelming success of AlexNet (Krizhevsky, Sutskever, and Hinton 2012), a large CNN for image classification, in the ILSVRC 2012 challenge (Russakovsky et al. 2015) has sparked significant interest in the CNN approach. Since then a vast number of architecture tweaks have emerged, each offering incremental improvements of image classification for different data sets. By using their original configuration, these networks perform the task of image classification. Due to their remarkable ability to extract features from images, they are also used as a backbone architecture for other image recognition tasks. Figure 2.1.3 provides an overview of the current main architecture choices.



(a) Schematic VGG architecture

(b) Inception module with dimension reductions

(c) Residual building block

Figure 2.1.3: Backbone architecture characteristics

The basic VGG family, introduced by Simonyan and Zisserman (2015), is typically used for its simple and easily understandable architecture (see Figure 2.1.3a). The Inception family of networks (Szegedy et al. 2015) relies on Inception modules (Figure 2.1.3b), where the input is processed by several parallel convolutional layers of different sizes whose outputs are then merged back. This enables the network itself to converge towards an optimal level of abstraction to represent a feature. Finally, the ResNet family (He et al. 2016) introduces residual blocks to the CNN (see Figure 2.1.3c). Their special features are shortcut connections parallel to the convolutional layers. This facilitates the efficient training of even deeper and more powerful networks (He et al. 2016). Moreover, these architectures are frequently used as a foundation to tailor customized CNN architectures towards a specific use.

Next, we want to match CNN architectures to the image recognition task categories. These suggestions should provide an informative starting point for determining a suitable architecture:

1. Classification: At present, the best performing classification models are, e.g., Inception-Resnet-V2 (Szegedy et al. 2016) or different versions of ResNet (i.e., ResNet51, ResNet101) or VGG.

2. Semantic segmentation: Depending on the purpose, variants of the U-Net (Ronneberger, Fischer, and Brox 2015) perform well on biomedical images such as 2D light microscopy cell segmentation. The 3D version of the U-Net is called V-Net (Milletari, Navab, and Ahmadi 2016). For more general-purpose applications we suggest a VGG based architecture such as a Fully Convolutional Network (Long, Shelhamer, and Darrell 2015).

3. Object classification and instance segmentation: As the approach of Mask R-CNN (He et al. 2017a) allows both object detection and instance segmentation within the same setting it is the best option for most multi-class segmentation applications. However, the U-Net variants can be extended by additional post-processing steps to enable instance segmentation. In particular, this approach showed very strong performance in the *Kaggle Data Science Bowl 2018* (Caicedo et al. 2019). Depending on the problem at hand, it can be rewarding to implement and evaluate both approaches.

Loss function and optimizer. The loss function (objective) and optimizer are the main components to configuring the learning process of a neural network. During the learning phase, the weights are adjusted so that the loss decreases. The loss function has to be chosen according to the task, the number of classes, or potential class imbalances. Due to its robustness and ability to handle non-linear effects, the binary cross-entropy loss is commonly used as the standard loss for binary classification tasks (picture- or sub-picture-wise). Accordingly, the categorical cross-entropy loss works well for all multi-class classification tasks

In pixel-wise segmentation tasks, there is typically an imbalance between pixel classes (i.e., many background pixels and few foreground pixels). There

are two common approaches to cope with this problem. On the one hand, Ronneberger, Fischer, and Brox (2015) propose the use of the weighted cross-entropy loss. On the other hand, the dice coefficient loss yields promising results as it handles true negatives as uninteresting defaults (Milletari, Navab, and Ahmadi 2016).

The optimizer determines the update process of the CNN by calculating the gradient. To tackle the high volumes of image recognition tasks, it is of paramount importance that the optimizer is computationally efficient, has little memory usage, and requires little tuning. We suggest using the optimizer Adam as it outperforms other common choices (e.g., SGD, AdaDelta, and RMSProp) with respect to computational overhead (Kingma and Ba 2015).

Evaluation metrics. A suitable evaluation metric is needed to assess a model's performance on the image recognition task. In contrast to the loss function, metrics do neither require to be mathematically differentiable nor used to train the model. Understanding the importance of the evaluation metric is fundamental for every data science project (Davis et al. 2007), including image recognition tasks.

Accuracy and the area under the curve (AUC) are metrics to evaluate the quality of classification results. For class-imbalanced problems, the Mathew correlation coefficient (MCC) is considered a robust measure (Powers 2011). Recall, precision, and F-Measure focus on the positive examples to capture information about the rates and kinds of errors made. The intersection-over-union (IoU) metric measures the similarity between the predicted region and the ground-truth region for an object present in the set of images. This is particularly suited for pixel-wise image segmentation tasks. There is clearly no gold standard for evaluation metrics, as they have to account for the specific properties of the given task and underlying data set. We suggest using a combination of different metrics in order to cover different aspects of the evaluation requirements. An exemplary combination of metrics for instance segmentation could be the IoU and recall. While the IoU measures the quality of the segmentation task, the recall accounts for the ability to detect all relevant instances.

Training Strategy

Having determined the composition of the neural network (by choosing an appropriate CNN architecture, loss function, optimizer, and evaluation metric) the final task of training this network on the data needs to be tackled. To this end, we introduce different concepts and best practices for model generalization, hyperparameter optimization, and hardware requirements.

Model generalization. The advantage of deep and complex CNN architectures is to better extract information from unstructured data. However, a large number of available parameters (weights) renders these networks prone to overfitting which prevents the model from generalizing well to unseen instances (Guo et al. 2016). We consider data-oriented techniques, transfer learning, and architectural tweaks to limit the overfitting tendencies of a model.

Data-oriented techniques prevent overfitting by restricting full access of the network to the training data. To this end, we apply methods such as data splitting and data augmentation. Data splitting partitions the data set into two subsets: training and validation. The model is then trained on the training data and evaluated on the validation data. Thus, it is possible to stop the training as soon as overfitting occurs. In a k-fold cross validation, this procedure is repeated k times (Kohavi et al. 1995).

Data augmentation artificially generates additional data without incurring extra labeling costs. In the case of image recognition, this is easily achieved by means of transformative methods, such as rotation, shearing, translation, flipping, elastic deformations, and random intensity jitter. This is especially useful for small data sets (Ronneberger, Fischer, and Brox 2015). Depending on the data set, some transformations should not be performed, i.e., in case of an object recognition task where objects are characterized by their shape, the shape should not be distorted.

Moreover, transfer learning leverages a pre-trained model as a feature extractor. To this end, the CNN is initialized with pre-trained parameters of a network that has been trained on another data set such as ImageNet (Russakovsky et al. 2015) or MS COCO (Andriluka et al. 2014). There are plenty of pre-trained models publicly available, e.g., in the repository of Keras (Chollet et al. 2015). The pre-trained model is fine-tuned subsequently. Thereby, the

pre-trained parameters of the initialized network are gradually adjusted to the new images during additional training steps. Depending on the problem, oftentimes the parameters of the majority of the layers are fixed while only a few parameters on top layers are adjusted. Optionally, some custom layers can be introduced and trained in parallel to fine-tune these layers on the new data set. In general, transfer learning accelerates the learning process and improves the generalization ability of a network (Guo et al. 2016).

Finally, architectural considerations such as dropout layers (Srivastava et al. 2014) can incorporate generalization approaches within the CNN composition. Dropout layers prevent the network from overfitting by randomly deactivating a share of the neurons during the training phase. Thereby, the model is forced to learn the same patterns using different neurons. During the prediction phase, the dropout is deactivated and all neurons can be utilized.

Hyperparameter optimization. There are numerous configuration settings in a CNN that can be tuned to improve the performance. Such parameters include, e.g., the activation function, learning rate, the number of training epochs, the batch size, the initial weight choices, and many more.

1. Each weight layer in a CNN is typically ensued by a non-linear activation function. The simplest activation function for binary classification decisions is the sigmoid function which is bounded between 0 and 1. The ReLU (Rectified Linear Unit) activation function (Nair and Hinton 2010) is commonly used for all layers except for the output layer in practice because of the constant slope for positive values.

2. The learning rate controls the magnitude weights adjustment after each iteration. If the learning rate is low, the training progresses slowly. In contrast, a high learning rate can prevent from converging to a possible minimum loss.

3. One epoch is when an entire dataset is passed through the neural network for training.

4. The batch size defines the number of samples propagated through the network in each step of gradient descent, i.e., learning.

Given the vast number of parameters manual tuning is impossible. Consequently, we suggest conducting an automated hyperparameter search based on either a random grid search (Bergstra and Bengio 2012) or a Bayesian optimization search (Golovin et al. 2017) to identify the promising parameter choices. Hardware Requirements. The training of CNNs requires a vast number of convolutional operations resulting in an enormous demand for computing power. Training the model on purpose-built hardware such as GPUs or TPUs is far more efficient than training on a universal CPU. The increased availability and reliability of cloud-computing services provide a strategic dynamic capability to scale up or down the IT infrastructure (Bharadwaj et al. 2013). Therefore, we suggest using Machine Learning as a Service (MLaaS) solutions. Such services are offered by all leading cloud operators.

2.1.2 A Biomedical Case Study

We illustrate the execution of an image recognition project based on the guidelines put forward above. To this end, we report learnings from a research collaboration with a group of neuroscientists. In a joint project, we developed a DL pipeline to automatically detect fluorescently stained neurons in tissue images of mice brains (Segebarth et al. 2018).

Defining the Task

To define the task, we first need to understand the underlying problem and data set. Figure 2.1.4 shows an excerpt of the image dataset obtained using a confocal microscope. The data comprises three different sub-regions of the dorsal hippocampus: dentate gyrus (DG), Cornu ammonis 1 (CA1), and CA3. As there is no ground truth for fluorescent signal segmentation, neurons are determined by their relative brightness (signal strength) to the background. For this purpose, the resulting segmentation maps are generated either by means of a heuristic, manual identification process, or by means of a (partially) automated threshold-based analysis. Due to the low signal-to-noise ratio of the data, threshold-based approaches do not work reliably as they fail to detect most of the fluorescent areas (see Figure 2.1.4).

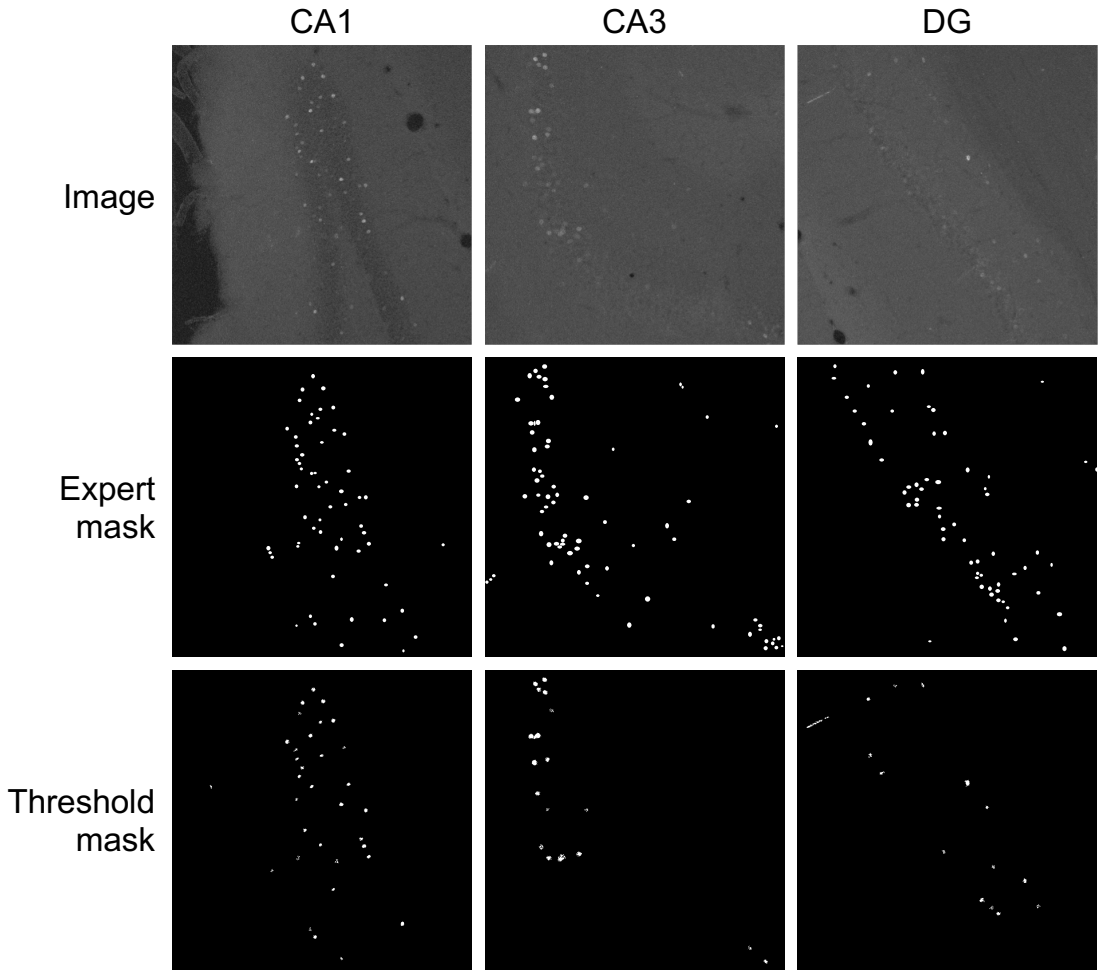


Figure 2.1.4: Different sub-regions of the dorsal hippocampus and the corresponding segmentation masks (here, the threshold only considers the 5% brightest pixels per image)

The goal of our image recognition is to automatically detect fluorescent neurons within a microscopy image. For biomedical evaluation, researchers require the position, size, and signal intensity of fluorescent neurons. Thus, our model needs to identify (i) object instances as well as (ii) the exact area (segmentation mask) rendering instance segmentation suitable for our task.

Composing the Neural Network

CNN architecture. According to the task definition we first used the Mask R-CNN approach based on a ResNet backbone architecture for instance segmen-

tation. This already yielded reasonably good results but also required a huge amount of computational resources. We also tried a U-Net-based approach similar to the winning solution of the Kaggle Data Science Bowl 2018 (Caicedo et al. 2019). In this particular case, instance segmentation is achieved by (i) performing pixel-wise binary classification with the U-Net and (ii) post-processing the resulting binary segmentation map. The post-processing pipeline includes a Watershed algorithm (Beucher 1979) and the removal of biologically implausible regions (i.e., too small or misshapen). As the U-Net approach yields better results we continue with it for the remainder of the study.

Loss and optimizer. As shown in Figure 2.1.4 the total number of fluorescent neurons (positive pixel class) is far less than the background (negative pixel class) resulting in high class imbalances among the whole dataset. Thus, we optimize (Adam algorithm) our model by minimizing a weighted combination of the cross-entropy loss and the dice loss to take advantage of their respective benefits. Here, the dice coefficient loss is particularly valuable as it handles true negatives as uninteresting defaults. We found that the outcome of the whole pipeline depends on a well-suited loss function.

Metrics. To evaluate the quality of our model we compare the expert segmentation masks to the post-processed output masks of our network. This comparison can either be performed pixel-wise or on an aggregated neuron level. For the pixel-wise comparison, we need to take the class imbalance into account. Hence, we leverage the IoU as we are mainly interested in identifying instances of the positive class (fluorescent neurons).

Considering the biomedical use case, researchers are particularly interested in the position and size of each neuron. However, in high-resolution images the exact boundaries of the neurons are difficult to define for human experts on a pixel level. As a result, there are often minimal deviations on pixel-level even though the same neuron is detected. To address this issue, we introduce another comparison process that (i) matches the corresponding neurons of two segmentation masks and (ii) calculates the accuracy as the proportion of matches divided by the total number of unique neurons found on both segmentation masks.

Training Strategy

Due to the high cost of both manual labeling and mice experiments, only a limited amount of training samples are available. Thus, we apply data augmentation as a combination of randomly rotating, flipping, and shifting the original image-mask pairs. As the shape of the neurons is important in the identification process, we do not use techniques that distort the shape (e.g., shearing). Figure 2.1.5 exemplifies this process with random parameters. Here, the origi-

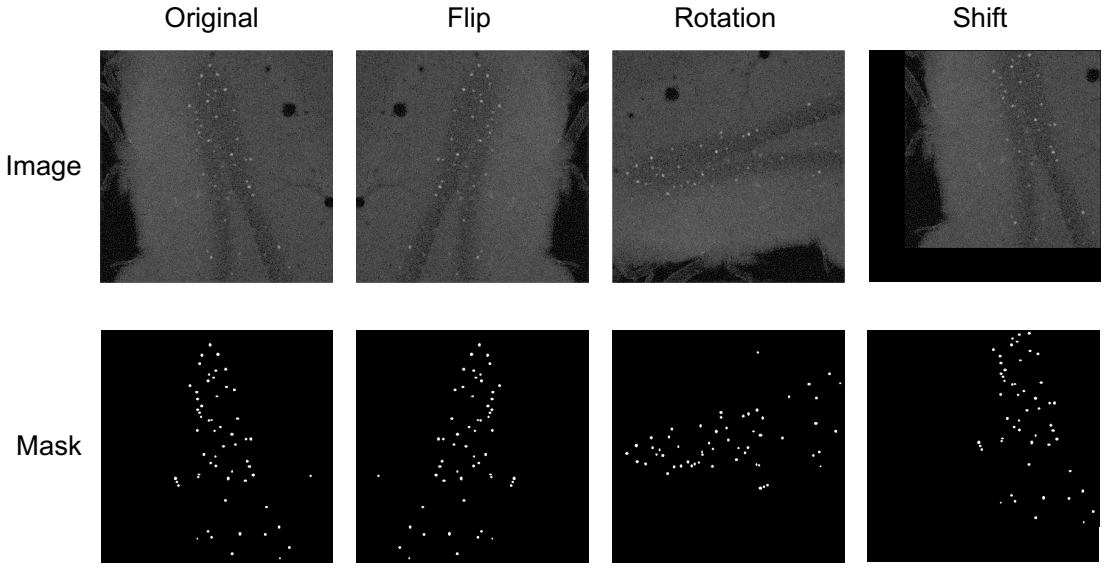


Figure 2.1.5: Data augmentation methods used in our project.

nal image-mask pair is horizontally flipped, rotated by 90 degrees clockwise and 20 percent shifted to the top and right. In light of the small dataset, data augmentation prevents overfitting and generalizes the model, e.g. by learning to detect neurons independent of their position. To further remedy the issue of limited training data we pre-trained our model on the Kaggle Data Science Bowl 2018 data set, which contains similar microscopy tissue images. To tune the parameters of the network we use a Bayesian optimization search. The model is trained and evaluated on multiple Nvidia Tesla V-100 GPUs.

To communicate our research, we provide a Jupyter-Notebook that requires no ML and almost no programming expertise. It can be executed on Google Colab with free access to high computing power[6].

[6] `https://colab.research.google.com/github/matjesg/DeepFLaSH/blob/master/DeepFLaSH.ipynb`

2.1.3 Prototype Summary

The DL solution for the segmentations of fluorescent neurons in microscopy images (Section 2.1.2) has already passed through several prototyping iterations, including data collection and annotation. The model demonstrates the potential of DL in biomedical applications and already achieves human-like performance in some cases. Automating the conventional (manual) image analysis process could reduce the workload and allow highly qualified researchers to focus on essential activities instead of tedious image labeling work. However, there are still issues regarding the reliability of the model training and the reproducibility of the results. Due to its paramount importance to the life sciences, we continue the development of this DL project along the ML life cycle in Chapter 3 and 4.

2.2 Augmented Intelligence for Industrial Assembly Processes

This section is adapted from the article of Krenzer, A., Stein, N., Griebel, M., & Flath, C. *Augmented Intelligence for Quality Control of Manual Assembly Processes using Industrial Wearable Systems* published in the proceedings of the International Conference on Information Systems (ICIS) 2019. I was primarily responsible for outlining the conceptual approach as well as for developing and documenting the DL solution (data preparation, modeling, and evaluation).

Recent advances in sensor technology, a continuing decline of hardware prices and ubiquitous networking capabilities have led to significant growth in Internet of Things (IoT) devices and applications. Fueled by innovations in machine learning and artificial intelligence, these new IoT devices become a leading driver of the ongoing digital transformation and enable a plethora of autonomous systems (Gubbi et al. 2013; Patel, Ali, and Sheth 2017). Driven by the digital transformation, an increasing number of tasks can be automated substituting human work and forcing workers to adapt to this changing environment. The impact of increasing automation has often been discussed controversially

(David and Dorn 2013; Rajnai and Kocsis 2017; Loebbecke and Picot 2015) and is attracting significant media attention. Still, many tasks cannot be fully automated. A case in point are complex assembly processes which easily surpass motion capabilities of current robot generations (David 2015; Gibbs 2016; Pfeiffer 2016). In these settings, digital transformation is not about automation but rather about assisting and improving human performance by means of smart IoT devices. As pointed out by Pavlou (2018) and Pan (2016), human-computer symbiosis, also referred to as augmented intelligence, has the potential to leverage the complementary strengths of humans and computers.

However, there is no one-size-fits-all solution to develop and implement augmented intelligence systems. As smart IoT devices have to be newly developed or at least redesigned for many use-cases, the unique combination of hardware (sensors, motors, signals) and data processing during highly specialized processes will most of the time limit the direct applicability of existing training data or pre-trained machine learning models.

By means of a use case from the manufacturing sector, we illustrate the bottom-up development process of an augmented intelligence system and highlight the important steps as well as the obstacles. Specifically, we design a wearable device for real-time quality control in an electronics assembly production step. Our example applications seeks to detect if connector systems (plugs) are properly connected during a manual assembly process. Driven by the "Poka-Yoke" principle, manufacturing companies strive to design fail-safe production processes (Dvorak 1998). It is for this reason that connector systems mechanically emit a distinctive acoustic signal ("click") to signify successful connections. However, such connections often have to be made under aggravated circumstances and outside a worker's line of sight (e.g., plugs have to be connected behind the glove compartment or in the drivetrain) while loud ambient noises overpower the click sound. Consequently, neither visual nor acoustic Poka-Yoke solutions are applicable. One way to overcome this obstacle is to augment the worker by means of a multi-use structure-borne noise sensor that can detect object vibrations beyond superhuman levels. Such sensors can be embedded in a wearable device that is positioned at the workers' wrist (close enough for reliable detection, not impairing assembly motions). The device can then continuously record a broad band of frequencies transmitted via air (acoustic signals) or vibrations (structure-borne noise). This

hardware needs to be paired with an analytic backend that identifies valid click sounds from the sensor data stream. In turn, the system can offer direct feedback with respect to the success of connections. Figure 2.2.1 illustrates a prototype of this IoT device.

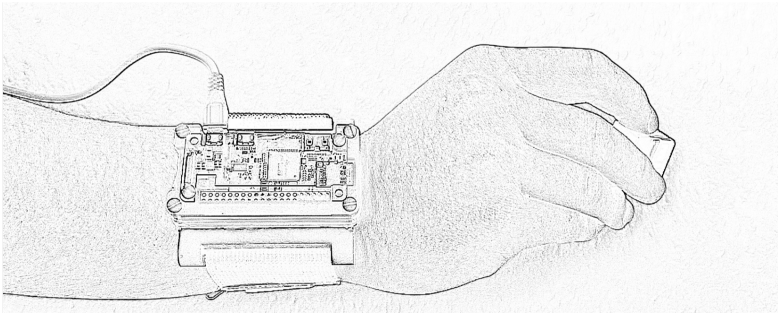


Figure 2.2.1: Prototype of the wearable IoT device

2.2.1 Conceptual Approach

We seek to complement a wearable sensor equipment with a data analytics backend to establish a real-time quality control system. To this end, we follow the Design Science Research paradigm which puts forward the development of useful artifacts as the central research goal (Baskerville et al. 2018; Hevner et al. 2004). Such artifacts can either embody (i) new solutions for known problems, (ii) known solutions extended to new problems, (iii) new solutions for new problems, or (iv) known solutions for a known problem (Gregor and Hevner 2013). Along these lines, our artifact instantiates as a new solution for a known problem as we combine existing components from different domains (information systems research, artificial intelligence) to a well-known problem from quality control. Gregor and Hevner (2013) refer to such an artifact as improvement.

The structure-borne noise sensor combined with a Raspberry Pi module is worn on the worker's wrist without restricting mobility. The device continuously streams sensor readings to a server using the Message Queuing Telemetry Transport protocol (MQTT), one of the standard IoT communication protocols (Al-Fuqaha et al. 2015). The predictive backend queries the data preparation module every second using the last five seconds of recording data. Based

on the prepared data, the classification module provides real-time feedback to the worker. Our artifact is outlined in Figure 2.2.2.

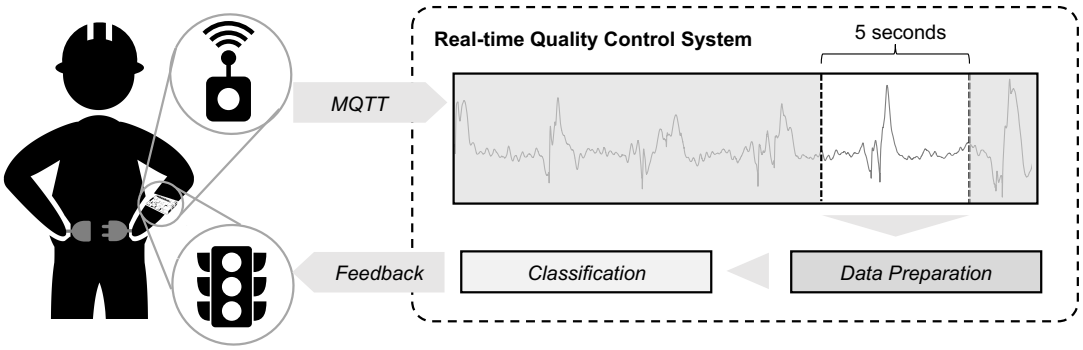


Figure 2.2.2: Artifact overview

2.2.2 Experimental Design and Data Collection

Following Basili (1996), the quality and efficacy of a system has to be rigorously demonstrated by means of an appropriately selected evaluation method. While we aim to evaluate the artifact in a real production environment, we first have to show its viability. Hence, our initial case study relies on an experimental replication of the real-world assembly process.

To collect sufficient training data, we created a training program that repeatedly instructs the test person to perform one of the following actions in the next five seconds:

- Assemble the plug appropriately and thereby generate a positive sample.
- Perform some other movement and generate a negative sample.

In order to ensure a similar distribution of the environmental sounds, the program randomly selects the action to be performed. Note that we opted for oversampling of negative examples as there is only one way to successfully connect the plugs but many ways to generate non-successful sounds (incomplete clicks, drops, walking, speaking, background noise).

Following this procedure, we collected a data set of 4,375 samples (1,525 positive and 2,850 negative). Each 5-second sample comprises an array of 160,000 sensor readings as well as a binary label (positive or negative). Figure

2.2.3 visualizes two examples of the raw data. In the right panel, a “click” is located between the two vertical dotted lines. Comparing the two samples it becomes obvious that there is a lot of noise in the data and that the correct assembly of the connector systems cannot readily be identified from raw data.

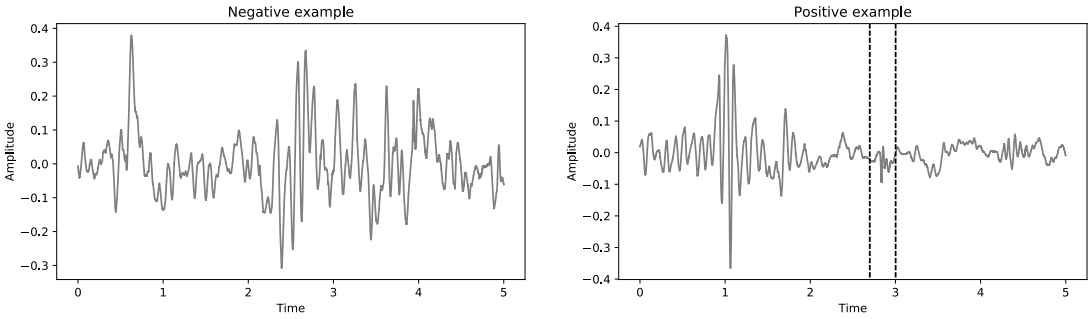


Figure 2.2.3: Sensor raw data without (left) and with “click” (right).

2.2.3 Data Preparation and Modeling

We apply a deep convolutional neural network (CNN) to classify whether or not a given sound sequence corresponds to a correct assembly of the plugs. In line with Agarwal and Dhar (2014)’s call to action, we primarily focus on problems and outcomes while limiting development efforts for new algorithms. Thereby, we follow Griebel, Dürr, and Stein (2019) and do not design new network architectures from scratch but select one from state-of-the-art research papers solving similar problems.

Data Preparation

Even though CNNs render the task of manual feature engineering obsolete, the raw data still needs to be transformed in order to effectively train meaningful models.

On the one hand, network architectures for sound classification are designed to classify an acoustic signal based on its frequency spectrum. To obtain this, we decompose each recorded five-second time window into its individual frequencies by means of the short-time Fourier transformation (Sejdić, Djurović, and Jiang 2009). This transformation splits a function of time (the sensor readings) into its frequencies (Bracewell and Bracewell 1986). Performing

the Fourier Transformation on our one-dimensional raw sensor data returns a two-dimensional spectrogram. On the other hand, neural networks converge faster and therefore perform better if the input variables follow a standard normal distribution (LeCun et al. 2012). Hence, we perform a log transformation on the spectrogram and subsequently standardize the input variables. Figure 2.2.4 shows the data preparation pipeline on a negative as well as on a positive example.

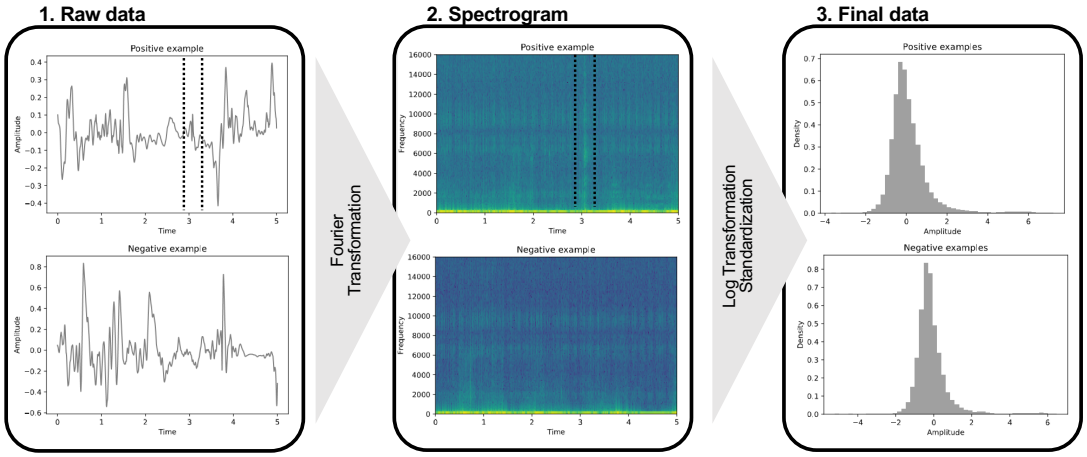


Figure 2.2.4: Data preparation pipeline

Modeling and Training

As stated above, DCASE provides best practice models for sound classification. Therefore, we adopt the current DCASE-19 baseline model[7] (Kong et al. 2018), which proved to be successful in the 2016 DCASE challenge (Valenti et al. 2017), to tackle the classification problem at hand. This CNN comprises two convolutional layers and one dense layer, followed by a sigmoid binary classification layer. For regularization, we included batch normalization (Ioffe and Szegedy 2015) after each convolutional layer and dropout (Srivastava et al. 2014) after all layers.

In order to avoid overfitting we split our data into a training set (3500 samples) and a test set (875 samples). This is done in a stratified manner, maintaining the ratio of positive and negative samples from the original data. We

[7] https://github.com/qiuqiangkong/dcase2018_task1

additionally draw a random validation set (350 samples) from the training data to monitor the performance during model training and tuning. We preserve the test set for the final evaluation.

To increase generalizability as well as training stability, data augmentation is commonly applied to train deep neural networks. For image recognition tasks, this involves random transformations of each image such as rotation, shearing, or flipping. In contrast to images, a spectrogram carries different information on each axis (i.e, frequency, amplitude, and time). Hence, we can only apply transformations that do not change the sequence of the data. This renders the addition of Gaussian noise to each training sample as a valid remaining option for our case.

We implement the final model using the Tensorflow framework (Abadi et al. 2016). The training is performed on an Nvidia Tesla P100 GPU to minimize the binary cross-entropy loss by means of the Adam optimizer (Kingma and Ba 2015).

2.2.4 Results

We implement different state-of-the-art audio classification approaches to assess the performance of our CNN. In contrast to deep neural networks, these models are based on hand-crafted features. Therefore, we extract 645 features from the spectrogram, namely the arithmetic mean, minimum, maximum, and median value for each frequency. We chose four different baseline models. These comprise two tree-based ensembles, a gradient tree boosting (XGB) (Chen and Guestrin 2016) and a random forest (RF) (Breiman 2001), as well as a support-vector machine (SVM) (Cortes and Vapnik 1995) and a Gaussian naive Bayes classifier (GNB) (Chan, Golub, and LeVeque 1982).

We chose the following evaluation metrics considering the class imbalance (more negative than positive samples) in our data set:

- *Matthews correlation coefficient* (MCC) is generally regarded as a good measure for imbalanced data (Powers 2011). It takes true positives (instances of correctly classified properly connected plugs), false positives (instances that contain falsely connected plug events but are erroneously classified as properly connected), true negatives (instances of falsely assembled plugs classified as falsely assembled plugs), and false negatives

(instances of properly assembled plugs that are erroneously classified as falsely assembled) into account.

- *Precision* reports the fraction of correctly classified correctly assembled plugs among all instances that are classified as correctly assembled, i.e., true positives divided by the sum of true positives and false positives.
- *Recall* indicates the fraction of correctly assembled plugs that are correctly classified (true positives) among all correctly assembled plugs (true positives and false negatives).
- *F-Measure* considers both, precision and recall, and is calculated as the harmonic mean of the two evaluation criteria.

Table 2.2.1: Classification results on the test set

Model	MCC	Precision	Recall	F-Measure
CNN	**98.74%**	99.67%	**98.69%**	**99.18%**
XGB	92.93%	96.32%	94.43%	95.36%
RF	90.98%	98.56%	89.51%	93.81%
SVM	76.58%	**100.00%**	68.52%	81.32%
GNB	25.22%	39.12%	99.67%	56.19%

As depicted in Table 2.2.1 the CNN achieves the best overall performance with an MCC of 98.74%, surpassing the second-best model (XGB) by 5.81%. Notably, the SVM yields a precision of 100% (CNN 99.67%). It flawlessly classified all correctly assembled plug instances as correctly assembled. This can be particularly interesting for quality control systems that require high reliability. However, such systems should preferably yield a high recall as well. This holds true for the CNN, but not for the SVM.

2.2.5 Prototype Summary

The presented prototype combines dedicated hardware with a deep convolutional neural network to perform real-time classification of the assembled plugs based on structure-borne noise signals. The prototype has completed

several development cycles and the evaluation results underline the feasibility of the suggested approach. However, our study was conducted with only a single plug and data collected from a limited number of test persons. Going forward, the test setting needs to be expanded to more complex scenarios using different plugs and additional test persons.

2.3 Designing a Fashion Curation System

This section is adapted from the article of Griebel, M., Welsch, G., Greif, T., & Flath, C. *A picture is worth more than a thousand purchases: designing an image-based fashion curation system* published in the proceedings of the European Conference on Information Systems (ECIS) 2019.

E-commerce has dramatically changed the retailing landscape and online retail continues exhibiting explosive growth in recent years (Doherty and Ellis-Chadwick 2010). This applies in particular to the fashion industry, where online sales are currently growing at an annual rate of ten percent (Amed et al. 2017). Vis-a-vis stationary retail, online fashion retail must overcome deficits with respect to product presentation as well as ancillary service offerings. Addressing these issues researchers have highlighted the importance of fostering the online shopping experience through social integration (Kim, Suh, and Lee 2013), improved visualization (Won Jeong et al. 2009) as well as optimized search (Mathwick and Rigdon 2004) and recommendations (Lin 2014). Amed et al. (2017) highlights the emergence of a new service with increasing importance to customers – *curation*.

Sebald and Jacob (2018) refer to such service as "[c]urated retailing [which] combines convenient online shopping with personal consultation service to provide a more personalized online experience through curated product selections, orientation and decision aids, and tailor-made solutions based on the customer's preferences". The increasing popularity of start-ups with curated retail logic (Modomoto or Outfittery) and the market entry of major players such as Zalando (Zalon) underline the potential of this service. In addition,

the next generation of customers is particularly open to new shopping models such as curated shopping (Heinemann 2017).

Clearly, these curation offerings critically depend on human stylists evaluating customer looks and proposing suitable outfits. Ultimately, such solutions cannot properly scale in an e-commerce environment. A naïve solution to address this scalability challenge is to replace the curator with a conventional recommendation system. One can distinguish between content-based methods and collaborative-filtering approaches (Adomavicius and Tuzhilin 2005). The former exploit similarities between item features, whereas the latter generate product suggestions based on the purchase behavior of users with similar preferences or frequently bought item pairs. Consequently, these algorithms offer a very limited form of personalization compared to curated shopping as they do not understand or even consider the style of the customer. Similarly, classic recommendation engines typically cannot incorporate cues from outside sources which are particularly relevant in the fashion domain, e.g., social media and influencers (Amed et al. 2017).

Computational understanding of fashion and clothing is fundamentally a challenge of computer vision requiring extensive analysis of unstructured image data. Not long ago, the execution of such tasks would have required an individually designed solution and extensive computational resources. However, the recent artificial intelligence (AI) revolution has brought forward comprehensive deep learning frameworks such as *TensorFlow* (Abadi et al. 2016) and *PyTorch* (Paszke et al. 2019) as well as powerful cloud-based computing platforms. Together they facilitate fast experimentation and prototyping through user-friendliness, modularity, and extensibility (Griebel, Dürr, and Stein 2019).

Several authors propose such AI-based solutions for fashion outfit recommendations (Vasileva et al. 2018; Han et al. 2017b; Wang et al. 2018) or trend forecasting (Matzen, Bala, and Snavely 2017; Al-Halah, Stiefelhagen, and Grauman 2017). However, these approaches do not explore the end-to-end automation of the curation process. Against this backdrop, our research is concerned with building and evaluating an AI-based curation system. To this end, we leverage deep learning techniques and follow up on the call for embracing the value of unstructured data in the design of analytical information systems put forward by Müller et al. (2016). Our research seeks to explore how AI components can be employed to instantiate an automated curation system.

2.3.1 Theoretical and Practical Background

Recently, deep learning enabled AI solutions for fashion have attracted great attention. While earlier computer vision models (Wang and Zhang 2011; Chen, Gallagher, and Girod 2012; Kiapour et al. 2014) mostly rely on handcrafted features, modern applications are typically build on top of deep convolutional neural networks (CNNs) that automatically learn features with multiple levels of abstraction (LeCun, Bengio, and Hinton 2015). Until today, approaches already cover fields such as clothing parsing and categorization (Yamaguchi et al. 2012; Yang, Luo, and Lin 2014; Yamaguchi et al. 2015), clothing attributes detection (Kiapour et al. 2014; Al-Halah, Stiefelhagen, and Grauman 2017) and object detection via fashion landmarks (Liu et al. 2016a; Liu et al. 2016b; Wang et al. 2018), bounding boxes (Huang et al. 2015; Hadi Kiapour et al. 2015) or semantic segmentation (Zheng et al. 2018). Moreover, researchers tackle the challenge of fashion trend forecasting using images from online shops (Al-Halah, Stiefelhagen, and Grauman 2017) or from social media (Matzen, Bala, and Snavely 2017; Gabale and Subramanian 2018). Finally, many authors focus on fashion recommendations.

Existing recommendation systems either seek to identify similar or complementary fashion items. Similar items are useful for cross-domain image retrieval, i.e., matching street clothing photos in online shops (Hadi Kiapour et al. 2015; Shankar et al. 2017; Huang et al. 2015; Liu et al. 2016a). In contrast, complementary fashion items are worn in the same outfit, for instance, a shirt that goes well with pair of pants. To this end, many approaches measure the pairwise compatibility of two items based on graphs (McAuley et al. 2015), Conditional Random Fields (Simo-Serra et al. 2015), Siamese networks (Veit et al. 2015; Tautkute et al. 2019), Conditional Similarity Networks (Veit, Belongie, and Karaletsos 2017), or unsupervised models (Hsiao and Grauman 2017).

However, an outfit is typically composed of more than two fashion items (e.g., a top, a pair of pants, and shoes), which renders pairwise compatibility insufficient. Vasileva et al. (2018) address this problem by jointly learning similarity and compatibility. Han et al. (2017b) model an outfit as a series of multiple fashion items using a bidirectional long term short term memory (LSTM) network. This approach can generate entire outfits and also processes input based on text or images or both. Nakamura and Goto (2018) extend this ap-

proach by adding a style component that is capable of learning and controlling the styles of generated outfits.

While all studies address important topics of visual fashion understanding they do not explore the end-to-end automation of the curation process. This process is commonly structured as follows (Sebald and Jacob 2018): First, customers register online and fill in a questionnaire about their fashion preferences. Afterwards, customers chose a curator, who is responsible for outfit selection. In case of special requests, customers optionally contact their curator via phone or chat. Then, based on the information on the customer, the curator selects and triggers the shipment of the personalized outfit. Finally, customers choose which garments to keep and which to return.

A closer inspection of the curation process reveals central weaknesses which can potentially limit its growth potentials.

- *Lack of immediateness:* In current curated shopping services customers may have to wait up to two weeks for delivery.
- *Lack of scalability:* Curated Shopping relies heavily on human expertise embodied by curators. Having humans in crucial positions of the process significantly limits growth potentials in times of very high employment.
- *Perceived financial risks:* Customers may expect curated shopping to be more expensive than regular online shopping due to the cost of curation (Cha and You 2018).

To address these weaknesses, we want to design a system that is at first capable of detecting fashion items in an image. Furthermore, it should learn from social media data about current fashion trends to recommend entire outfits, and finally, obtain similar articles from the recommended outfit in the product catalog of the retailer.

2.3.2 Methodology

As our research aims at building an artificial fashion curator, we follow the Design Science Research (DSR) paradigm which is particularly concerned with the development of useful artifacts (Hevner et al. 2004; Baskerville et

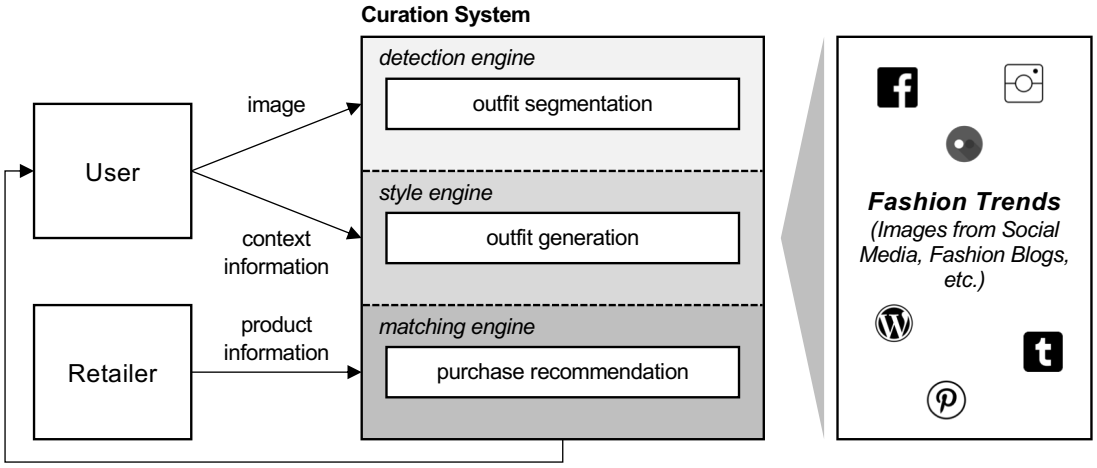


Figure 2.3.1: Conceptual approach

al. 2018). Such artifacts can either embody (i) new solutions for known problems, (ii) known solutions extended to new problems, or (iii) new solutions for new problems (Gregor and Hevner 2013). As we want to enrich the known domain of curated shopping with an innovative fashion curation system, we consider our artifact as a new solution for a known problem. Gregor and Hevner (2013) refer to such type of artifact as *improvement*.

Artifact overview

Figure 2.3.2 illustrates the three components of the artifact and their respective outputs (white boxes) by means of an exemplary user query. Here, the input comprises a picture of the user and a text query with contextual information (outfit style: casual, reference item in picture: pants). First, the picture passes through the detection engine that identifies four distinct fashion items: a blazer, a t-shirt, a pair of pants and a pair of high heels. Secondly, this information as well as the context information is fed into the style engine. This engine generates a casual outfit that goes with the previously detected pants based on its knowledge about (current) styles and trends. Finally, the images of the new outfit are forwarded into the matching engine that finds articles in the retailer's product database that are as similar as possible to the ones in the generated outfit. Subsequently, these products are recommended to the user.

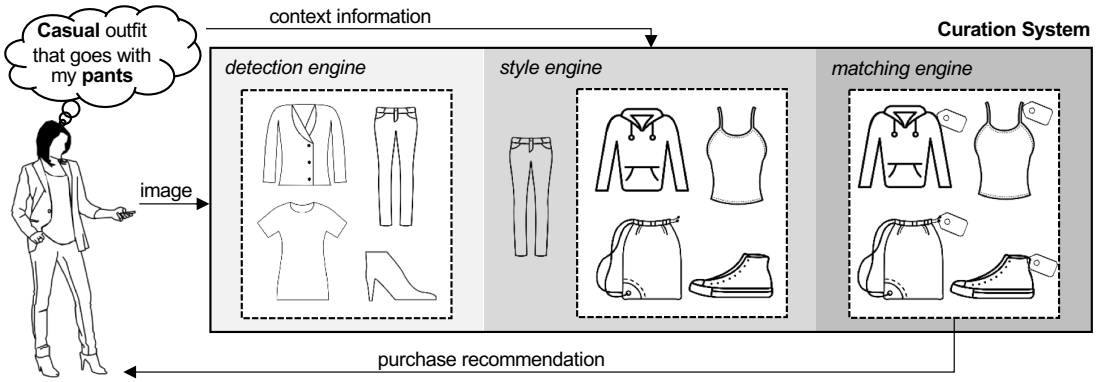


Figure 2.3.2: Functionalities of the components on an exemplary user query.

It is illustrative to describe our artifact as a guidance system following the taxonomy of Morana et al. (2017) for guidance design features in information systems. The target of our curation system is to support the customer in choosing an outfit in form of a suggestive fashion recommendation. The system works in a participative mode after a user-invoked request with concurrent timing. The intention is to recommend fashion items to a mostly novice audience. As the system incorporates the knowledge of fashion bloggers and influencers, it provides expert knowledge on outfit recommendations, which renders the content type as terminological.

Design Science Process

To carry out our study and build the artifact, we follow the DSR methodology introduced by Peffers et al. (2007). Besides conceptual principles and practice rules, the methodology provides a process for executing and presenting a DSR project. Figure 2.3.3 depicts the current status of our study within the process. At the current state, our artifact has a prototypical character and will be

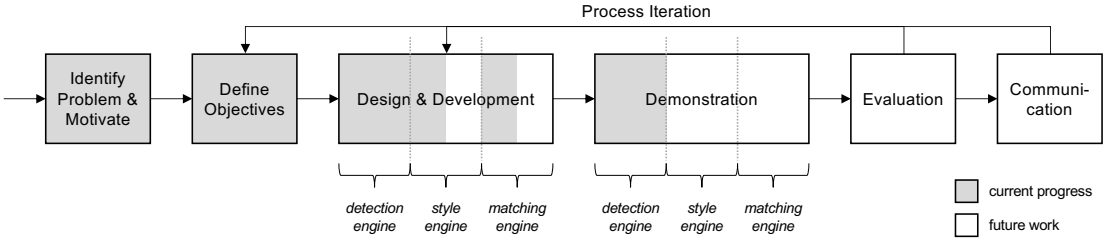


Figure 2.3.3: Project status mapped to Peffers et al. (2007) DSR process.

gradually improved on the basis of our experience and findings from the evaluation. Notably, this involves feedback from both the fashion retailer and the customer. The following sections explore the artifact's components in more detail.

2.3.3 Detection Engine

We design the detection engine for automated and reliable identification of fashion items and their exact position within images. To this end, we leverage state-of-art image recognition techniques based on supervised deep learning. Supervised learning requires a dataset of labeled images (LeCun, Bengio, and Hinton 2015), e.g., fashion images with attributes considering shape, color, or pattern. There are plenty of fashion datasets addressing various applications such as cross-scenario clothing retrieval, attributes recognition, clothing parsing, image retrieval, and aesthetic evaluation (Zou, Wong, and Mo 2018). To implement our detection engine, we utilize the brand-new street fashion dataset *ModaNet* (Zheng et al. 2018). It is built on top of the Paperdoll dataset (Yamaguchi et al. 2015) and adds large-scale polygon-based fashion product annotations for 52,377 training images and a 2,799 validation images. These annotations render *ModaNet* the only publicly available dataset that enables semantic image segmentation[8]. Figure 2.3.4 depicts some training images and the corresponding segmentation masks.

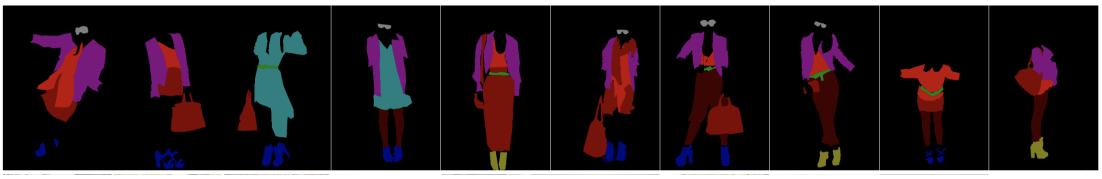


Figure 2.3.4: Training images and corresponding segmentation masks of *ModaNet* (Zheng et al. 2018).

Zheng et al. (2018) benchmark several deep learning approaches for semantic image segmentation on the *Modanet* dataset for which DeepLabv3+ (Chen et al. 2018b) yields the best results. As the *ModaNet* dataset only consists of images showing a single person (and a single outfit respectively), the choice

[8]Semantic segmentation means understanding an image at pixel level. For instance, a pixel is assigned to the class *dress* or *background*

of semantic image segmentation is appropriate and also applicable for detecting our user input images. However, we expect our fashion trend data to be more diversified, e.g., containing multiple persons in one image. To reuse our detection module for processing the trend data, it is necessary to combine the concepts of instance detection (e.g., differentiation between two persons in an image) and semantic segmentation. Several approaches such as Mask R-CNN (He et al. 2017a), MaskLab (Chen et al. 2018a) or PANet (Liu et al. 2018) have been proposed for this task. These instance segmentation methods detect objects in an image while simultaneously generating a high-quality segmentation mask for each instance.

Our detection engine rests upon the popular TensorFlow (Abadi et al. 2016) implementation of Mask-RCNN (Abdulla 2017), this setup provides deployment into production systems via TensorFlow-serving. It is based on a Feature Pyramid Network (Lin et al. 2017a), ResNet101 backbone (He et al. 2016) and uses pre-trained weights from the COCO dataset (Lin et al. 2014b). To adopt our model for fashion purposes, we used the *ModaNet* dataset for further training. This enables our model to distinguish between 13 meta fashion categories (*bag, belt, boots, footwear, outer, dress, sunglasses, pants, top, shorts, skirt, headwear, scarf/tie*). We demonstrate the detection engine in the section below.

Demonstration. Figure 2.3.5 highlights the detection functionality and limitations of our detection engine by means of two example images. In the upper picture, all five garments are detected and masked accurately. The person in the lower picture wears six different kinds of garments and accessories partly hidden from each other. Notwithstanding the high difficulties, the detection engine recognizes all garments. However, the mask of the coat (class *outer*) only classifies the left part correctly and adds the bag instead of the coat's right part. Such misclassification errors occasionally occur on photographs that contain multi-layered outfits or are captured under difficult light conditions. Assuming that customers use photos on which the relevant clothing is easily recognizable, we consider the detection engine to be fully functional for our purposes. In addition, we expect that further training, improved algorithms, and more training data will improve the model performance in the future.

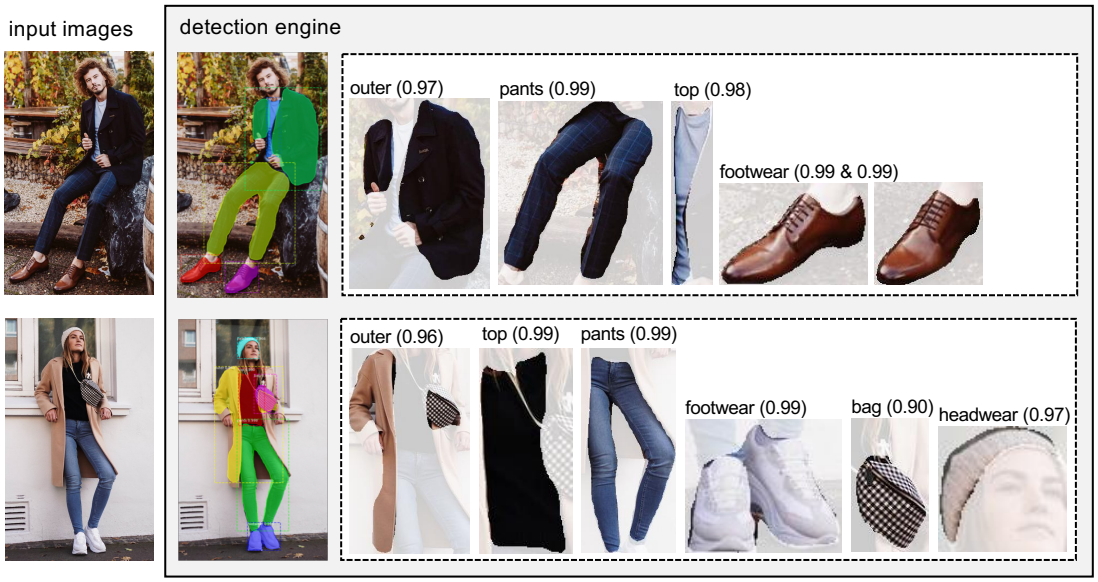


Figure 2.3.5: Demonstration of the detection engine (example images taken from zalando.com). The detected garment class names and probabilities are shown in the white box.

2.3.4 Style Engine

Our style engine creates fashionable outfits based on image and/or textual inputs. Therefore, we follow the approach of Han et al. (2017b) and Nakamura and Goto (2018) for outfit generation and style extraction. The deep learning approach comprises multiple convolutional neural networks, a visual-semantic embedding space (VSE) (Han et al. 2017a), a bidirectional LSTM and a style extraction autoencoder (SE) He et al. (2017b). The CNNs are used to extract features of the input images. These features are combined with contextual information (i.e., text input) within the VSE. Simultaneously, the SE module extracts the style of the images. Finally, the LSTM combines all information to model an outfit as a series of fashion items.

For training, we feed complete outfits (i.e., series of single fashion items) into the model. We utilize the Polyvore dataset, which comprises 21,889 outfits from polyvore.com, a former fashion website where users could create and upload outfit data (Han et al. 2017b). In order to adapt the model to current fashion trends, we are constantly scraping images from relevant social media platforms, fashion blogs, and magazines. These images are preprocessed by our detection engine (having a series of single fashion items as output) and

put into the style engine for further training. Figure 2.3.6 depicts the training procedure of the style engine.

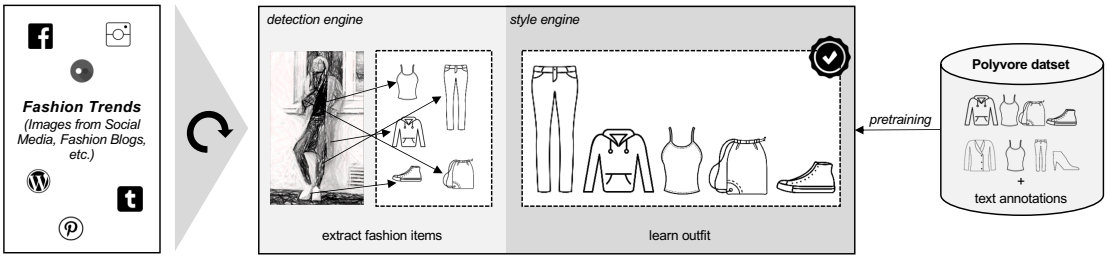


Figure 2.3.6: Training procedure of the style engine.

During inference, one or more fashion items and/or contextual information serve as input. Subsequently, the model calculates the remaining items and returns the corresponding images. This approach also enables the evaluation of a complete outfit (Han et al. 2017b).

As fashion photographs often contain multi-layered outfits that are partly hiding garments of the lower layers (see Figure 2.3.5), we need to carefully evaluate the style engine performance on such fashion items. A potential strategy to avoid malfunction of the style engine is to learn an additional model that replaces the partly visible item with a picture of a fully visible, similar item using generative models.

2.3.5 Matching Engine

As the user is interested in buying the generated outfit, the matching engine has to find similar products in the retailer's product database. Simple similarity measures such as pixel-wise comparison often fail this task given the variety of clothes. Our matching engine needs to learn abstract high-level concepts as well as low-level details.

To this end, we adapted the approach by Shankar et al. (2017) who designed a specific CNN based on the VGG-16 architecture (Simonyan and Zisserman 2015). Using the triplet-based deep ranking paradigm (Wang et al. 2014), this method is capable of ranking the images concerning their similarity. In this context, triplets are sets of three images containing a street photo and two shop photos. Street photos are real-world photos of people wearing fashion items, captured in everyday uncontrolled settings, whereas shop photos are

captured by professionals in more controlled settings. One of the shop photos matches the street photo (positive) while the other one is different (negative).

Similar to Shankar et al. (2017), we train the model with two types of triplets. On the one hand, there are out-of-class triplets containing negatives that differ significantly from our street photos. On the other hand, there are in-class triplets containing negative images which are very similar to the street photo. While the out-of-class triplets train the model to learn coarse differences, the in-class triplets let the model pay attention to minor differences as well.

The *Exact Street2Shop* dataset created by Hadi Kiapour et al. (2015) is the most popular dataset containing triplets. With only 39,479 exact matching street-to-shop pairs, this dataset is comparatively small and we need to pre-train the model. We use street-to-shop pairs (only street photo and positive) from the DeepFashion dataset (Liu et al. 2016a) for this task. The missing negative photos for the triplets are sampled based on a set of basic image similarity scoring techniques (Shankar et al. 2017).

During inference, our style engine matches the street photo to the article in the retailer's product database by finding the nearest neighbor in the embedding space of the model.

2.3.6 Proposed Evaluation Strategy

To evaluate the extent to which a DL-based curation system achieves the recommendation quality and acceptance of a human curator, we envision multiple studies. To evaluate the recommendation quality, we will ask a group of stylists (i.e., curators) to create several outfits based on different "input images" and context information. For comparison, our DL curation system generates outfits based on the same information. Hereafter, the stylists evaluate outfits (except for their own creations) without knowing the source. Another study will be based on extensive curation A/B testing. Here, one part of the customers is directed to the expert-based curation system website (A) and its traditional curation process. The other part of the customers is directed to the website with our AI curation system (B). This enables the comparison of key indicators such as turnover, return rate, and repurchases. A final study comprises a survey among the participants. It aims to measure satisfaction, trust, perceived usefulness, and ease of use based on Xiao and Benbasat (2007).

2.3.7 Prototype Summary

In Section 2.3, we sketch an artificial curation system that leverages DL techniques and unstructured data. The system design consists of three components: the detection, style, and matching engine. The detection engine has already passed through some prototyping iterations and we demonstrate its functionality. The style and matching engines are still at a conceptual stage. Further development and evaluation of the different engines would require collaboration with real-world partners, e.g., fashion retailers. This would facilitate the collection of the required training data and provide resources for the envisioned evaluation.

2.4 Idea Generation in the Creative Sphere

This section is adapted from the article of Griebel, M., Flath, C., & Friesike, S. *Augmented Creativity: Leveraging Artificial Intelligence for Idea Generation in the Creative Sphere* published in the proceedings of the European Conference on Information Systems (ECIS) 2020.

October 25th, 2018 was a noteworthy date for those gauging the rise of artificial intelligence (AI). But this time there was no high-profile match of Poker, Chess or Go, no improvement of the predictive performance on a large dataset, or the demonstration of the newest generation of self-driving cars. The event was the first successful auction of a computer-generated piece of art. "Edmond de Belamy" is a distinctly blurred portrait of a man dressed in a black robe with a striking white collar. Yet what made headlines was not the man's 18th-century attire but the fact that the painting changed hands for over US$ 400,000 in an auction hosted by Christie's[9].

There had been previous instances of intelligent systems creating images, music, and texts but those were primarily confined to the demonstration of what is technically possible. However, the result of the auction confirmed for

[9] https://www.christies.com/features/A-collaboration-between-two-artists-one-human-one-a-machine-9332-1.aspx

the first time, in a sense, that the creativity of AI can make a real contribution to human culture in general and to the art world in particular.

The example is also part of a larger trend, which extents what humans can expect of IT systems. Traditionally, computer systems were utilized to execute well-defined tasks at speeds far beyond human capabilities (e.g., calculations) or to manage *structured data* assets (e.g., databases). With the rise of more "intelligent" systems the realm of computerized assistance expanded to also cover knowledge management on *unstructured data* (e.g., automated tagging or classification). The ability of AI systems to produce unique creative results (e.g., images, music, or text) now presents the newest addition to the skillset of IT systems.

This presents an intriguing challenge to the traditional concepts and taxonomies of IT support for knowledge workers. Caught in the tension between cutting-edge technology, human behavior, and business needs, this is an exciting opportunity for information systems (IS) research to explore new forms of human-computer collaboration. And thus, to investigate how AI can best support creative activities. Designers, artists, and researchers have already turned their eyes to the use of AI in creative processes (Du Sautoy 2019; Miller 2019). Surrounding debates cover a wide range of possible consequences of using AI in creative processes. These range from the analogy of a powerful tool to the replacement of human creative activities by machines. For the most part, the discussions are extrapolations of a few cases and exude a touch of science fiction. A more rigorous, academic engagement with the topic might lead us to insights that promise a more direct implementation.

Within the IS domain, there is a strong record of creativity-related research (Couger, Higgins, and McIntyre 1993; Seidel, Müller-Wienbergen, and Becker 2010; Muller and Ulrich 2013). The major research strand focuses on the creative environment, i.e., how IS can support creativity through a combination of creativity management techniques and computer technology (Muller and Ulrich 2013). As a result, several design theories for creativity support systems have emerged (Müller-Wienbergen et al. 2011; Voigt, Niehaves, and Becker 2012).

To expand this literature in the context of AI technology, we investigate how different algorithms of AI can contribute to the creative process. Furthermore, we unite creative AI with the IS design theory of Müller-Wienbergen et

al. (2011), which we consider being particularly well suited for mapping AI methods. Building upon these insights we illustrate the findings in the context of fashion design.

2.4.1 AI in the Creative Process

Creative processes are typically characterized by two complementary stages. First, a divergent stage in which many ideas are generated and, subsequently, a convergent stage in which the most promising idea is selected and developed further. This creative "diamond" approach (see Figure 2.4.1) is the backbone of the "Creative Problem Solving" model originally introduced by Osborn in his book *Applied Imagination* (Osborn 1953). The divergence-convergence-dualism is to this day a prevailing pattern to structure creative processes. It is, for instance, part of the widely used design thinking method (Lindberg, Meinel, and Wagner 2011). Idea generation typically takes place in an almost random fashion, where quantity (as many ideas as possible) is aimed at. The goal is to generate as many options as possible which individually might be trivial or even inadequate. Yet, by not filtering ideas at this stage, truly different and novel ideas can emerge. They are not automatically shut down, rather they can inspire follow-up ideas and therefore open up novel ways of looking at a given problem. Once the idea generation stage is completed and a sufficient number of ideas has been generated the process shifts towards selecting a solution. Convergent thinking is then the process of singling out the best candidate among all available options. To this end, the generated ideas have to

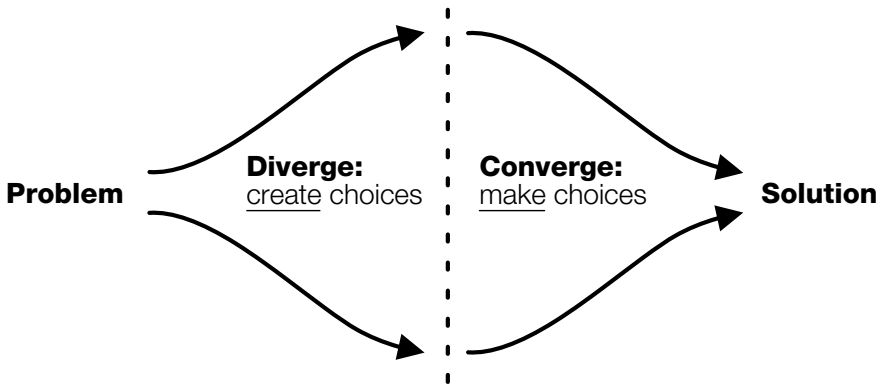


Figure 2.4.1: Divergence-convergence dualism in creative thinking.

be validated against the requirements, available resources, and technical constraints. Beyond these basic feasibility checks, quality needs to be assessed or predicted for instance by applying a suitable metric.

To scope the research context of augmenting creative processes with information technology we next map AI applications to the two stages of the creative process. We start with the converging phase as it is by its very nature a more structured process which seems more naturally reflected in IT systems.

Supervised Learning as a Form of Convergent Thinking

The convergent thinking stage seeks to identify the single best available solution to a particular problem. Traditionally, this type of thinking is therefore associated with existing knowledge as this knowledge is used in the decision-making process (Cropley 2006). The natural equivalent in the realm of machine learning and artificial intelligence is *supervised learning*. This class of machine learning procedures infers a function from labeled training data consisting of a set of training examples to subsequently predict previously unknown instances (Mohri, Rostamizadeh, and Talwalkar 2012). Supervised learning can be further separated with respect to the nature of the target variable: If it is categorical, we refer to the setting as a *classification task*, if it is numerical one is faced with a *regression task*.

Casting these instances to the setting of convergent thinking in creative processes is straightforward: If we assess solution suitability in a binary evaluation scheme (suitable vs. not suitable) the process corresponds to a classification system. Conversely, if we seek to assess idea quality along a continuum, we are in the regression learning setting. Such supervised learning applications are already commonly used in creative processes across different industries. Notable examples include Burbidge et al. (2001), who describe how classification algorithms can support drug design research. Similarly, (Christensen et al. 2017) leverage text mining to identify promising new product ideas in online communities.

Unsupervised Learning as a Form of Divergent Thinking

As highlighted above the divergent thinking stage is about generating many options. (Müller-Wienbergen et al. 2011) note that “divergent thinking requires

imagination, provocation, unstructured syntheses, serendipitous discovery, and answers that break with conformity". Human brains achieve divergent thinking by virtue of accessing and associating concepts stored in long-term memory systems.

Given the analogy between convergent thinking and supervised learning one is tempted to draw a direct analogy between the divergent thinking paradigm and the AI research area of *unsupervised learning*. Here, systems learn from unlabeled data by either identifying common structures in the data (clustering) or by establishing compact representations (dimensionality reduction). Fundamentally, both variants of unsupervised learning offer means for organizing knowledge more efficiently. Notably, (Tassoul and Buijs 2007) argue that such activities may constitute a phase of their own between divergent and convergent thinking.

Interestingly, for structured creation tasks which are characterized by reusable patterns such as music or text, variational auto-encoders (VAE) (Bowman et al. 2016; Roberts, Engel, and Eck 2017) as well as recurrent neural networks (Eck and Schmidhuber 2002; Sutskever, Martens, and Hinton 2011) have successfully been applied to generate new content. Similarly, there have been successful instantiations of *neural style transfer* (Gatys, Ecker, and Bethge 2016; Huang and Belongie 2017), where properties of some object (e.g., impressionistic painting style) are "transferred" upon another input image as illustrated in Figure 2.4.2. The style transfer paradigm offers a powerful tool for

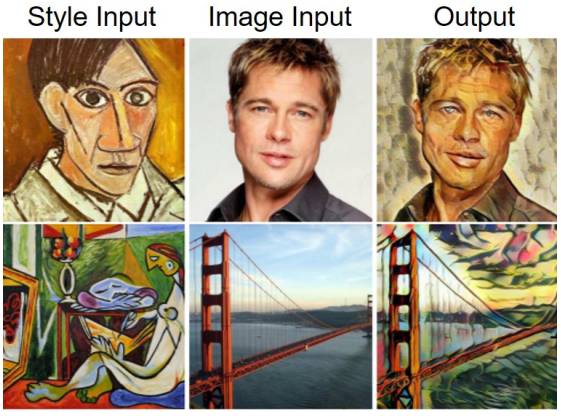


Figure 2.4.2: Example of style transfer (Huang and Belongie, 2017).

what-if questions circling around alternative design variants of a given visual instantiation. Such divergent AI systems are typically referred to as generative.

While both forms of generative models are impressive in their own ways, they fundamentally suffer from a narrow scope as framed by the rigid underlying structure - music and text are generated from discrete building blocks while style transfer boils down to a mapping from input to output. Consequently, traditional unsupervised learning approaches have been unable to reflect the full range of creative activities as necessitated by divergent thinking.

The GAN Revolution

Generative AI systems have exhibited impressive performance in structured, repetitive tasks of content creation but were fundamentally incapable of synthesizing completely new contents. This conventional wisdom of AI limitations was torn apart when Goodfellow et al. (2014) introduced the ground-breaking concept of generative adversarial networks (GAN). Facebook's chief AI scientist Yann LeCun called GANs "the coolest idea in deep learning in the last 20 years". GANs pit two neural networks – the generator and the discriminator – against each other in a competitive manner. The main idea is typically exemplified by a stylized setting where the generator acts as an art forger and the discriminator works as an art curator. The forger creates artworks trying to fool the curator into believing that these are authentic. Initially, the forger has essentially no clue and will have all of his paintings rejected by the curator. However, over (a very long) time the forger understands which image traits are successful and which are not and continuously improves the quality of the counterfeit paintings. Simultaneously, the curator has to step up his game and get better at telling apart fake and real artworks. Ultimately, we end up with a very skilled forger *and* a very skilled curator. Note that these two roles exactly mimic the spheres of divergent (generative) and convergent (discriminative) thinking. This is illustrated in Figure 2.4.3.

Current standard implementations (e.g., DC-GAN by Radford, Metz, and Chintala 2016) instantiate both generator and discriminator by means of convolutional neural networks – very similar to auto-encoder structures. However, decoupled training of the encoding and decoding parts in the architecture unleashes the creative power of GANs. This essentially resembles the necessity

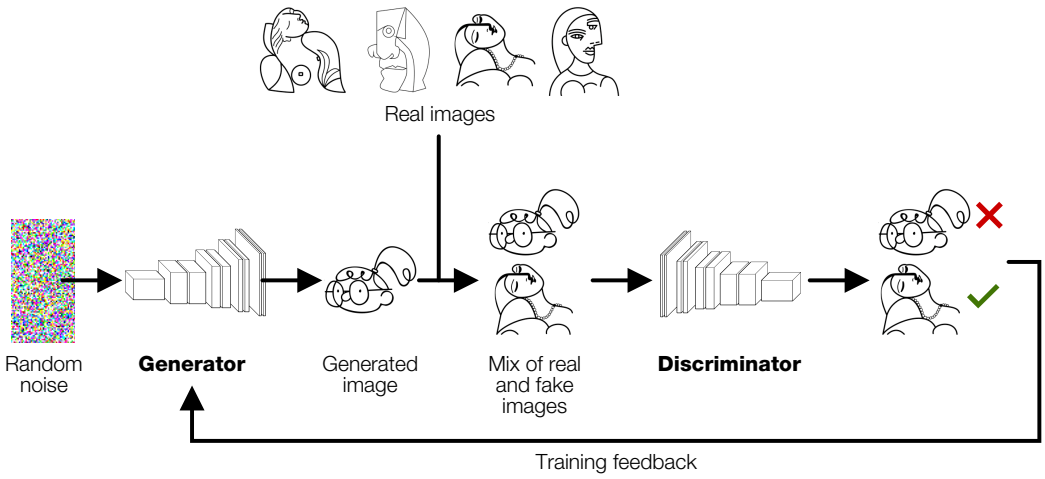


Figure 2.4.3: Basic DC-GAN architecture.

of separating idea generation and idea evaluation in human brainstorming processes to ensure a large variety of ideas. The GAN technology has seen wide adoption in various areas of research as well as very unique practical applications. For instance, Hwang et al. (2018) built a GAN for dental restorations. Their model creates dental crowns by considering natural spatial profiles between opposing teeth, which is hard to account for by technicians but important for proper biting and chewing.

The chAIr project[10] constitutes a further application. Here, the designer trained a GAN on a dataset of iconic 20th-century chairs to "generate a classic". The resulting model was used to generate new chairs which in turn were used for the actual creation of prototypical chair designs.

2.4.2 Uniting Creative AI and Design Theory

Müller-Wienbergen et al. (2011) positioned the importance of information systems as a central source of inspiration for new ideas and creative problem-solving. Their design theory puts forward the need for supporting both convergent and divergent thinking in systems that facilitate creative work through knowledge provision. As proposed by Gregor, Jones, et al. (2007) the theory is based on eight components, including the design requirements depicted in Ta-

[10] https://philippschmitt.com/work/chair

ble 2.4.1. The usefulness of the systems they describe ultimately rests on structuring and presenting the underlying knowledge base. We argue that state-of-the-art AI algorithms can go a step further and support converging and diverging creativity through automatic curation and creation of ideas. Thereby enormous potential in the creative process can be unleashed.

Table 2.4.1 maps the design requirements by Müller-Wienbergen et al. (2011) to their corresponding AI applications. We will highlight these opportunities in the next section.

Table 2.4.1: Design requirements and their corresponding AI applications.

Creative Component	**Design Requirement**	**Exemplary AI Approach**
Convergent	*C1*: Organize knowledge hierarchically; *C2*: Provide diverse perspective on existing knowledge	Sorting and exploring unstructured data (images/text) via unsupervised learning methods such as T-distributed Stochastic Neighbor Embedding (Van der Maaten and Hinton 2008)
	C3: Enable dynamic filtering of the knowledge database	Supervised Learning for automated classification (tagging)
Divergent	*D1*: Provide external stimuli	VAE and DC-GANs to recombine and create new ideas
	D2: Provide different levels of stimuli	Customizable and controllable levels of content generation, e.g., using conditional GANs (cGAN) (Mirza and Osindero 2014)
	D3: stimulate both symbolic systems of human cognition	Content generation using Visual Semantic Embeddings for images and text (Frome et al. 2013)

2.4.3 AI-assisted Fashion Design

To illustrate the practical relevance of AI for supporting creative processes we now turn to one of the oldest trades in humankind – fashion design. Over the last decades, this industry has seen a constant shift towards a quicker

turnover of collections (fast fashion) as well as increasing customer desire for individual design. It is hence not surprising that leading innovators have started to explore the possibilities of artificial intelligence across all parts of the fashion value chain. Fashion technology companies such as StitchFix or Zalando already rely on AI as a crucial component of their business model and employ large research teams to retain their competitive edge. Here, we want to highlight concrete AI application opportunities in the context of the fashion design process.

Fashion Design Process

Due to its myriad of individual approaches, changing contexts, and altering environments, the fashion design process is very complex and hardly traced. To establish a common understanding, McKelvey and Munslow (2011) summarize the fashion design process as depicted in Figure 2.4.4.

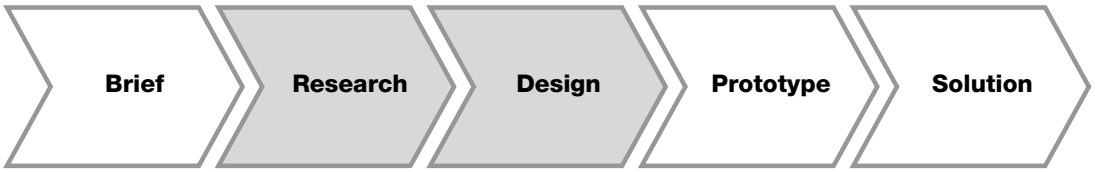


Figure 2.4.4: Fashion Design Process (McKelvey and Munslow 2011).

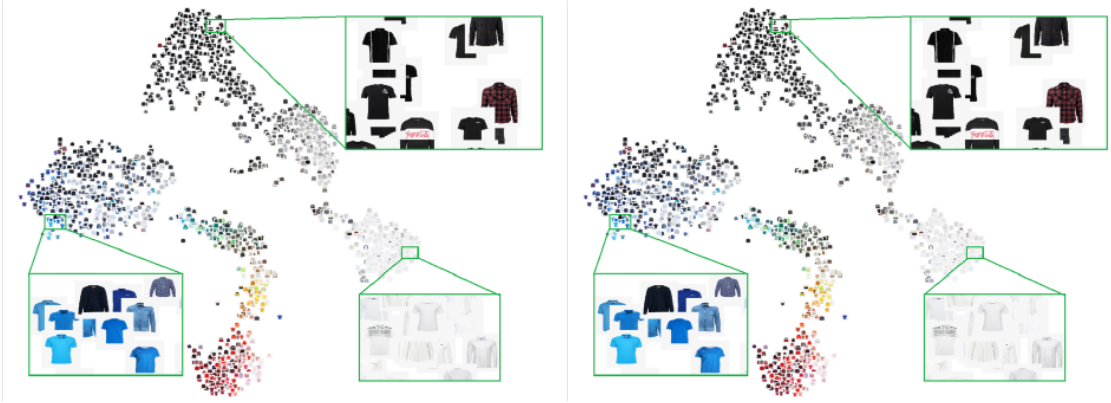
The design brief phase emerges from the design situation and defines the goals and constraints as well as the problems to be dealt with. The research phase sketches the period of collecting materials for creative inspiration and experimentation. This comprises general topics such as market analysis and trend forecasting, as well as specific product-related topics such as colors, shape, pattern, texture, and fabrication. The results are further tested during prototyping. Here, an additional emphasis is placed on construction, proportion, cut, drape, fastening, movement, and stretch of the samples. Finally, the design and quality of the samples are realized in the solution phase. At this point, the performance, sales, and merchandising of the potential products are evaluated.

Selected Fashion AI Applications

Given this fashion design process, we seek to carve out opportunities for augmenting human creative processes. Natural candidates include the research phase which has to process huge amounts of data as well as the design phase where designers need to create a meaningful solution space for an upcoming selection. We will also relate our findings to the design principles from the previous section. The fashion AI applications outlined below are only of a prototypical nature and need to be adapted and evaluated in further studies.

Research: Style Forecasting and Exploration. As stated above, trend forecasting usually happens during the research phase. AI presents a great opportunity to support this process. Every day, billions of photos are uploaded to social media platforms and blogs. These images are crammed with information on people's lives and preferences – as well as the clothes they are wearing. Matzen, Bala, and Snavely (2017) leverage this abundance of data to understand fashion and style trends by analyzing clothing and fashion across millions of images. Leveraging state-of-the-art supervised deep learning techniques, they discover visually consistent style clusters that capture useful visual correlations. Thus, they derive visual insight, producing global and percity fashion choices, and spatio-temporal trends. Following this trajectory, the European Union established the FashionBrain project, envisioning to understand Europe's fashion data universe. The project aims at combining data from different sources, comprising manufacturers and distribution networks, online shops, large retailers, market observers, call centers, press, magazines, and social media to predict upcoming fashion trends (Checco et al. 2017).

Another helpful application for (fashion) research is to explore data through visualization of embeddings. Conditional Similarity Networks (CSNs), introduced by Veit, Belongie, and Karaletsos (2017) learn embeddings that are differentiated into semantically distinct subspaces. These subspaces can be trained to capture different notions of similarities, such as *sleeve length* (Figure 2.4.5a) or *sleeve color* (Figure 2.4.5b). Dimension reduction and visualization techniques, e.g., TSNE (Van der Maaten and Hinton 2008), allow a low dimensional representation of the high dimensional embedding space. These examples address the convergent design requirements C1-3.



(a) Sleeve length subspace **(b)** Color subspace

Figure 2.4.5: TSNE visualization of the CSN subspaces

Design Phase: Inspiration. In contrast to the forecasting problem, we need to generate AI tools to support the development of inspiration for themes, color, texture, or shape. To this end, we train a DC-GAN to create new T-shirts designs as depicted in Figure 2.4.6. The DC-GAN is trained on a dataset comprising about 6,000 different T-shirts (mainly short, but also long-sleeved). The generator of the DC-GAN does not have any information on existing fashion conventions, which surprisingly leads to the creation of T-shirts with one long and one short sleeve. This little idiosyncrasy depicts the potential of AI to explore new concepts of fashion by enabling the designer to think about new concepts.

Based on a DC-GAN, Kato et al. (2017) pitch a first draft of a GAN enabled clothes design, framed as DeepWear. Their model learns the feature of specific brand clothes and then generate images of new clothes. These images

Figure 2.4.6: T-shirts created by a DC-GAN trained on T-Shirts and long sleeves.

are used as a basis for pattern and clothes design. Beyond the capabilities of the simple DC-GAN, BigGAN (Brock, Donahue, and Simonyan 2019) presents another great opportunity. It creates high-resolution images and allows fine control over the trade-off between sample fidelity and variety by truncating the latent space. Moreover, it is possible to create interpolation images between the different classes it is trained on. Figure 2.4.7 shows the results of an interpolation between category T-shirt and space rocket of a BigGAN trained on the Imagenet dataset (Russakovsky et al. 2015). The images in the center could have inspired the designer for a space-related theme. DC-GANs relate to design requirement D1, whereas BigGAN relates to D1 and D2 as it also provides controllable stimuli on different levels.

Figure 2.4.7: BigGAN interpolation results from T-shirt to rocket.

Design Phase: Experimentation. Finally, neural networks also facilitate the task of experimentation with shape, color, and pattern leveraging style transfer concepts. Yildirim, Seward, and Bergmann (2018) propose a method that disentangles the effects of multiple input conditions in such systems. Thereby, their model allows control over color, texture, and shape of a generated garment image. This method is capable of generating novel and realistic images of clothing articles. It constitutes a variant of a cGAN as proposed for design requirement D2.

2.4.4 Prototype Summary

In Section 2.4.3 we showcase several DL prototypes that have the potential to enhance the traditional fashion design process. The first prototype addresses the style exploration problem. The presented CSN facilitates the visualization of embeddings that capture different notions of similarity, such as sleeve length or color. The second prototype, a DC-GAN that designs T-shirts, can be used for inspiration during the design phase. The last prototype, BigGAN, can also be applied for inspirational purposes. It allows interpolating between different visual concepts to develop new ideas (e.g., T-shirt to rocket). However, these applications are still at an early prototyping stage of their life cycle.

2.5 Kaggle Competition: Hacking the Kidney

This section outlines our approach for the Kaggle data science competition *HuBMAP – Hacking the Kidney* hosted by the HuBMAP consortium. Our solution for the segmentation of glomeruli in human kidney tissue images won a gold medal (10th place on the private leaderboard[11]), the *Innovation Prize*, and the *Most Entertaining* award[12].
Team *deepflash2* leader: Matthias Griebel (matjes); Team members: Philipp Sodmann (theudas), Thomas Lux (maddonix)

Competition Overview[13] The adult human body contains about 37 trillion cells. Determining the function and relationships among these cells is a monumental endeavor, and many areas of human health could be impacted if we better understood cellular activities. Similar to the way the Human Genome Project maps all human DNA, the Human BioMolecular Atlas Program (HuBMAP) is working to catalyze the development of a framework for mapping the human body at the level of glomeruli functional tissue units for the first time in history. HuBMAP aims to be an open map of the human body at the cellular level.

[11] `https://www.kaggle.com/c/hubmap-kidney-segmentation/leaderboard`
[12] `https://hubmapconsortium.github.io/ccf/pages/kaggle.html`
[13] adapted from `https://www.kaggle.com/c/hubmap-kidney-segmentation/overview`

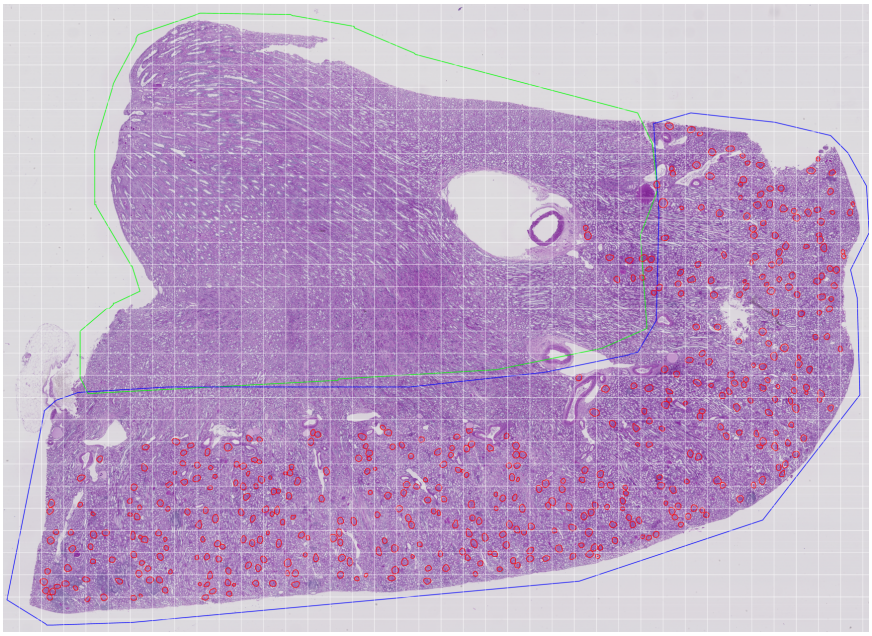


Figure 2.5.1: Image "b9a3865fc" (31295×40429 pixel) including annotations for glomeruli (red) and anatomical regions (cortex:blue; medulla:green).

The competition *Hacking the Kidney* starts by mapping the human kidney at single-cell resolution. The challenge is to detect functional tissue units (FTUs) across different tissue preparation pipelines. An FTU is defined as a "three-dimensional block of cells centered around a capillary, such that each cell in this block is within diffusion distance from any other cell in the same block" (Bono et al. 2013). The competition's goal is to implement a successful and robust glomeruli FTU detector.

Data Description[14] The competition data includes 11 fresh frozen and 9 Formalin-Fixed Paraffin-Embedded (FFPE) PAS kidney images. Glomeruli FTU annotations exist for all 20 tissue samples; 15 are shared for training, five are used for testing (public test images). The private test set (undisclosed) is larger than the public test set. All images are provided as very large (>500MB - 5GB) TIFF files. Both the training and public test sets also include anatomical structure segmentations. Teams are invited to develop segmentation algorithms that identify glomeruli in microscopy data. Figure 2.5.1 shows an exemplary competition image, including annotations.

[14]adapted from `https://www.kaggle.com/c/hubmap-kidney-segmentation/data`

Competition Metric[15] The competition is evaluated with the Dice score, which is commonly used to compare the pixel-wise agreement between a predicted segmentation and its corresponding ground truth. The dice score is defined as

$$\frac{2|X \cap Y|}{|X| + |Y|},$$

where X is the predicted set of pixels and Y is the ground truth set of pixels. The leaderboard score is the average Dice score for each image in the test set.

2.5.1 Methodology

Our solution follows the workflow depicted in Figure 2.5.2. The results can be reproduced using different Jupyter notebooks (kernels) that are publicly available on Kaggle (Appendix Table B.1). The *File Conversion* and *Sampling Preparation* notebooks convert the images and masks into *.zarr* arrays to allow memory-efficient sampling during *Training* and *Validation*. The *Inference* kernel is primarily designed for predictions on the public and private test set.

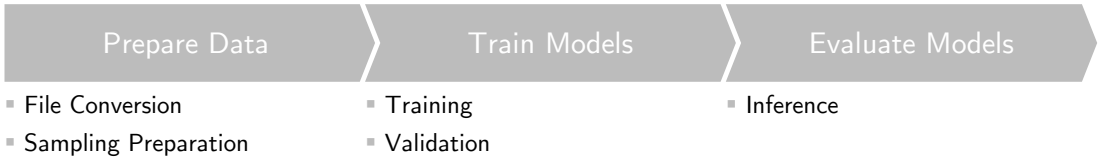


Figure 2.5.2: Proposed workflow.

2.5.2 Efficient Sampling

A common approach for DL model training on very large (>500MB - 5GB) image files is to decompose the images into smaller patches (tiles), for instance, by using a sliding window approach. However, the whole slide images in the competition data only contain a few relevant regions. In contrast, large areas of the images are either blank or contain tissue without the target class. Instead of preprocessing the images by saving them into fixed tiles, we combine two sampling approaches that are performed at runtime:

[15] adapted from `https://www.kaggle.com/c/hubmap-kidney-segmentation/overview/supervised-ml-evaluation`

Figure 2.5.3: Exemplary sampling of 512×512 pixel tiles from image "0486052bb" during one training epoch, showing the segmentation masks of the annotated glomeruli (top left) and anatomical region probabilities (top right). The colors indicate the sampling probabilities, from low (purple) to high (yellow). The sampled tiles (squares) during one training epoch are depicted at the bottom.

1. Sampling tiles via center points in the proximity of every glomerulus. This ensures that each glomerulus is seen during one training epoch at least once.

2. Sampling random tiles based on region probabilities (e.g., medulla, cortex, other).

We use the provided anatomical information to sample examples of the cortex region more often than the medulla regions, as glomeruli have a higher abun-

dance in this region. We also sample a few tiles outside the anatomic regions to ensure that our model can interpret these. Figure 2.5.3 depicts the sampling results of one image during one training epoch.

This biologically inspired sampling exhibits some desired properties during model training. Considering the exemplary batch of 16 images and the corresponding pixel distribution (Figure 2.5.4), we see that the data distribution loosely follows a normal distribution. This property is beneficial when using pre-trained models and generally speeds up learning and leads to faster convergence during the training of artificial neural networks. Moreover, the sampling is model-agnostic and can be used with any model on several other tasks (e.g., classification, object detection).

In addition, a beneficial side effect of our sampling method is that FTUs that were missed during the annotation process are rarely sampled during training.

2.5.3 Training and Evaluation

Our training procedure is based on best practices for training schedules (*fastai*), architecture (*segmentation-models.pytorch*), image augmentations (*albumentations*) and inference (tile shift and gaussian weighting, Isensee et al. 2021).

Hyperparameter search. We trained and validated our models using five-fold cross-validation to find the best hyperparameter settings. Each fold is trained on twelve and validated on three whole slice images. During training, we logged all parameters as well as pixel-level metrics (precision, recall, (soft) dice score, loss) for each epoch using *wandb*[16]. Throughout the challenge, we trained and tested different DL architectures for image segmentation, such as U-Net (Ronneberger, Fischer, and Brox 2015), U-Net++ (Zhou et al. 2018) or DeepLabV3+ (Chen et al. 2018b). We also tried different encoders, e.g., ResNets (He et al. 2016) and EfficientNets (Tan and Le 2019). However, we found no significant difference in performance. Therefore, we decided to use a reasonable small encoder (EfficientNet-b2) and a standard U-Net in the end. We also tried different optimizers (SGD, AdamW, Ranger, Madgrad) and found that Ranger

[16] `https://wandb.ai`

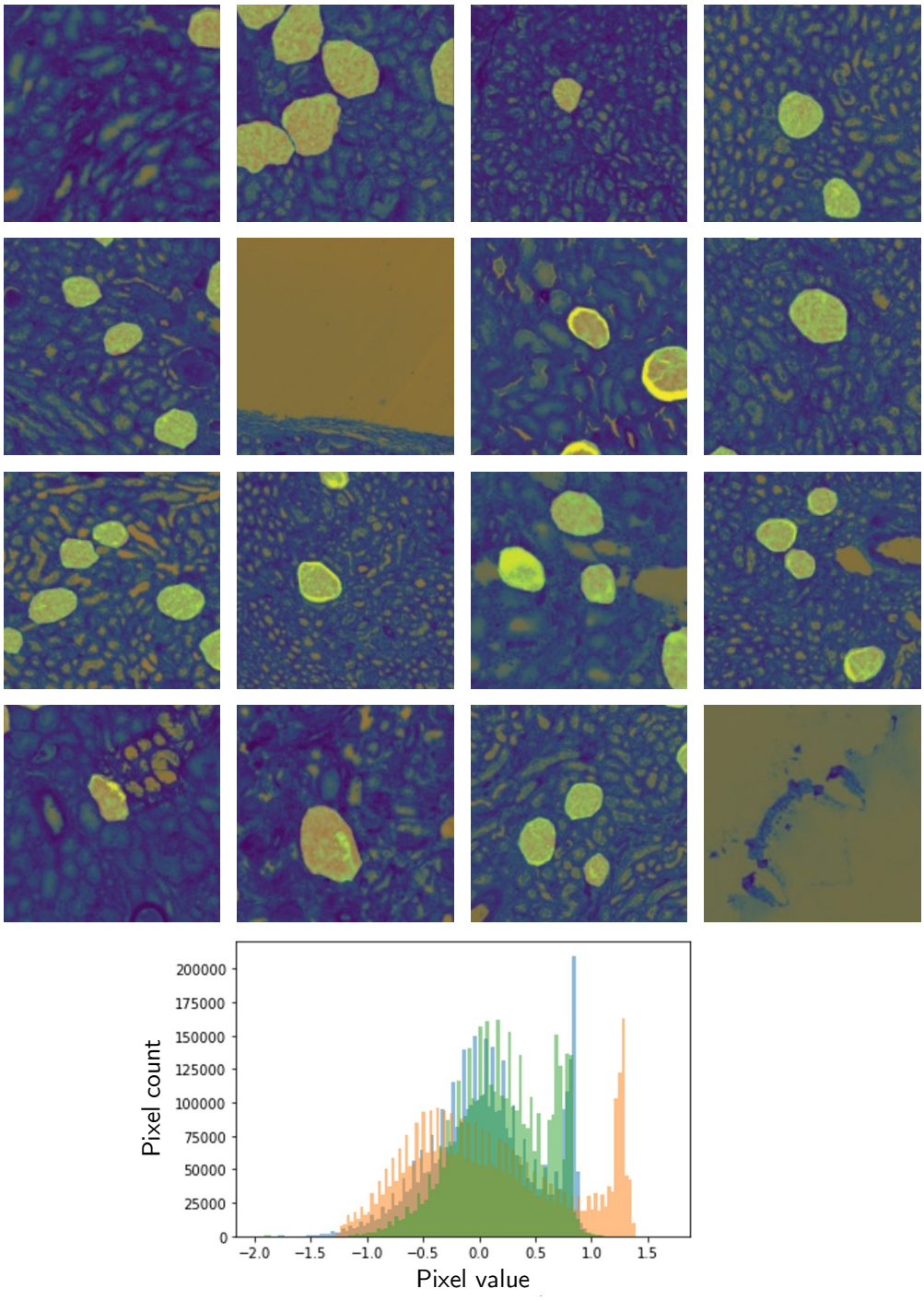


Figure 2.5.4: The upper part depicts a single batch of 16 512×512 pixel patches downscaled by factor 3. Due to our sampling strategy, 14 out of 16 examples contain the foreground class “glomerulus”. The lower plot depicts the corresponding pixel distribution after normalization. Colors indicate the respective channel (RGB).

(Yong et al. 2020) performed most consistently. In addition, we compared a variety of loss functions and chose to train our final models using a balanced dice cross-entropy loss. Moreover, we considered different magnifications of the training images. While a higher resolution might be beneficial to identify the glomerular border (bowman's capsule) correctly, a reduced resolution (equal to less magnification) provides more context for the annotated area. We compared a resolution reduction of factors 2, 3, 4, 6, and 8. The factor 3 resolution reduction resulted in the best dice score for a single model. Our data augmentation strategy comprises random rotation, flipping, deformation, brightness and contrast adjustments, desaturation, and contrast limited adaptive histogram equalization.

Inference. To achieve reliable predictions during test time, we combined several best practices such as overlapping tiles (shift factor 0.8), gaussian weighting (Isensee et al. 2021), pre-filtering of empty tiles, and test-time augmentation (horizontal and vertical flip). These "tricks" removed almost any prediction artifacts, such as half cut-off glomeruli or noise.

We additionally compared different softmax probability post-processing variants in our cross-validation experiment and compared them with a 0.5 threshold value as our baseline. Using a conditional random field had a negative impact on the dice score. We found no benefit in removing positive areas that were significantly smaller than the average glomeruli size. This is most likely due to the negligible amount of pixels affected compared with the total amount of positive pixels. We observed promising results when the softmax score was locally thresholded with Otsu's method. However, this did not improve the average dice when applied to all data. Thus, we did not use any post-processing in the final submission. However, we used post-processing to approximate better uncertainty scores based on the visual results during cross-validation. We decided to post-process areas with less than 10k pixel (at scale 2) with Otsu's method and "fill holes" for all other regions.

Competition results. The competition rules allow selecting two submissions for the final score on the private (undisclosed) test set. Our first submission was based on five models that were trained via cross-validation, and the image resolution was downscaled with factor 3. The second submission consisted

Table 2.5.1: Parameters for the best model ensemble.

Hyperparameter	Value
Architecture	U-Net with EfficientNet-b2 enoder
Pretraining	imagenet
Loss	Dice-CrossEntropy
Optimizer	ranger (max. learning rate 1e-3)
Batch size	16 with 512×512 tiles
Training iterations	3000
Ensembling	3 models trained on all data
Resolution downscaling	Factor 2,3, and 4

of only three models trained on all available data (training data and public test data with refined pseudo labels, see Section 2.5.4) but with different zoom scales (downscaling factors 2,3, and 4) . The latter approach produced an average dice score of 0.9485 on the private test set, ranking 10th on the final (private) leaderboard. The most important settings for our final model ensemble are specified in Table 2.5.1.

Detection performance on instance level. In addition to the pixel level evaluation (dice score) in the competition, we computed instance level (glomeruli level) metrics that account for the *detection* quality of our model. Here, we calculated M_{Recall}, $M_{Precision}$, and $M_{F1score}$ (see Section 3.2.2). The cross-validation results on instance level are depicted in Figure 2.5.5. The high recall indicates

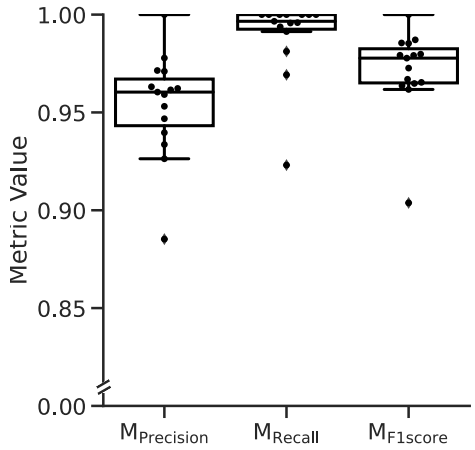


Figure 2.5.5: Cross-validation results with instance level metrics.

that the models detect almost all annotated glomeruli. The precision is slightly lower, which indicates that the models are sometimes misled by vessels or other tissue that look similar to a glomerulus.

2.5.4 Uncertainty Estimation

To robustly estimate the confidence of our prediction, we adapted the energy-based approach of Liu et al. (2020) for the image segmentation task. When applying a softmax prediction, neural networks often overestimate their confidence when predicting out of distribution data. Using the energy score can help to find false positives in such cases. Liu et al. (2020) define the free energy function as

$$E(x; f) := -T \cdot log(\sum_{i}^{K} e^{f_i(x)/T}), \tag{2.1}$$

where K is the number of classes, x the logits and f the function (here the neural network). As suggested by the authors, we chose the temperature parameter $T = 1$. To allow a more intuitive interpretation of the energy (which means generating mostly positive numbers), we calculated the *negative* energy score $-E$ in our experiments. Thus, our reported energy scores always describe the *negative* energy.

Figure 2.5.6 shows an example of a correctly detected glomerulus (true positive, top row) and a falsely detected glomerulus (false positive, bottom row). The predicted probability is similar in both examples. However, the true positive example exhibits a high mean energy score, while the false positive's mean energy score is relatively low. The mean energy score is defined as the average (pixel) energy for a single glomerulus instance.

Figure 2.5.7 summarizes the mean energy score values (based on cross-validation results) and confirms the visual impression from Figure 2.5.6. The values show a positive correlation between the intersection-over-union (IoU) metric[17] and the mean energy score. Low IoU values indicate a false positive. The results show that high energy values are a strong indicator for correct predictions. In turn, predicted instances with low energy values should be treated with caution.

[17] Similar to the dice score, the IoU accounts for the overlap of prediction and ground truth but is only used on instance level in this context; see Section 3.2.2 for a detailed description.

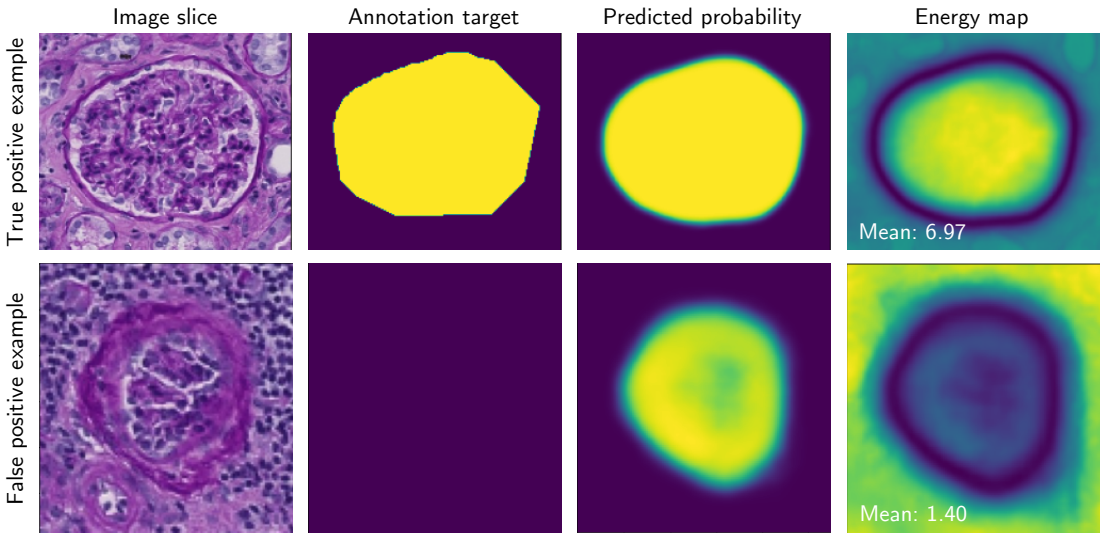


Figure 2.5.6: True positive and false positive examples. Colors indicate the respective values, from low (purple) to high (yellow). Probability and energy values are at different scales.

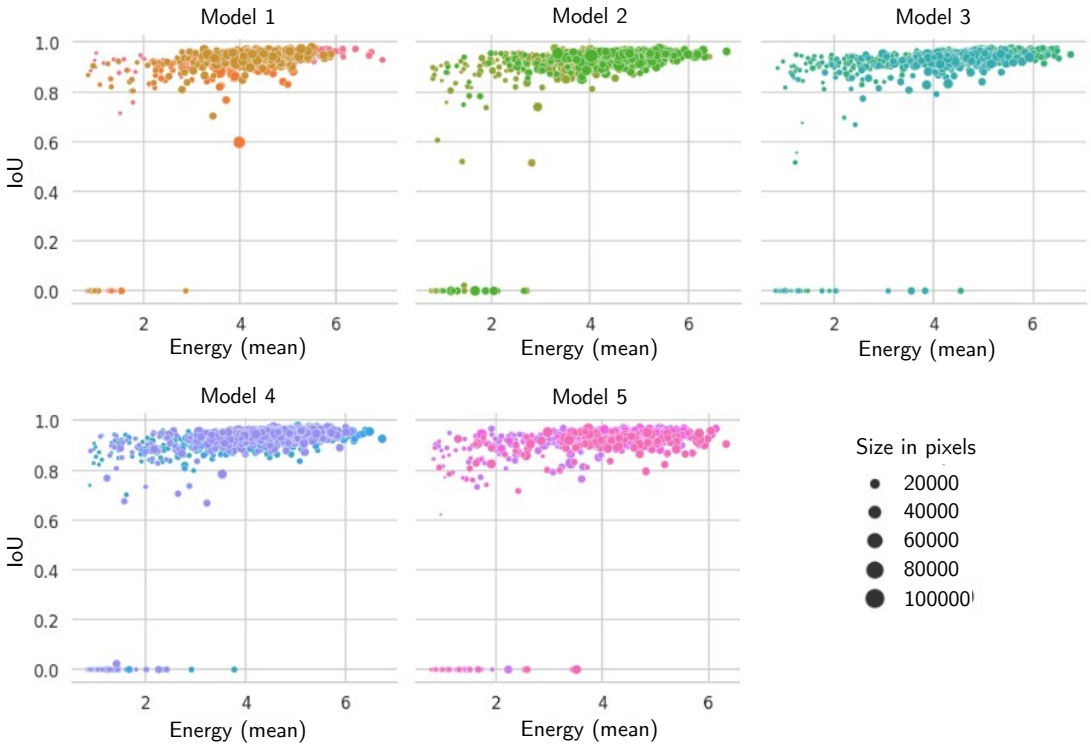


Figure 2.5.7: Cross-validation results of IoU scores and mean energy values. Colors indicate the different validation images.

Pseudo labels and annotation refinement. To extend our training data, we predicted labels on the publicly available test set as well as on the unused whole slide images published on the HuBMAP Portal[18]. The resulting predictions were manually refined by a physician in uncertain regions using *qupath* and a Wacom drawing tablet. The uncertainty was estimated by computing the energy score on the logits (Equation 2.5.4). Positive instances with a low energy score were explicitly reviewed. Glomeruli were excluded if more than half of their area was destroyed. The entire workflow is depicted in Figure 2.5.8.

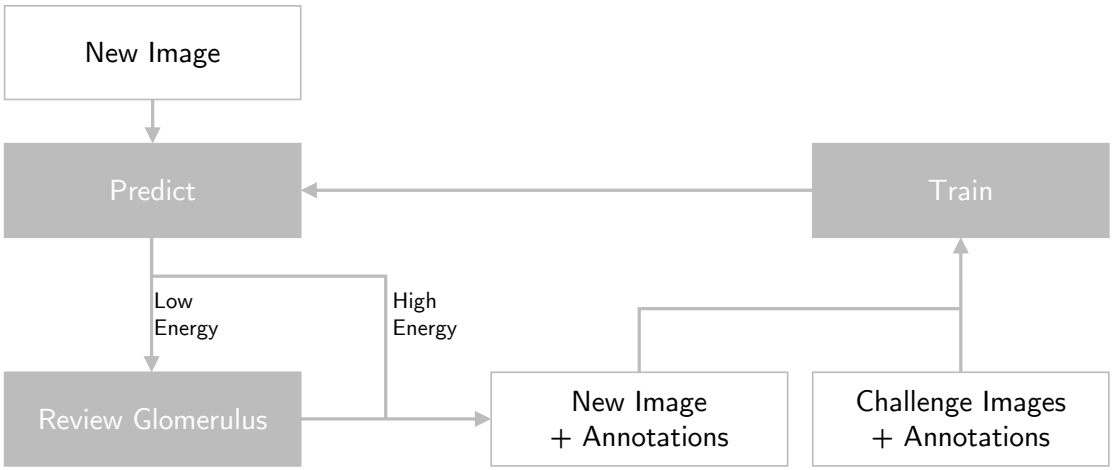


Figure 2.5.8: Proposed human-in-the-loop annotation refinement.

Reference glomeruli. Our approach also provides valuable insights to generate reference glomeruli for inclusion into a Human Reference Atlas (Figure 2.5.9). To identify typical glomeruli in an image, we utilize their energy score. A high energy score helps to locate typical and artifact-free glomeruli on a whole slide image.

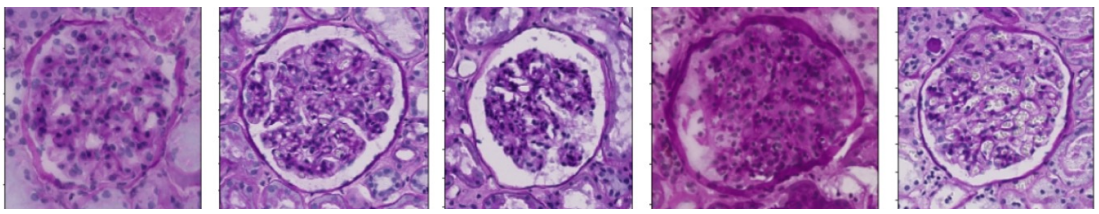


Figure 2.5.9: Proposed reference glomeruli with high energy scores.

[18] `https://portal.hubmapconsortium.org/`

2.5.5 Generalizability

To test the generalizability of our approach, we applied our pipeline to tissue images from another organ, the pancreas.

Data. Similar to the glomeruli in the challenge data, pancreatic islets are FTUs of the pancreas. They function as the endocrinologic system of the pancreas and produce insulin. The cells of the pancreatic islets are destroyed in cases of diabetes type 1. A major difference to the challenge data is that the slides are stained with hematoxylin and eosin and not PAS. We annotated pancreatic islets in three whole slice images in the same way as the challenge data. Figure 2.5.10 shows an exemplary image and the corresponding annotation data from the pancreas dataset.

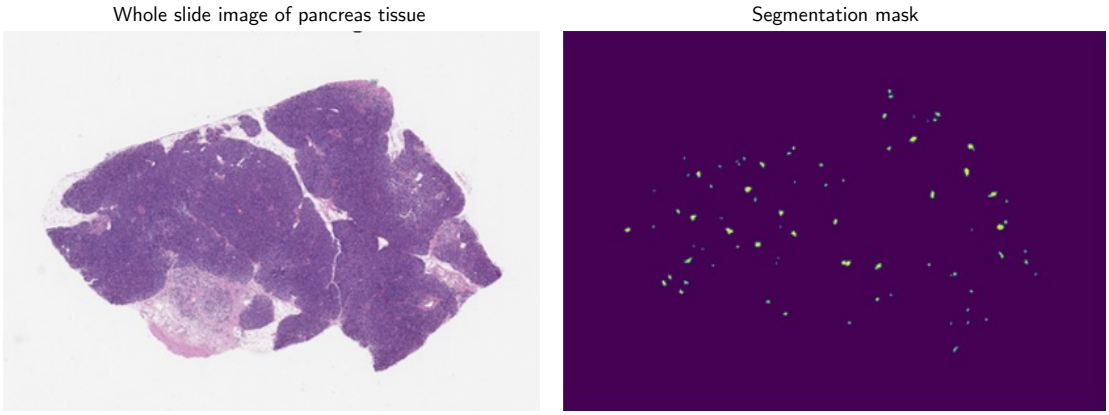


Figure 2.5.10: Exemplary image and segmentation mask from the pancreas dataset.

Figure 2.5.11 shows a zoom-in on the manual annotation (green line) of a pancreatic islet. The data was downloaded from The Cancer Imaging Archive[19] and only cancer-free tissue was included. Again, our notebooks for the segmentation of the pancreatic islets follow the workflow described in Section 2.5.1 and an overview is provided in Table B.2.

Training and evaluation. We trained the models using a three-fold cross-validation approach with the same model architecture and training routine

[19] `https://www.cancerimagingarchive.net/`

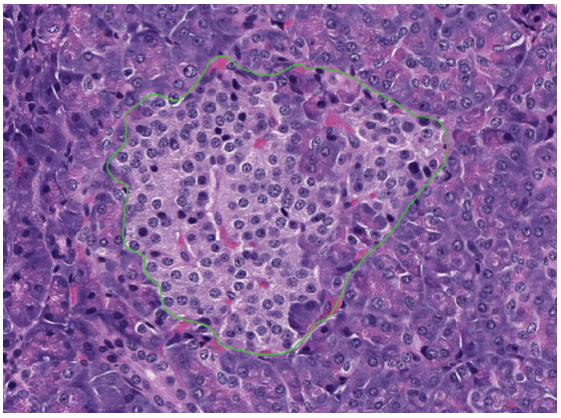


Figure 2.5.11: Manual annotation of a pancreatic islet.

as for the competition data. We only adjusted the mean and standard deviation of the image data. Even though the training was performed on only two images and validation on only one image, the cross-validation results (Figure 2.5.12) are remarkably stable and indicate high predictive performance.

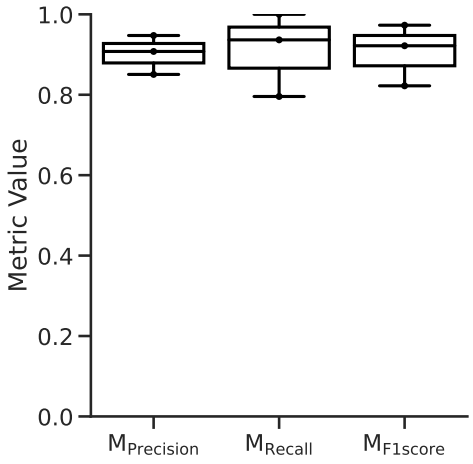


Figure 2.5.12: Cross-validation results (instance level) on the pancreas dataset.

Uncertainty estimation. We used the same code for validation and uncertainty estimation for the pancreas data. Figure 2.5.13 depicts exemplary predictions. The visual impression is very similar to the impression in the competition dataset (Figure 2.5.6). The positive correlation between the mean energy and the detection performance is confirmed in Figure 2.5.14.

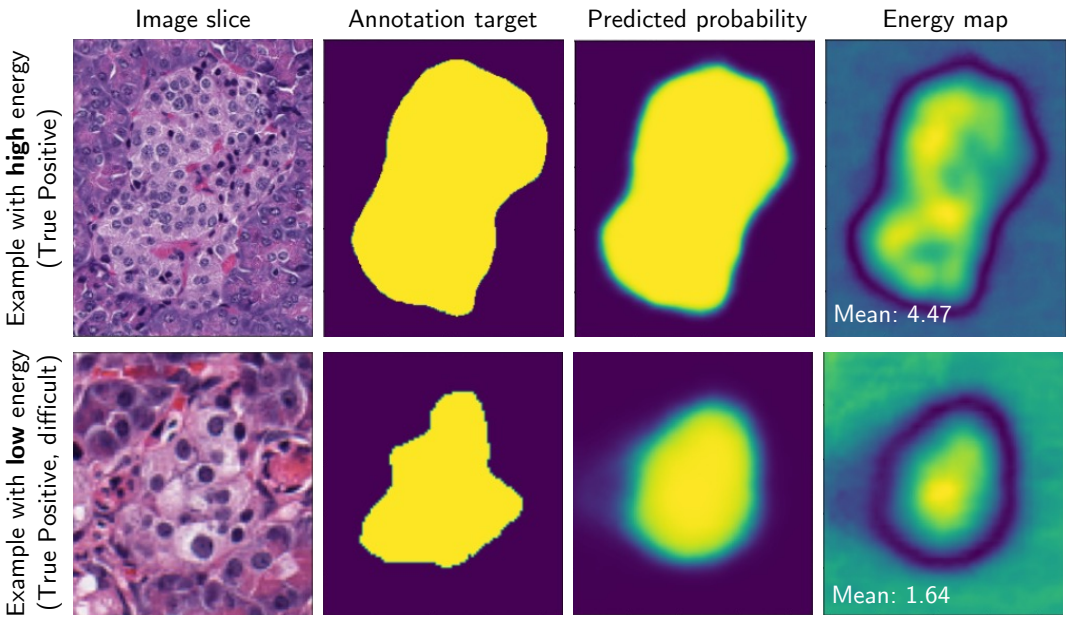


Figure 2.5.13: True positive examples from the pancreas dataset. Colors indicate the respective values, from low (purple) to high (yellow). Probability and energy values are at different scales.

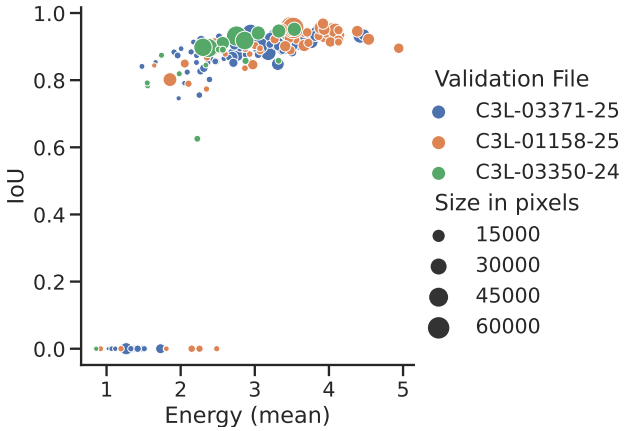


Figure 2.5.14: Cross-validation results of IoU scores and mean energy values from the pancreas dataset.

2.5.6 Limitations

Since our models are not trained on the entire data, it is possible that rare artifacts like small air bubbles are not seen during training and get misclassified by the model for glomeruli. These regions have a lower energy score than normal glomeruli and can easily be found with human-in-the-loop qual-

ity control. When retraining the model, these regions can be upsampled as well. Furthermore, we wanted to keep our solution as simple as possible and thus refrained from stacking different architectures or differently trained models with, for instance, deviating loss functions.

2.5.7 Prototype Summary

This section describes the development of two DL prototypes for the semantic segmentation of tissue in whole slide images. The model agnostic sampling strategy enables fast and reliable model training on standard GPUs. Moreover, the energy-based uncertainty scores facilitate a semi-automated annotation to create more training data painlessly. The first prototype detects glomeruli in human kidney tissue images and has passed through many development iterations during the Kaggle challenge. The second prototype detects Langerhans islets in pancreas tissue images and has only passed through the prototyping cycle once, reusing the settings from the first prototype. The detection and segmentation performance of the presented prototypes is auspicious. However, their application in medical research would require further evaluation regarding their reliability and robustness.

2.6 Discussion

In this chapter, I present DL prototypes addressing different problems from different domains. The prototypes are all based on DL architectures with convolutional layers. The models for biomedical image segmentation (fluorescent neurons in microscopy images, Section 2.1; glomeruli in human kidney tissue images, Section 2.5) both leverage variants of the U-Net architecture. The detection engine for the segmentation of fashion images (Section 2.3) is based on the Mask-RCNN architecture. The DL system for classifying structure-borne noise signals (Section 2.2) uses a typical CNN architecture for sound classification. The prototypes for the enhancement of the fashion design process (Section 2.4.3) leverage CNN architectures for feature extraction (CSNs for fashion similarity) and generative processes (DC-GAN for T-Shirt design, BigGAN for interpolation between different visual concepts). The CNN architectures were

selected simply because they represented the best available model architecture when the prototype was developed. Thus, these architectures only reflect a snapshot of the current developments.

Having passed one or more prototyping iterations, the presented DL solutions exhibit different maturity levels. Even though the development of some prototypes is already at an advanced stage, they have never been productionized or reached another phase of the ML life cycle. This status is to some extent due to the academic environment. Yet it also illustrates the complexity of DL model development and why the majority of promising DL projects never get beyond the piloting phase (see Section 1). To investigate the challenges of the ML life cycle phases beyond prototyping, I continue with the development of the biomedical image segmentation project from Section 2.1 in the following chapters.

3 On the Objectivity, Reliability, and Validity of Deep Learning enabled Bioimage Analyses

This chapter is adapted from the article of Segebarth, D., Griebel, M., Stein, N., et al. *On the objectivity, reliability, and validity of deep learning enabled bioimage analyses* published in Elife (9) 2020[20]. All data can be accessed via Dryad[21]. The source code is available on GitHub[22]. Please refer to the original article for detailed information on animal experiments and data acquisition.

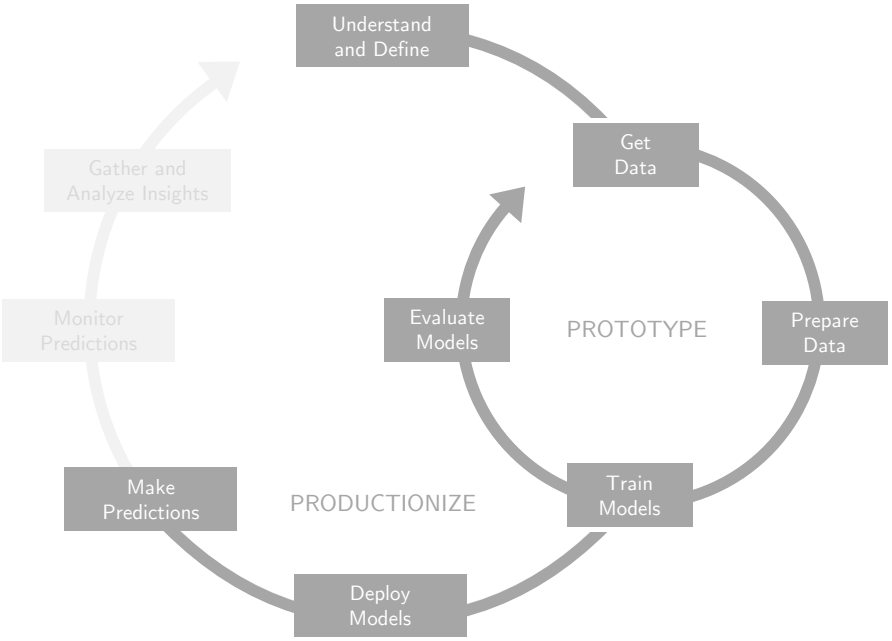
This chapter conceptually covers all phases of the ML life cycle (Section 1.2) until *Make Predictions*. Thereby, it follows the typical structure of a life science article (Introduction, Methods, Results, Discussion). Section 3.1 introduces the primary concepts of this chapter – objectivity, reliability, and validity – and foremost constitutes the *Understand and Define* phase. Section 3.2 provides the methodological details from data preparation until the statistical evaluation of the predictions. Section 3.3 covers the results of the *Prototyping* phase as well as the resulting predictions once the model is deployed. Finally, Section 3.4 discusses the results and limitations.

Summary. Bioimage analysis of fluorescent labels is widely used in the life sciences. Recent advances in DL allow automating time-consuming manual im-

[20] D. Segebarth and M. Griebel contributed equally. Thereby, D. Segebarth was primarily responsible for the biological aspects of the paper. M. Griebel was responsible for DL model development and evaluation.

[21] `www.doi.org/10.5061/dryad.4b8gtht9d`

[22] `www.github.com/matjesg/bioimage_analysis`



Machine learning life cycle stages covered in Chapter 3.

age analysis processes based on annotated training data. However, manual annotation of fluorescent features with a low signal-to-noise ratio is somewhat subjective. Training DL models on subjective annotations may be unstable or yield biased models. In turn, these models may be unable to reliably detect biological effects. An analysis pipeline integrating data annotation, ground truth estimation, and model training can mitigate this risk. To evaluate this integrated process, we compare different DL-based analysis approaches. With data from two model organisms and five laboratories, we show that ground truth estimation from multiple human annotators helps to establish objectivity in fluorescent feature annotations. Furthermore, ensembles of multiple models trained on the estimated ground truth establish reliability and validity. Our research provides guidelines for reproducible DL-based bioimage analyses.

3.1 Introduction

Modern microscopy methods enable researchers to capture images that describe cellular and molecular features in biological samples at an unprecedented scale. One of the most frequently used imaging methods is fluores-

cent labeling of biological macromolecules, both *in vitro* and *in vivo*. In order to test a biological hypothesis, fluorescent features have to be interpreted and analyzed quantitatively, a process known as bioimage analysis (Meijering et al. 2016). However, fluorescence does not provide clear signal-to-noise borders, forcing human experts to utilize individual heuristic criteria, such as morphology, size, or signal intensity to classify fluorescent signals as background, or to, often manually, annotate them as a region of interest (ROI). This cognitive decision process depends on the graphical perception capabilities of the individual annotator (Cleveland and McGill 1985). Constant technological advances in fluorescence microscopy facilitate the automatized acquisition of large image datasets, even at high resolution and with high throughput (Li et al. 2010; McDole et al. 2018; Osten and Margrie 2013). The ever-increasing workload associated with image feature annotation therefore calls for computer-aided automated bioimage analysis. However, attempts to replace human experts and to automate the annotation process using traditional image thresholding techniques (e.g., histogram shape-, entropy-, or clustering-based methods; Sezgin and Sankur 2004) frequently lack flexibility, as they rely on a high signal-to-noise ratio in the images or require computational expertise for user-based adaptation to individual datasets (Chamier, Laine, and Henriques 2019). In recent years, DL and in particular deep convolutional neural networks have shown remarkable capacities in image recognition tasks, opening new possibilities to perform automatized image analysis. DL-based approaches have emerged as an alternative to conventional feature annotation or segmentation methods (Caicedo et al. 2019) and are even capable of performing complex tasks such as artificial labeling of plain bright-field images (Chamier, Laine, and Henriques 2019; Christiansen et al. 2018; Ounkomol et al. 2018). The main difference between conventional and DL algorithms is that conventional algorithms follow predefined rules (hard-coded), while DL algorithms are flexible to learn the respective task on the basis of a training dataset (LeCun, Bengio, and Hinton 2015). Yet, the deployment of DL approaches necessitates both computational expertise and suitable computing resources. These requirements frequently prevent non-AI experts from applying DL to routine image analysis tasks. Initial efforts have already been made to break down these barriers, for instance, by integration into prevalent bioimaging tools such as *ImageJ* (Falk et al. 2019) and *CellProfiler* (McQuin et al. 2018), or using cloud-

based approaches (Haberl et al. 2018). To harness the potentials of these DL-based methods they require integration into the bioimage analysis pipeline. We argue that such integration into the scientific process ultimately necessitates DL-based approaches to meet the same standards as any method in an empirical study. We can derive these standards from the general quality criteria of qualitative and quantitative research: objectivity, reliability, and validity (Frambach, Vleuten, and Durning 2013).

Objectivity refers to the neutrality of evidence, with the aim to reduce personal preferences, emotions, or simply limitations introduced by the context in which manual feature annotation is performed (Frambach, Vleuten, and Durning 2013). Manual annotation of fluorescent features has long been known to be subjective, especially in the case of weak signal-to-noise thresholds (Schmitz, Korr, and Heinsen 1999; Collier et al. 2003; Niedworok et al. 2016). Notably, there is no objective ground truth reference in the particular case of fluorescent label segmentation, causing a critical problem for the training and evaluation of DL algorithms. As multiple studies have pointed out that annotations of low quality can cause DL algorithms to either fail to train or to reproduce inconsistent annotations on new data (Chamier, Laine, and Henriques 2019; Falk et al. 2019) this is a crucial obstacle for applying DL to bioimage analysis processes.

Reliability is concerned with the consistency of evidence (Frambach, Vleuten, and Durning 2013). To allow an unambiguous understanding of this concept, we further distinguish between repeatability and reproducibility. Repeatability or test-retest reliability is defined as "closeness of the agreement between the results of successive measurements of the same measure and carried out under the same conditions" (Taylor and Kuyatt 1994, 14), which is guaranteed for any deterministic DL model. Reproducibility, on the other hand, is specified as "closeness of the agreement between the results of measurements of the same measure and carried out under changed conditions" (Taylor and Kuyatt 1994, 14), e.g., different observer, or different apparatus. This is a critical point since the output of different DL models trained on the same training dataset can vary significantly. This is caused by the stochastic training procedure (e.g., random initialization, random sampling, and data augmentation), the choice of model parameters (e.g., model architecture, weights, activation functions), and the choice of hyperparameters (e.g., learning rate, mini-batch

size, training epochs). Consequently, the reproducibility of DL models merits careful investigation.

Finally, *validity* relates to the truth value of evidence, i.e, whether we in fact measured what we intended to. Moreover, validity implies reliability - but not vice versa (Frambach, Vleuten, and Durning 2013). On a basis of a given ground truth, validity is typically measured using appropriate similarity measures such as F1 score for detection and Intersection over Union (IoU) for segmentation purposes (Ronneberger, Fischer, and Brox 2015; Falk et al. 2019; Caicedo et al. 2019). In addition, the deep learning community has established widely accepted standards for training models. These comprise, among other things, techniques to avoid overfitting (regularization techniques and cross-validation), tuning hyperparameters, and selecting appropriate metrics for model evaluation. However, these standards do not apply for the training and evaluation of a DL model in the absence of a ground truth, like in the case of fluorescent features.

Taken together and with regard to the discussion about a reproducibility crisis in the fields of biology, medicine, and artificial intelligence (Siebert, Machesky, and Insall 2015; Baker 2016; Ioannidis 2016; Hutson 2018; Fanelli 2018; Chen et al. 2019), these limitations indicate that DL could aggravate this crisis by adding even more unknowns and uncertainties to bioimage analyses.

However, the present study asks whether DL, if instantiated in an appropriate manner, also holds the potential to instead enhance the objectivity, reproducibility, and validity of bioimage analysis. To tackle this conundrum, we investigated different DL-based strategies on five fluorescence image datasets. We show that training of DL models on the pooled input of multiple human experts utilizing ground truth estimation (consensus models) increases the objectivity of fluorescent feature segmentation. Furthermore, we demonstrate that ensembles of consensus models are even capable of enhancing the reliability and validity of bioimage analysis of ambiguous image data, such as fluorescence features in histological tissue sections.

3.2 Methods

This section covers the topics of ground truth estimation (Section 3.2.1), evaluation metrics (Section 3.2.2), details on the DL approach (Section 3.2.3), quantification of fluorescent features (Section 3.2.4), and statistical analysis (Section 3.2.5).

3.2.1 Ground Truth Estimation

In absence of an objective ground truth, we derived a probabilistic estimate of the ground truth by running the expectation-maximization algorithm for simultaneous truth and performance level estimation (STAPLE, Warfield, Zou, and Wells 2004). The STAPLE algorithm iteratively estimates the ground truth segmentation (est. GT) based on the expert segmentation maps. During each algorithm iteration 2 steps are performed:

1. **Estimation step** The ground truth segmentation's conditional probability is estimated based on the expert decisions and previous performance parameter estimates.
2. **Maximization step** The performance parameters (sensitivity and specificity) for each expert segmentation are estimated by maximizing the conditional expectation.

Iterations are repeated until convergence is reached. We implemented the algorithm using the Simplified interface to the Insight Toolkit (SimpleITK 1.2.4, Lowekamp et al. 2013).

3.2.2 Evaluation Metrics

All evaluation metrics were calculated using Python (version 3.7.3), SciPy (version 1.4.1), and scikit-image (version 0.16.2).

Segmentation and detection

Following Caicedo et al. (2019) we based our evaluation on identifying segmentation and detection similarities on object-level (ROI-level). In a segmentation

mask, we define an object as a set of pixels that were horizontally, vertically, and diagonally connected (8-connectivity). We only considered ROIs at a biologically justifiable size, depending on the data set characteristics. We approximated the minimum size based on the smallest area that was annotated by a human expert (*Lab-Mue*: 30px, *Lab-Inns1*: 16px, *Lab-Inns2*: 60px, *Lab-Wue1*: 30px, *Lab-Wue2*: 112px).

To compare the segmentation similarity between a source and a target segmentation mask, we first computed the intersection-over-union (IoU) score for all pairs of objects. The IoU, also known as Jaccard similarity, of two sets of pixels $a = \{1, ..., A\}$ and $b = \{1, ..., B\}$ is defined as the size of the intersection divided by the size of the union:

$$M_{\text{IoU}}(a, b) := \frac{|a \cap b|}{|a \cup b|} \tag{3.1}$$

Second, we used the pairwise IoUs to match the objects of each mask. We solved the assignment problem by maximizing the sum of IoUs by means of the Hungarian Method (Kuhn 1955). This ensures an optimal matching of objects in the case of ambiguity, i.e., an overlap of one source object with one or more targets object. We reported the segmentation similarity of two segmentation masks by calculating the arithmetic mean of M_{IoU} over all matching objects:

$$\bar{M}_{\text{IoU}} = \frac{1}{N} \sum_{i=1}^{N} M^i_{\text{IoU}} \tag{3.2}$$

where $i \in \{0, ..., N\}$ is an assigned match and N denotes the number of matching objects. By this definition, the *Mean IoU* only serves as a measure for the segmentation similarity of matching objects and neglects objects that do not overlap at all.

To address this issue, we additionally calculated measures to account for the detection similarity. Therefore, we define a pair of objects with an IoU is above a threshold t as correctly detected (true positive - TP). Objects that match with an IoU at or below t or have no match at all are considered to be false negative (FN) for the source mask and false positive (FP) for the target mask. This allows us to calculate the Precision $M_{\text{Precision}}$, Recall M_{Recall}, and F1

score $M_{\text{F1 score}}$ as the harmonic mean of $M_{\text{Precision}}$ and M_{Recall}:

$$M_{\text{Precision}}(t) := \frac{TP(t)}{TP(t) + FP(t)} \tag{3.3}$$

$$M_{\text{Recall}}(t) := \frac{TP(t)}{TP(t) + FN(t)} \tag{3.4}$$

$$M_{\text{F1 score}}(t) := 2 \cdot \frac{\mathrm{M}_{\text{Precision}}(\mathrm{t}) \cdot \mathrm{M}_{\text{Recall}}(\mathrm{t})}{\mathrm{M}_{\text{Precision}}(\mathrm{t}) + \mathrm{M}_{\text{Recall}}(\mathrm{t})} \tag{3.5}$$

with $t \in [0, 1]$ as a fixed IoU threshold. If not indicated differently, we used $t = 0.5$ in our calculations.

Inter-rater reliability

To quantify the reliability of agreement between different annotators we calculated Fleiss' κ (Fleiss and Cohen 1973). In contrast to the previously introduced metrics, Fleiss' κ accounts for the agreement that would be expected by chance. For a collection of segmentation masks of the same image, each object (ROI) $i \in \{1, ..., N\}$ is assigned to a class $j \in \{0, ..., K\}$. Here, N denotes the total number of unique objects (ROIs) and K the number of categories ($K = 1$ for binary segmentation). Then, n_{ij} represents the number of annotators who assigned object i to class j. We again leveraged the IoU metric to match the ROIs from different segmentation masks above a given threshold $t \in [0, 1]$. Following Fleiss and Cohen (1973) we define the proportion of all assignments for each class:

$$p_j(t) := \frac{1}{Nd} \sum_{i=1}^{N} n_{ij}(t) \tag{3.6}$$

where d denotes the count of the annotators. We define the extent to which the annotators agree on the i-th object as

$$P_i(t) := \frac{1}{d(d-1)} \sum_{j=1}^{K} n_{ij}(t) \left(n_{ij}(t) - 1\right) \tag{3.7}$$

Subsequently, we define the mean of the $P_i(t)$ as

$$\bar{P}(t) := \frac{1}{N} \sum_{i=1}^{N} P_i(t) \tag{3.8}$$

and

$$\bar{P}_e(t) := \sum_{j=1}^{K} p_j(t)^2 \tag{3.9}$$

Finally, Fleiss' κ at a given threshold t is defined as

$$\kappa(t) := \frac{\bar{P}(t) - \bar{P}_e(t)}{1 - \bar{P}_e(t)} \tag{3.10}$$

where $1 - \bar{P}_e(t)$ denotes the degree of agreement attainable above chance and $\bar{P}(t) - \bar{P}_e(t)$ the actually achieved agreement in excess of chance. To allow a better estimate of the chance we randomly added region proposals of class $j = 0$ (background). If not indicated differently, we use $t = 0.5$ in our calculations.

3.2.3 Deep Learning Approach

The deep learning pipeline was implemented in Python (version 3.7.3), TensorFlow (version 1.14.0), Keras (version 2.2.4), scikit-image (version 0.16.2), and scikit-learn (version 0.21.2).

Network Architecture

We instantiated all DL models with a U-Net architecture (Ronneberger, Fischer, and Brox 2015), a fully convolutional neural network for semantic segmentation. The key principle of a U-Net is that one computational path stays at the original scale, preserving the spatial information for the output, while the other computational path learns the specific features necessary for classification by applying convolutional filters and thus condensing information (Ronneberger, Fischer, and Brox 2015). We adopted the model hyperparameters (e.g., hidden layers, activation functions, weight initialization) from Falk et al. (2019) as these are extensively tested and evaluated on different biomedical data sets. The layers of the U-Net architecture are logically grouped into an encoder and a decoder.Following Falk et al. (2019) the VGG-like encoder consists of five convolutional modules. Each module comprises two convolution layers with no padding, each followed by a leaky ReLU with a leakage factor of 0.1 and a max-pooling operation with a stride of two. The last module, how-

ever, does not contain the max-pooling layer and constitutes the origin of the decoder. The decoder consists of four (up-) convolutional modules. Each of these modules comprises a transposed convolution layer (also called up- or deconvolution), a concatenate layer for the corresponding cropped encoder feature map, and two convolution layers. Again, each layer is followed by a leaky ReLU with a leakage factor of 0.1. The final layer consists of a 1 x 1 convolution with a softmax activation function. The resulting (pseudo-) probabilities allow a comparison to the target segmentation mask using cross-entropy on pixel level. Unless indicated differently, we used a kernel size of 3 x 3. To allow faster convergence during training we included batch normalization layers (Ioffe and Szegedy 2015) after all (up-) convolutions below the first level. By this, an unnormalized path from the input features to the output is remaining to account for absolute input values, e.g., the brightness of fluorescent labels.

Weighted soft-max cross-entropy loss

Fluorescent microscopy images typically exhibit more background than fluorescent features of interest. To control the impacts of the resulting class imbalance we implemented a pixel-weighted softmax cross-entropy loss. Thus, we compute the loss from the raw score (logits) of the last 1x1 convolution without applying the softmax. As proposed by Falk et al. (2019) we define the weighted cross-entropy loss for an input image I as

$$L_{\mathsf{wce}}(I) := -\sum_{x\in\Omega} w(x)\log\frac{\exp\left(\hat{y}_{y(x)}(x)\right)}{\sum_{k=0}^{K}\exp\left(\hat{y}_k(x)\right)} \tag{3.11}$$

where x is a pixel in image domain Ω, $w:\Omega\rightarrow\mathbb{R}_{\geqslant 0}$ the pixel-wise weight map, $y:\Omega\rightarrow\{0,...,K\}$ the target segmentation mask, $\hat{y}_k:\Omega\rightarrow\mathbb{R}$ the predicted score for class $k\in\{0,...,K\}$, and K the number of classes ($K=1$ for binary classification). Consequently, $\hat{y}_{y(x)}(x)$ is the predicted score for the target class $y(x)$ at position x.

Similar to Falk et al. (2019) we compose the weight map w from two different weight maps w_{bal} and w_{sep}. The former allows mitigating the class imbalances by decreasing the weight of background pixels by the factor $v_{\mathsf{bal}}\in[0,1]$. We add a smoothly decreasing Gaussian function at the edges of the foreground

objects accordingly and define

$$w_{bal}(x) := \begin{cases} 1 & y(x) > 0 \\ v_{\text{bal}} + (1 - v_{bal}) \exp\left(-\frac{d_1^2(x)}{2\sigma_{\text{bal}}^2}\right) & y(x) = 0 \end{cases} \tag{3.12}$$

where $d_1(x)$ denotes the distance to the closest foreground object and σ_{bal} the standard deviation of the Gaussian function.

By definition, semantic segmentation performs a pixel-wise classification and is unaware of different object instances (ROIs). Following Falk et al. (2019) we the force learning of the different instances by increasing the weight of the separating ridges. We estimate the width of a ridge by adding d_1 (distance to nearest ROI) and d_2 (distance to second nearest ROI) at each pixel. We define

$$w_{\text{sep}}(x) := \exp\left(-\frac{(d_1(x) + d_2(x))^2}{2\sigma_{\text{sep}}^2}\right) \tag{3.13}$$

where σ_{sep} defines the standard deviation of the decreasing Gaussian function. The final weight map is given by

$$w := w_{\text{bal}} + \lambda w_{\text{sep}} \tag{3.14}$$

where $\lambda \in \mathbb{R}_{\geqslant 0}$ allows to control the focus on instance separation. We used the following parameter set in our experiments: $\lambda = 50$, $v_{bal} = 0.1$, $\sigma_{\text{bal}} = 10$ and $\sigma_{\text{sep}} = 6$.

Tile sampling and augmentation

Given limited training data availability, we leveraged effective data augmentation techniques for biomedical images as proposed by (Falk et al. 2019). These comprise transformations and elastic deformations by means of a random deformation field. To become invariant to the input sizes (image shapes) we leveraged the overlap tile strategy introduced by (Ronneberger, Fischer, and Brox 2015). Thus, images of any size can be processed. Both data augmentation and overlap tile strategy were adopted from a TensorFlow implementation of (Falk et al. 2019). We used an input tile shape of 540×540×1 (height×width×channels) and a corresponding output tile shape of 356×356×1 for all our experiments.

Training, evaluation, and model selection

We trained, evaluated, and selected all deep learning models for our different strategies – *expert models*, *consensus models*, *consensus ensembles* – following the same steps:

1. Determining an appropriate learning rate using the *learning rate finder* (Smith 2018)
2. Splitting the data into train and validation set (random stratified sampling)
3. Training the model on the train set according to the *fit-one-cycle* policy of Smith (2018)
4. Selecting the model with the highest $M_{\text{F1 score}}$ median on the validation set (post-hoc evaluation).

We used the annotations from individual experts to train the *expert models* and the consensus annotations (est. GT) for the *consensus models* and *consensus ensembles*. The post-hoc evaluation on the validation set was performed using the saved model weights (checkpoints) from each epoch. For the similarity analysis, we converted the model output (pixel-wise softmax score) to a segmentation mask by assigning each pixel to the class with the highest softmax score. For the *consensus ensemble* approach, we repeated the steps above according to the principle of *k-fold cross-validation*. We ensembled the resulting k models by averaging the softmax predictions.

Our initial experimental results have indicated that an adequately trained DL-model performs on par with a human expert. However, insufficient training data may impair the model performance. As there were only five annotated training images for the external laboratories (*Lab-Mue, Lab-Inns1, Lab-Inns2*, and *Lab-Wue2*), we additionally defined a model selection criterion to establish trust in our consensus ensemble approaches: A selected consensus model must at least match the performance of the "worst" human expert for each validation image (measured as the $M_{\text{F1 score}}$ to the estimated ground truth). This selection criterion serves as a lower bound for individual model performance. In those cases where the criterion discarded models, the issue was typically due to a validation image being very different from the training data for a given

train-validation split. This issue was often resolved when pretrained model weights were used. For the *frozen* approach (see 3.2.3) the models never met the selection criterion. Yet, we decided to retain these models to facilitate a comparison among the different approaches. We also indicated that these models and ensembles should be considered with caution and did not use them for further biological analyses.

We trained all models on an NVIDIA GeForce GTX 1080 TI with 11 GB GDDR5X RAM using the Adam optimizer (Kingma and Ba 2015) and a mini-batch size of four. If not indicated differently, the initial weights were drawn from a truncated normal distribution (He et al. 2015). We chose the appropriate maximum learning rates according to the learning rate finder (step two). For *Lab-Wue1* we selected a maximum learning rate of 4e-4 and a minimum learning rate of 4e-5 over a training cycle length of 972 iterations within $k = 4$ validation splits. For *Lab-Mue, Lab-Inns1, Lab-Inns2*, and *Lab-Wue2* we chose a maximum learning rate of 1e-4 and a minimum learning rate of 1e-5 over a training cycle length of 972 iterations within $k = 5$ validation splits.

Transfer Learning

To implement transfer learning we adapted the training procedure from above. For the *fine-tuning* approach, we initialized the weights from the *consensus models* of *Lab-Wue1* and performed all steps for model training, evaluation, and selection. For the *frozen* approach we also initialized the weights from the *consensus models* of *Lab-Wue1* but skipped steps two (finding a learning rate) and three (model training). Hence, we did not adjust the model weights to the new training data. Hardware and training hyperparameters remained unchanged.

3.2.4 Quantification of Fluorescent Features

Fluorescent features were analyzed on the base of the binary segmentation masks derived from the output of DL models or model ensembles or counted manually by lab-specific experts. In order to compare the number of fluorescent features across images, we normalized in each image the number of annotated fluorescent features to the area of the analyzed region (e.g. the num-

ber of cFOS-positive features per NeuN-positive area for *Lab-Wue1*). For one set of experiments, we pooled this data for each condition (e.g. H, C- and C+ for *Lab-Wue1*) and the analyzed brain region (e.g. whole DG, infrapyramidal DG, suprapyramidal DG, CA3, or CA1 for *Lab-Wue1*). To compare different sets of experiments with each other, we normalized all relative fluorescent feature counts to the mean value of the respective control group (e.g. H for *Lab-Wue1*).

The mean signal intensity for each image was calculated as the mean signal intensity of all ROIs annotated within the analyzed NeuN-positive region (only performed for *Lab-Wue1*). Subsequent pooling steps were identical as described above for the count of fluorescent features.

3.2.5 Statistical Analysis

All statistical analyses were performed using Python (version 3.7.3), SciPy (version 1.4.1), and Pingouin (version 0.3.4). In box plots, the area of the box represents the interquartile range (IQR, 1^{st} to 3^{rd} quartile) and whiskers extend to the maximal or minimal values, but no longer than $1.5 \times$ IQR.

Statistical analysis of fluorescent feature quantifications

All DL-based quantifications of fluorescent features were tested for significant outliers (Grubb's test). If an image was detected as a significant outlier in several DL-based quantification results, it was visually inspected by an expert and excluded from the analysis if abnormalities (e.g. clusters of fluorescent particles or folding of the tissue) were detected. Throughout all bioimage analyses, N represents the number of investigated animals and n the number of analyzed images. Normality (Shapiro-Wilk) and homogeneity of variance (Levenes) were tested for all DL-based quantification results. For the comparison of multiple quantifications of the same image dataset, non-parametric statistical tests were applied to all bioimage analyses. This ensured comparability of the results. To compare two experimental conditions (*Lab-Mue, Lab-Inns1,* and *Lab-Wue2*), Mann-Whitney-U tests were used. In case of three experimental conditions (*Lab-Wue1* and *Lab-Inns2*), Kruskal-Wallis-ANOVA followed by Mann-Whitney-U tests with Bonferroni correction for multiple comparisons was applied.

Effect size calculation

Effect sizes (η^2) were calculated for each pairwise comparison. First, the Z-statistic (Z) was calculated from the U-statistic (U) of the Mann-Whitney-U test as:

$$Z = \frac{U - \frac{n_1 \cdot n_2}{2}}{\sqrt{\frac{n_1 \cdot n_2 \cdot (n_1 + n_2 + 1)}{12}}} \tag{3.15}$$

where n_1 and n_2 are the numbers of analyzed images of the two compared groups, group 1 and group 2, respectively. Following Rosenthal and DiMatteo (2002), η^2 was calculated as:

$$\eta^2 = \frac{Z^2}{n_1 + n_2} \tag{3.16}$$

Furthermore, the three critical values of η^2 that mark the borders between the four significance levels (e.g. for p = 0.05, p = 0.01, and p = 0.001 for a pairwise comparison without Bonferroni correction for multiple comparisons) were calculated from the chi-square distribution.

All other statistical analyses

Data were tested for normal distribution (Shapiro-Wilk) and homoscedasticity (Levenes) and parametric or non-parametric tests were used accordingly, as reported in the figure legends (parametric: one-way ANOVA, followed by T-tests (or Welchs T-test for unequal sample sizes) with Bonferroni correction for multiple comparisons; non-parametric: Kruskal-Wallis ANOVA, followed by Mann-Whitney tests with Bonferroni correction for multiple comparisons).

3.3 Results

To evaluate the impact of DL on bioimage analysis results, we instantiated three exemplary DL-based strategies (Figure 3.1; strategies color-coded in gray, blue, and orange) and investigate them in terms of objectivity, reliability, and validity of fluorescent feature annotation. The first strategy, *expert models* (gray), reflects mere automation of the annotation process of fluorescent fea-

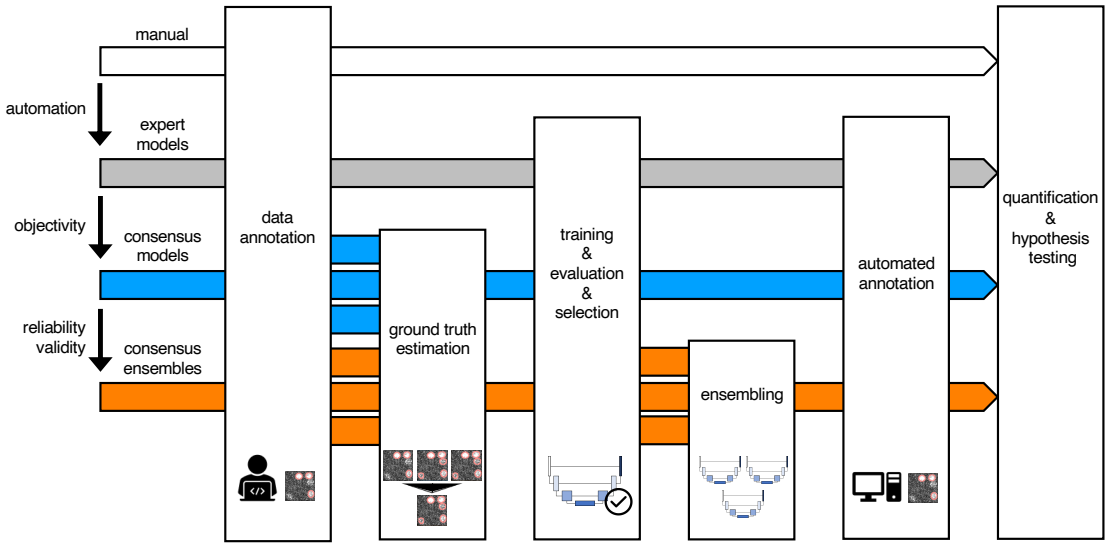


Figure 3.1: Schematic illustration of bioimage analysis strategies and corresponding hypotheses. Four bioimage analysis strategies are depicted. Manual (white) refers to manual, heuristic fluorescent feature annotation by a human expert. The three DL-based strategies for automatized fluorescent feature annotation are based on expert models (gray), consensus models (blue), and consensus ensembles (orange). For all DL-based strategies, a representative subset of microscopy images is annotated by human experts. Here, we depict labels of cFOS-positive nuclei and the corresponding annotations (pink). These annotations are used in either individual training datasets (gray: expert models) or pooled in a single training dataset by means of ground truth estimation from the expert annotations (blue: consensus models, orange: consensus ensembles). Next, deep learning models are trained on the training dataset and evaluated on a holdout validation dataset. Subsequently, the predictions of individual models (gray and blue) or model ensembles (orange) are used to compute binary segmentation masks for the entire bioimage dataset. Based on these fluorescent feature segmentations, quantification and statistical analyses are performed. The expert model strategy enables the automation of a manual analysis. To mitigate the bias from subjective feature annotations in the expert model strategy we introduce the consensus model strategy. Finally, the consensus ensembles alleviate the random effects in the training procedure and seek to ensure reliability and eventually, validity.

tures in microscopy images. Here, manual annotations of a single human expert are used to train an individual (and hence expert-specific) DL model with a U-Net (Ronneberger, Fischer, and Brox 2015) architecture. U-Net and its variants have emerged as the de facto standard for biomedical image segmentation purposes (McQuin et al. 2018; Falk et al. 2019; Caicedo et al. 2019). The sec-

ond strategy, *consensus models* (blue), addresses the objectivity of signal annotations. Contrary to the first strategy, simultaneous truth and performance level estimation (STAPLE) (Warfield, Zou, and Wells 2004) is used to estimate a ground truth and create consensus annotations. The estimated ground truth (est. GT) annotation reflects the pooled input of multiple human experts and is therefore thought to be less affected by a potential subjective bias of a single expert. We then train a single U-Net model to create a consensus model. The third strategy, *consensus ensembles* (orange), seeks to ensure reliability and eventually validity. Going beyond the second strategy, we train multiple consensus U-Net models to create a consensus ensemble. Such model ensembles are known to be more robust to noise (Dietterich 2000). Hence, we hypothesize that the consensus ensembles mitigate the randomness in the training process. Moreover, deep ensembles are supposed to yield high-quality predictive uncertainty estimates (Lakshminarayanan, Pritzel, and Blundell 2017).

For each of the three strategies, we complete the bioimage analysis by performing quantification and hypothesis testing on a typical fluorescent microscopy image datasetThese images describe changes in fluorescence signal abundance of a protein called cFOS in brain sections of mice. cFOS is an activity-dependent transcription factor and its expression in the brain can be modified experimentally by behavioral testing of the animals (Gallo et al. 2018). The low signal-to-noise ratio of this label, its broad usage in neurobiology, and the well-established correlation of its abundance with behavioral paradigms render it an ideal bioimage dataset to test our hypotheses (Shuvaev et al. 2017; Gallo et al. 2018).

3.3.1 Similarity Analysis for Validity and Reproducibility

The primary goal in bioimage analysis is to rigorously test a biological hypothesis. To leverage the potentials of DL models within this procedure, we need to trust our model – by establishing objectivity, reliability, and validity. Pertaining to the case of fluorescent labels, validity (measuring what is intended to be measured) requires objectivity to know what exactly we intend to measure in the absence of a ground truth. Similarly, reliability in terms of repeatability and reproducibility is a prerequisite for a valid and trustworthy model. Starting from the expert model strategy, we seek to establish objectivity (consensus

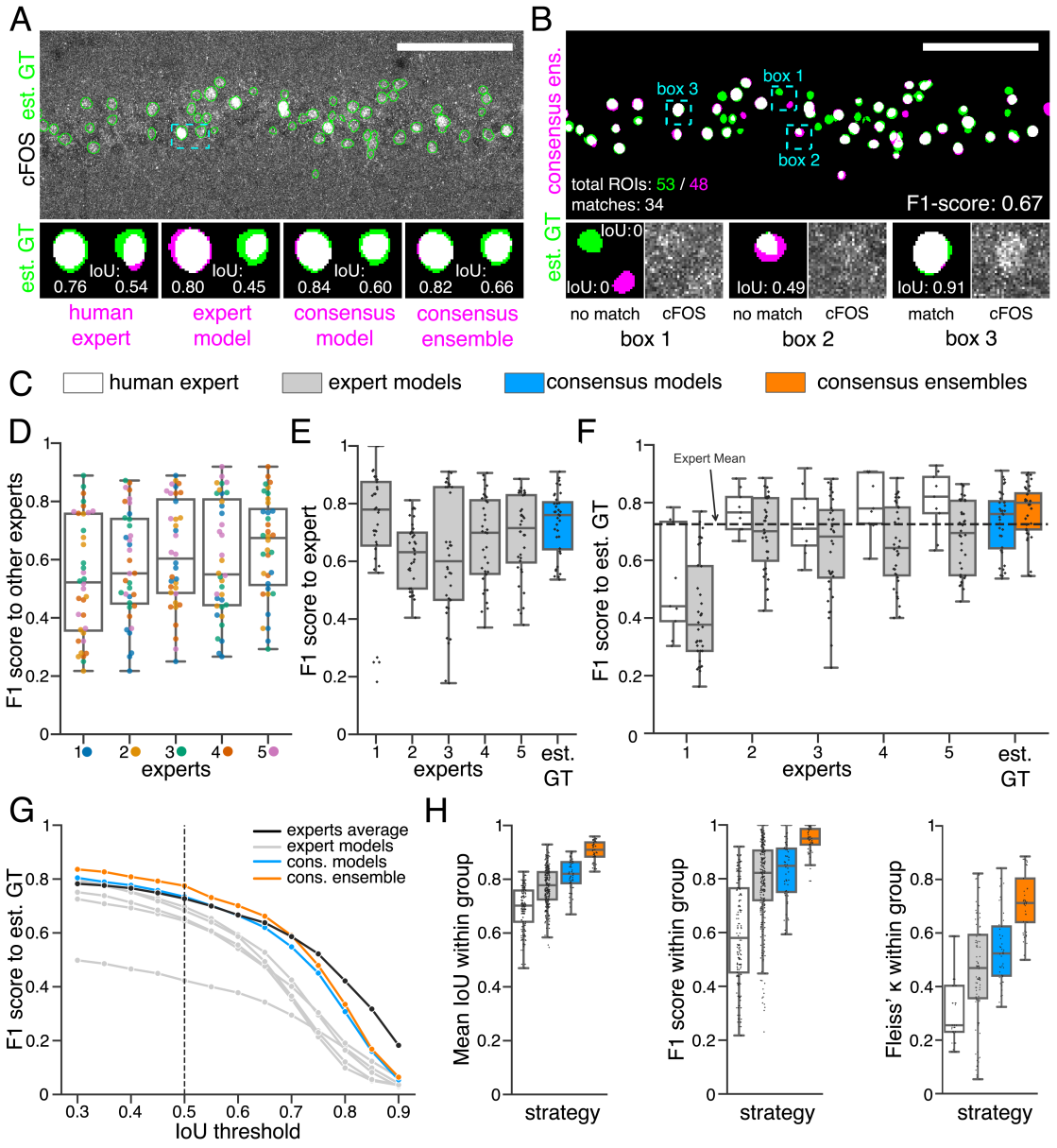


Figure 3.2: Similarity analysis of fluorescent feature annotations by manual or DL-based strategies. (**A**) Representative example of M_{IoU} calculations on a field of view (FOV) in a bioimage. Image raw data show the labeling of cFOS in a maximum intensity projection image of the CA1 region in the hippocampus (brightness and contrast enhanced). The similarity of estimated ground truth (est. GT) annotations (green), derived from the annotations of five expert neuroscientists, are compared to those of one human expert, an expert model, a consensus model, and a consensus ensemble (magenta, respectively). IoU results of two ROIs are shown in detail for each comparison (magnification of cyan box). Scale bar: 100 µm. (**B**) F1 score $M_{\text{F1 score}}$ calculations on the same FOV as shown in (A). [*Caption continues on next page*]

Figure 3.2: [*continued*] The est. GT annotations (green; 53 ROIs) are compared to those of a consensus ensemble (magenta; 48 ROIs). IoU-based matching of ROIs at an IoU-threshold of $t = 0.5$ is depicted in three magnified subregions of the image (cyan boxes 1-3). Scale bar: 100 µm. (**C-H**) All comparisons are performed exclusively on a separate test set which was withheld from model training and validation. (**C**) Color coding refers to the individual strategies, as introduced in Figure 3.1: white: manual approach, gray: expert models, blue: consensus models, orange: consensus ensembles. (**D**) $M_{\text{F1 score}}$ between individual manual expert annotations and their overall reliability of agreement given as the mean of Fleiss' κ. (**E**) $M_{\text{F1 score}}$ between annotations predicted by individual models and the annotations of the respective expert (or est. GT), whose annotations were used for training. (**F**) $M_{\text{F1 score}}$ between manual expert annotations, the respective expert models, consensus models, and consensus ensembles compared to the est. GT as reference. A horizontal line denotes the human expert average. (**G**) Means of $M_{\text{F1 score}}$ of the individual DL-based strategies and of the human expert average compared to the est. GT plotted for different IoU matching thresholds t. A dashed line indicates the default threshold $t = 0.5$. (**H**) Annotation reliability of the individual strategies assessed as the similarities between annotations within the respective strategy. We calculated $\bar{M}_{\text{IoU}}$, $M_{\text{F1 score}}$ and Fleiss' κ.

models) and, successively, reliability and validity in the consensus ensemble strategy. In the following analysis, we first turn towards a comprehensive evaluation of objectivity and its relation to validity before moving on to the concept of reliability.

To assess the three different strategies, a training dataset of 36 images and a test set of nine microscopy images (1024 x 1024 px, 1.61 px / µm, on average $\sim$35 nuclei per image) showing cFOS immunoreactivity were manually annotated by five independent experts (experts 1-5). In absence of a rigorously objective ground truth, we used STAPLE (Warfield, Zou, and Wells 2004) to compute an estimated ground truth (est. GT) based on all expert annotations for each image. First, we trained a set of DL models on the 36 training images and corresponding annotations, either made by an individual human expert or as reflected in the est. GT (see methods for the data set and detailed training, evaluation, and model selection strategy). Then, we used our test set to evaluate the segmentation (Mean IoU) and detection (F1 score) performance of human experts and all trained models by means of similarity analysis on the level of individual images.

For the pairwise comparison of annotations (segmentation masks) we calculated the intersection over union (IoU) for all overlapping pairs of ROIs between two segmentation masks (Figure 3.2A; see Section 3.2.2). Following Maška et al. (2014), we consider two ROIs with an IoU of at least 0.5 as matching and calculated the F1 score $M_{\text{F1 score}}$ as the harmonic mean of precision and recall (3.2B; see Figure 3.2.2). As bioimaging studies predominantly use measures related to counting ROIs in their analyses, we also focused on the feature detection performance ($M_{\text{F1 score}}$). The color coding (gray, blue, orange) introduced in Figure 3.2C refers to the different strategies depicted in Figure 3.1 and applies to all figures, if not indicated otherwise.

To better grasp the difficulties in annotating cFOS-positive nuclei as fluorescent features in these images, we first compared manual expert annotations (Figure 3.2D). The analysis revealed substantial differences between the annotations of the different experts and shows varying inter-rater agreement (Schmitz, Korr, and Heinsen 1999; Collier et al. 2003; Niedworok et al. 2016). The level of inter-rater variability was inversely correlated with the relative signal intensities (see Figure 3.3).

By comparing the annotations of the expert models (gray) to the annotations of the respective expert (Figure 3.2E) we observed a higher $M_{\text{F1 score}}$ median compared to the inter-rater agreement (Figure 3.2D) in the majority of cases. Furthermore, comparing the similarity analysis results of human experts with those of their respective expert-specific models revealed that they are closely related (Figure 3.2F). As pointed out by Chamier, Laine, and Henriques (2019), this indicates that our expert models are able to learn and reproduce the annotation behavior of the individual experts. This becomes particularly evident in the annotations of the DL models trained on expert 1 (Figure 3.2F).

Overall, the expert models yield a lower similarity to the est. GT compared to the consensus models (blue) or consensus ensembles (orange). Notably, both consensus models and consensus ensembles perform on par with human experts. Hereby, the consensus ensembles outperform all other strategies, even at varying IoU thresholds (Figure 3.2F and Figure 3.2G).

In order to test for the reliability of our analysis, we measured the repeatability and reproducibility of fluorescent feature annotation of our DL strategies. We assumed that the repeatability is assured for all our strategies due to the deterministic nature of our DL models (unchanged conditions imply un-

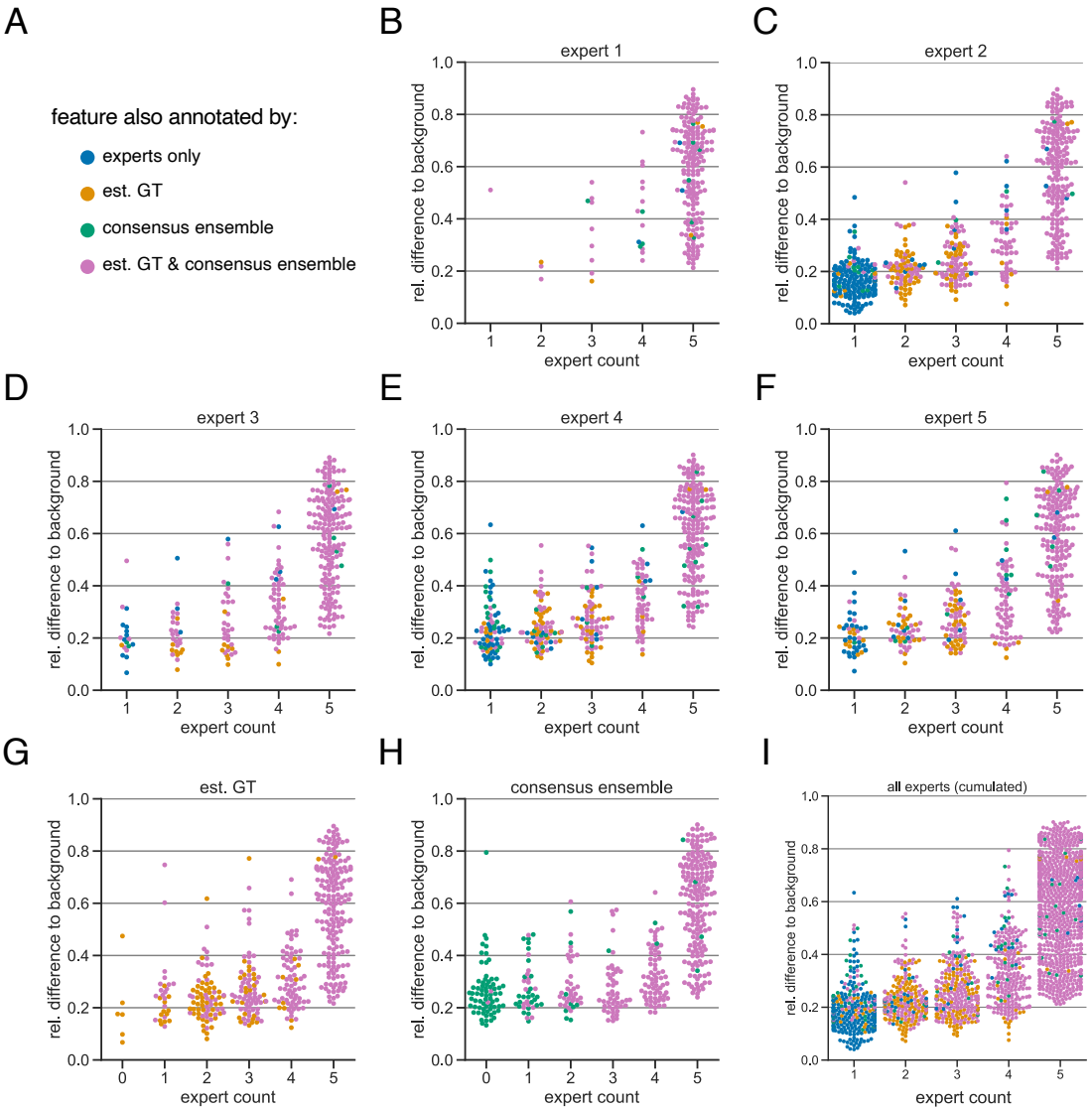


Figure 3.3: Annotation subjectivity analysis. The subjectivity analysis depicts the relationship between the relative intensity difference of a florescent feature (ROI) to the background and the annotation count of human experts. A visual interpretation indicates that the annotation probability of a ROI is positively correlated with its relative intensity. The relative intensity difference is calculated as $\frac{\mu_{inner}-\mu_{outer}}{\mu_{inner}}$, where μ_{inner} is the mean signal intensity of the ROI and μ_{outer} the mean signal intensity of its nearby outer area. We considered matching ROIs at an IoU threshold of $t = 0.5$. The expert in the title of the respective plot was used to create the region proposals of the ROIs, i.e., the annotations served as the origin for the other pairwise comparisons. (**A**) Legend of color codes: blue depicts that a ROI was only annotated by one or more human experts; yellow depicts the ROIs that were present in the estimated ground truth; green shows the ROIs that are only present in an exemplary consensus ensemble; pink depicts ROIs that are present in both estimated ground truth and consensus ensemble. [*Caption continues on next page*]

Figure 3.3: [*continued*] (**B-I**) All calculations are performed on the test set which was withheld from model training and validation. (**B-F**) The individual expert analysis shows the effects of different heuristic evaluation criteria. (**G**) The analysis of the est. GT annotations reveals the limitations of the ground truth estimation algorithm, which is based on the human annotations. An expert count of zero can result from merging different ROIs. (**H**) The analysis of a representative consensus ensemble shows that human annotators may have missed several ROIs (green) even with a large relative difference to the background. (**I**) Cumulative summary of **B-F**.

changed model weights). Hence, our evaluation was focused on the reproducibility, meaning the impact of the stochastic training process on the output. Inter-expert and inter-model comparisons within each strategy unveiled a better performance of the consensus ensembles strategy concerning both detection ($M_{\text{F1 score}}$) and segmentation ($\bar{M}_{\text{IoU}}$) of the fluorescent features (Figure 3.2H). Calculating the Fleiss' kappa value (Fleiss and Cohen 1973) revealed that consensus ensemble annotations show a high reliability of agreement (Figure 3.2H). Following the Fleiss' kappa interpretation from Landis and Koch (1977) the results for the consensus ensembles indicate a substantial or almost perfect agreement. In contrast, the Fleiss' kappa values for human experts refer to a fair agreement, while the results for the alternative DL strategies lead to a moderate agreement (Figure 3.2H).

To determine an appropriate size for the *consensus ensembles*, we analyzed the homogeneity of the results through a similarity analysis. Therefore, we calculated the $M_{\text{F1 score}}$ at an IoU matching threshold of $t = 0.5$ for each ensemble size $i \in \{1, ..., 10\}$ on the holdout test set. Stratified on the cross-validation splits we randomly sampled the ensembles from a collection of trained *consensus models*. We repeated this procedure five times to mitigate the random effect of the ensemble composition ($N_{\text{ensembles}}$=5 for each i). The results are depicted in Figure 3.4.

In summary, the similarity analysis of the three strategies shows that training DL models solely on the input of a single human expert imposes a high risk of incorporating an intrinsic bias and therefore resembles a mere automation of manual image annotation. Both consensus models and consensus ensembles perform on par with human experts regarding the similarity to the est. GT, but the consensus ensembles yield by far the best results regarding

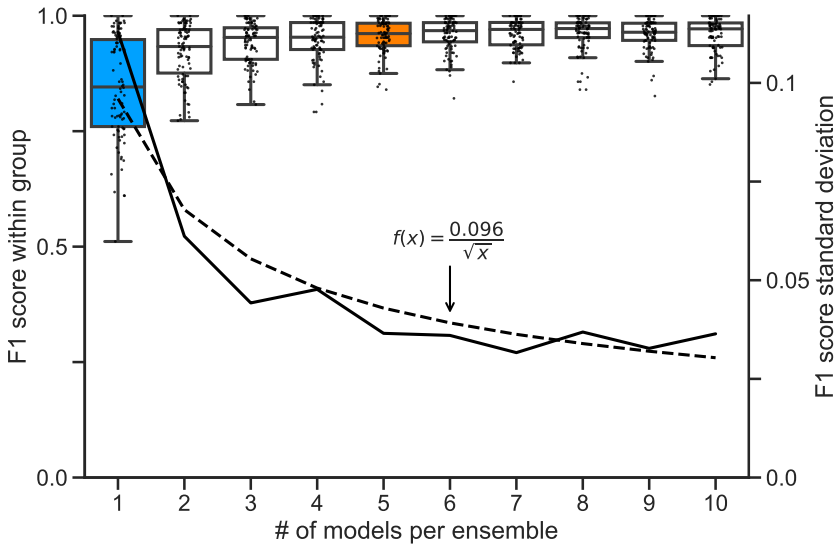


Figure 3.4: Ensemble size and reliability. The blue box depicts the variability between different *consensus models*. The orange box shows the variability of the finally chosen size for the *consensus ensembles*, as no substantial reduction in variation can be observed for larger i. The black line denotes the standard deviation of $M_{\text{F1 score}}$, which is scaled at the right y-Axis. The dashed black line denotes the best fitting function of type $f(x) = {}^{a}/_{\sqrt{x}}$ with $a = 0.096$ for the standard deviation.

their reproducibility. We conclude that, in terms of similarity metrics, only the consensus ensemble strategy meet the bioimaging standards for objectivity, reliability, and validity.

3.3.2 Bioimage Analysis Results

Similarity analysis is inevitable to assess the quality of a model's output, i.e., the predicted segmentations (Ronneberger, Fischer, and Brox 2015; Caicedo et al. 2019; Falk et al. 2019). However, the primary goal of bioimage analysis is the unbiased quantification of distinct image features that correlate with experimental conditions. So far, it has remained unclear whether objectivity, reliability, and validity for bioimage analysis can be inferred directly from similarity metrics.

In order to systematically address this question, we used our image dataset to quantify the abundance of cFOS in brain sections of mice after Pavlovian

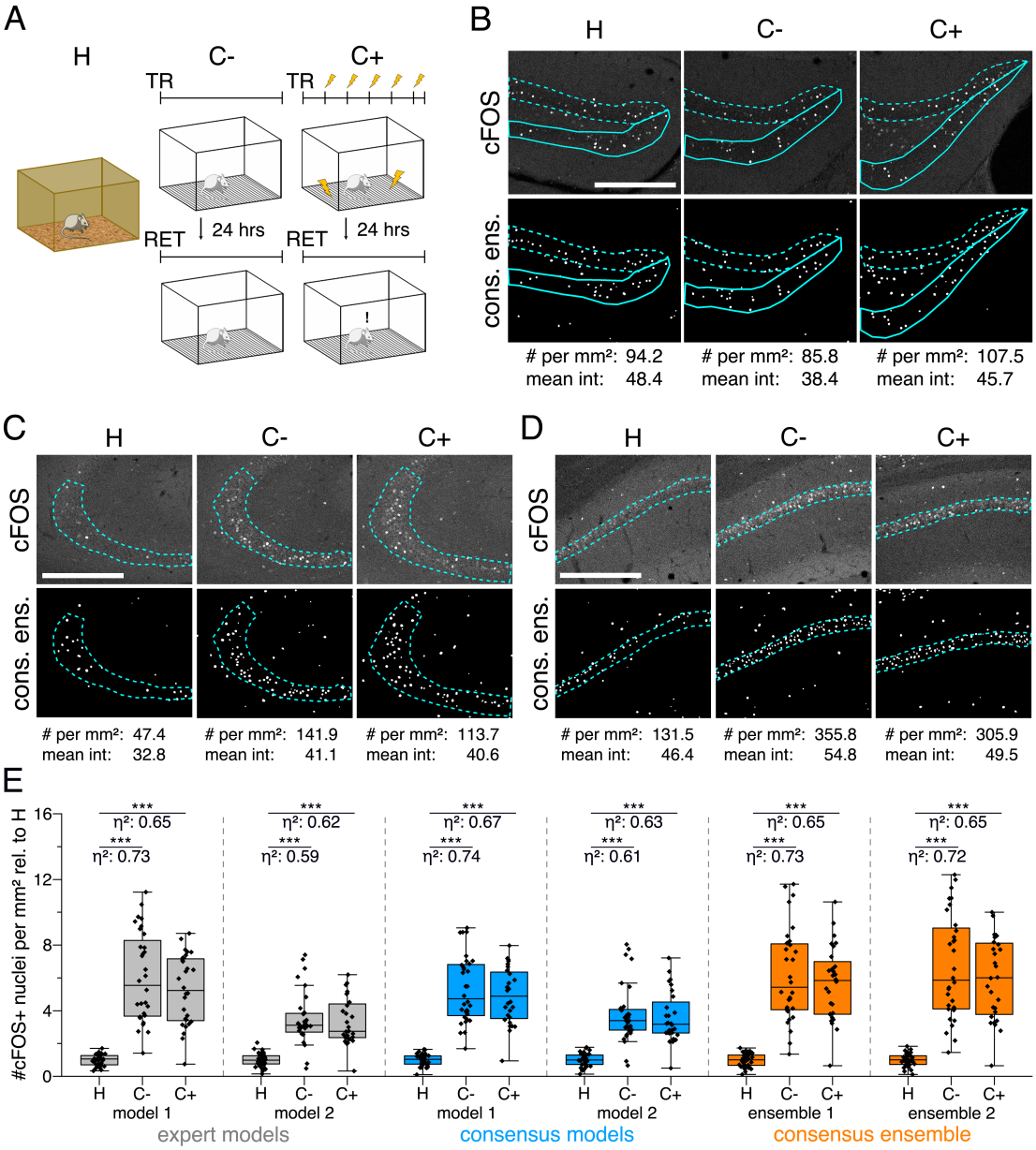


Figure 3.5: Application of different DL-based strategies for fluorescent feature annotation. The figure introduces how three DL-based strategies are applied for annotation of a representative fluorescent label, here cFOS, in a representative image data set. Raw image data show behavior-related changes in the abundance and distribution of the protein cFOS in the dorsal hippocampus, a brain center for encoding of context-dependent memory. (**A**) Three experimental groups were investigated: Mice kept in their homecage (H), mice that were trained to a context, but did not experience an electric foot shock (C-), and mice exposed to five foot shocks in the training context (C+). 24 hours after the initial training (TR), mice were re-exposed to the training context for memory retrieval (RET). Memory retrieval induces changes in cFOS levels. [*Caption continues on next page*]

Figure 3.5: [*continued*] (**B-D**) Brightness and contrast enhanced maximum intensity projections showing cFOS fluorescent labels of the three experimental groups (H, C-, C+) with representative annotations of a consensus ensemble, for each hippocampal subregion. The annotations are used to quantify the number of cFOS-positive nuclei for each image (#) per mm^2 and their mean signal intensity (mean int., in bit-values) within the corresponding image region of interest, here the neuronal layers in the hippocampus (outlined in cyan). In B: granule cell layer (supra- and infrapyramidal blade), dotted line: suprapyramidal blade, solid line: infrapyramidal blade. In C: pyramidal cell layer of CA3; in D: pyramidal cell layer in CA1. Scale bars: 200 µm. (**E**) Analyses of cFOS-positive nuclei per mm^2, representatively shown for stratum pyramidale of CA1. Corresponding effect sizes are given as η^2 for each pairwise comparison. Two quantification results are shown for each strategy and were selected to represent the lowest (model 1 or ensemble 1) and highest (model 2 or ensemble 2) effect sizes (increase in cFOS) reported within each annotation strategy. ***: $p < 0.001$ with Mann-Whitney-U test.

contextual fear conditioning. It is well established in the neuroscientific literature that mice show changes in the distribution and abundance of cFOS in a specific brain region, namely the hippocampus, after processing information about places and contexts (Keiser et al. 2017; Campeau et al. 1997; Huff et al. 2006; Ramamoorthi et al. 2011; Tayler et al. 2013; Murawski, Klintsova, and Stanton 2012; Guzowski et al. 2001). Consequently, our experimental dataset offered us a second line of evidence, the objective analysis of mouse behavior, in addition to the changes of fluorescent features to validate the bioimage analyses results of our DL-based strategies.

Our dataset comprised three experimental groups (Figure 3.5A). In one group, mice were directly taken from their home cage as naïve learning controls (H). In the second group, mice were re-exposed to a previously explored training context as context controls (C-). Mice in the third group underwent Pavlovian fear conditioning and were also re-exposed to the training context (C+) (Figure 3.5A). These three groups of mice showed different behavioral responses. For instance, fear (threat) conditioned mice (C+) showed increased freezing behavior after fear acquisition and showed strong freezing responses when re-exposed to the training context 24 h later. After behavioral testing, brain sections of the different groups of mice were prepared and labeled for the neuronal activity-related protein cFOS by indirect immunofluorescence.

Sections were also labeled with the neuronal marker NeuN (Fox3), allowing the anatomical identification of hippocampal subregions of interest. Images were acquired as confocal microscopy image stacks (x,y-z) and maximum intensity projections were used for subsequent bioimage analysis. Overall, we quantified the number of cFOS-positive nuclei and their mean signal intensity in five regions of the dorsal hippocampus (DG as a whole, suprapyramidal DG, infrapyramidal DG, CA3, and CA1), and tested for significant differences between the three experimental groups (3.5B-D). To extend this analysis beyond hypothesis testing at a certain significance level, we calculated the effect size (η^2) for each of these 30 pairwise comparisons.

We illustrate our metrics with the detailed quantification of cFOS-positive nuclei in the stratum pyramidale of CA1 as a representative example and show two analyses for each DL strategy (Figure 3.5E). These two examples represent those two models of each strategy that yielded the lowest and the highest effect sizes, respectively (Figure 3.5E). Despite a general consensus of all models and ensembles on a context-dependent increase in the number of cFOS-positive nuclei, these quantifications already indicate that the variability of effect sizes decreases from expert models to consensus models and is lowest for consensus ensembles (3.5E).

The analysis in Figure 3.6 allows us to further explore the impact of the different DL strategies on the bioimage analysis results for each hippocampal subregion. Here, we display a high-level comparison of the effect sizes and corresponding significance levels of 20 independently trained expert models (four per expert), 36 consensus models, and nine consensus ensembles (each derived from four consensus models). In contrast to the detailed illustration of selected models in Figure 3.5E, Figure 3.6A, for instance, summarizes the results for all analyses of the stratum pyramidale of CA1. As indicated before, all models and ensembles show a highly significant context-dependent increase in the number of cFOS-positive nuclei, but also a notable variation in effect sizes for both expert and consensus models. Moreover, we identify a significant context-dependent increase in the mean signal intensity of cFOS-positive nuclei for all consensus models and ensembles. The expert models, by contrast, yield a high variation in effect sizes at different significance levels. Interestingly, all four expert models trained on the annotations of expert 1 (and two other expert models only in the case of H vs. C+) did not yield a significant

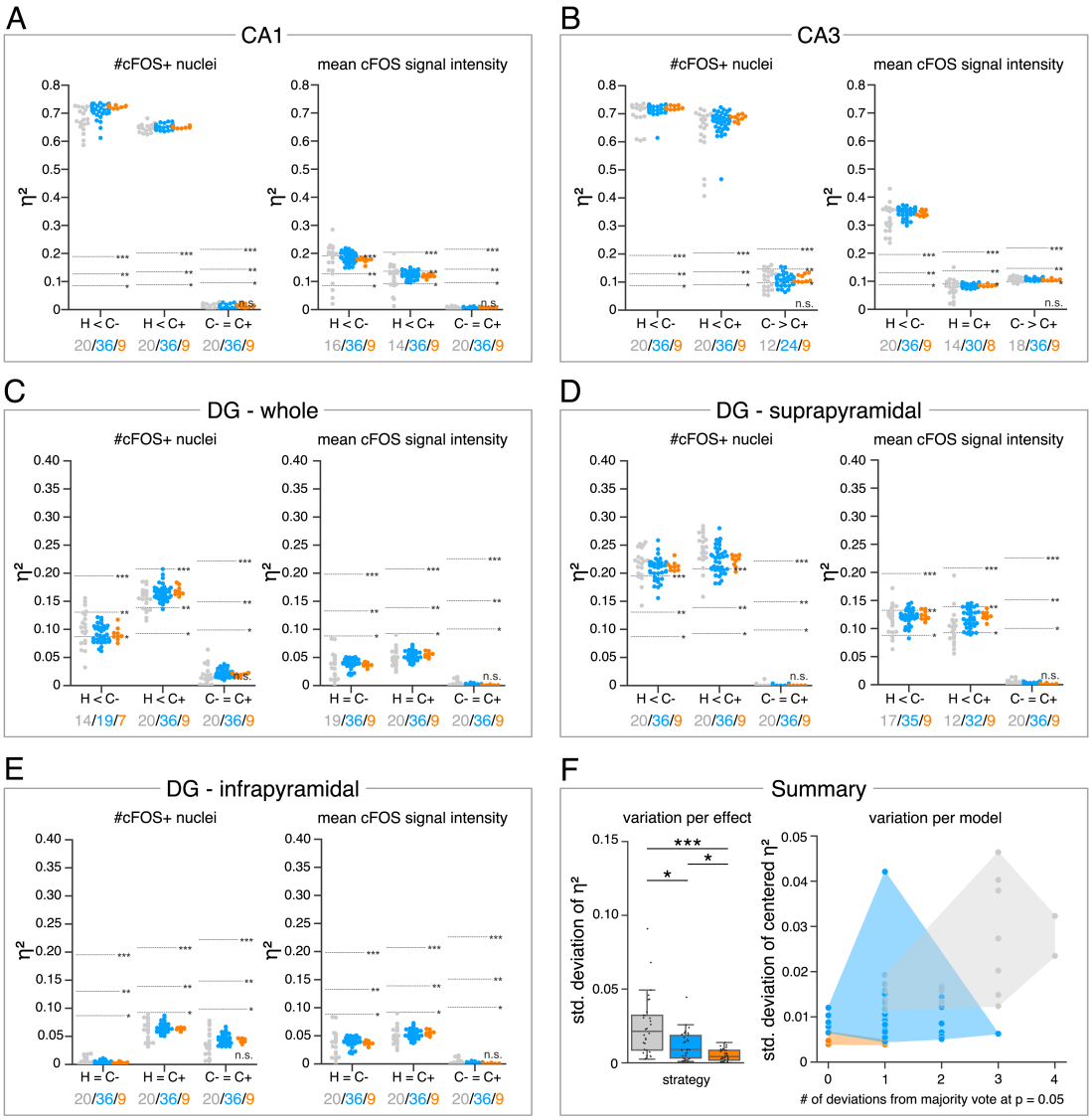


Figure 3.6: Consensus ensembles significantly increase reliability of bioimage analysis results. (**A-E**) Single data points represent the calculated effect sizes for each pairwise comparison of all individual bioimage analyses for each DL-based strategy (gray: expert models, blue: consensus models, orange: consensus ensembles) in indicated hippocampal subregions. Three horizontal lines separate four significance intervals (n.s.: not significant, *: $0.05 \geqslant p > 0.01$, **: $0.01 \geqslant p > 0.001$, ***: $p \leqslant 0.001$ after Bonferroni correction for multiple comparisons). The quantity of analyses of each strategy that report the respective statistical result of the indicated pairwise comparison (effect, x-axis) at a level of $p \leqslant 0.05$ are given below each pairwise comparison in the corresponding color coding. (**A**) Analyses of cFOS-positive nuclei in stratum pyramidale of CA1. [*Caption continues on next page*]

Figure 3.6: [*continued*] (**B**) Analyses of cFOS-positive nuclei in stratum pyramidale of CA3. (**C**) Analyses of cFOS-positive nuclei in the granule cell layer of the whole DG. (**D**) Analyses of cFOS-positive nuclei in the granule cell layer of the suprapyramidal blade of the DG. (**E**) Analyses of cFOS-positive nuclei in the granule cell layer of the infrapyramidal blade of the DG. (**F**) The reliability of bioimage analysis results is assessed as *variation per effect* (left side) and *variation per model* (right side). For the *variation per effect*, single data points represent the standard deviation of reported effect sizes (η^2), calculated within each DL-based strategy for each of the 30 pairwise comparisons. Consensus ensembles show significantly lower standard (std.) deviations of η^2 per pairwise comparison compared to alternative strategies ($X^2(2) = 26.472$, $p < 0.001$, Kruskal-Wallis ANOVA followed by pairwise Mann-Whitney tests with Bonferroni correction, *: $p < 0.05$, ***: $p < 0.001$). For the *variation per model*, the standard deviation of centered η^2 across all pairwise comparisons was calculated for each individual model and ensemble (y-axis). In addition, the number of deviations from the congruent majority vote (at $p \leqslant 0.05$ after Bonferroni correction for multiple comparisons) were determined for each individual model and ensemble across all pairwise comparisons (x-axis). Visualizing the interaction of both measures for each model or model ensemble individually reveals that consensus ensembles show the highest reliability of all three DL-based strategies.

increase, indicating that expert 1's annotation behavior was incorporated into the expert-1-specific models and that this also affects the bioimage analysis results (Figure 3.6A).

The meta-analysis discloses a context-dependent increase of cFOS in almost all analyzed hippocampal regions (Figure 3.6A-D), except for the infrapyramidal blade of the dentate gyrus (Figure 3.6E). Notably, the majority votes of all three strategies at a significance level of $p \leqslant 0.05$ (after Bonferroni correction for multiple comparisons) are identical for each pairwise comparison (Figure 3.6A-E). However, the results can vary between individual models or ensembles (Figure 3.6A-E).

In order to assess the reliability of bioimage analysis results of the individual strategies, we further examined the variation per effect and variation per model in Figure 3.6F. For the variation per effect, we calculated the standard deviation of reported effect sizes within each strategy for every pairwise comparison (effect). This confirmed the visual impression from Figure 3.6A-E as the consensus ensembles yield a significantly lower standard deviation com-

pared to both alternative strategies (Figure 3.6F). To illustrate the variation per model, we show the interaction between the number of biological effects that the corresponding model (or ensemble) reported differently compared to the congruent majority votes versus the standard deviation of its centered effect sizes across all 30 analyzed effects. This analysis shows that no expert model detected all biological effects in the microscopy images that were defined by the majority votes of all models. This is in stark contrast to the consistency of effect interpretation across the consensus ensembles (Figure 3.6F).

Consequently, we conclude that the consensus ensemble strategy is best suited to satisfy the bioimaging standards for objectivity, reliability, and validity.

3.3.3 Bioimage Analysis of External Datasets

Bioimage analysis of fluorescent labels comes with huge variability in terms of investigated model organisms, analyzed fluorescent features, and applied image acquisition techniques (Meijering et al. 2016). In order to assess our consensus ensemble strategy across these varying parameters, we tested it on four external datasets that were created in a fully independent manner and according to individual protocols (*Lab-Mue, Lab-Inns1, Lab-Inns2*, and *Lab-Wue2*). Due to limited dataset sizes, the lab-specific training datasets consisted of just five microscopy images each and the corresponding est. GT based on the annotations from multiple experts. In the biomedical research field, the limited availability of training data is a common problem when training DL algorithms. For this reason, extensive data augmentation and regularization techniques, as well as transfer learning strategies are widely used to cope with small datasets (Ronneberger, Fischer, and Brox 2015; Christiansen et al. 2018; Falk et al. 2019). Transfer learning is a technique that enables DL models to reuse the image feature representations learned on another source, such as a task (e.g. image segmentation) or a domain (e.g. the fluorescent feature, here cFOS-positive nuclei). This is particularly advantageous when applied to a task or domain where limited training data is available (Yosinski et al. 2014; Oquab et al. 2014). Moreover, transfer learning might be used to reduce observer variability and to increase feature annotation objectivity (Bayramoglu and Heikkilä 2016). There are typically two ways to implement transfer learning for DL models, either by

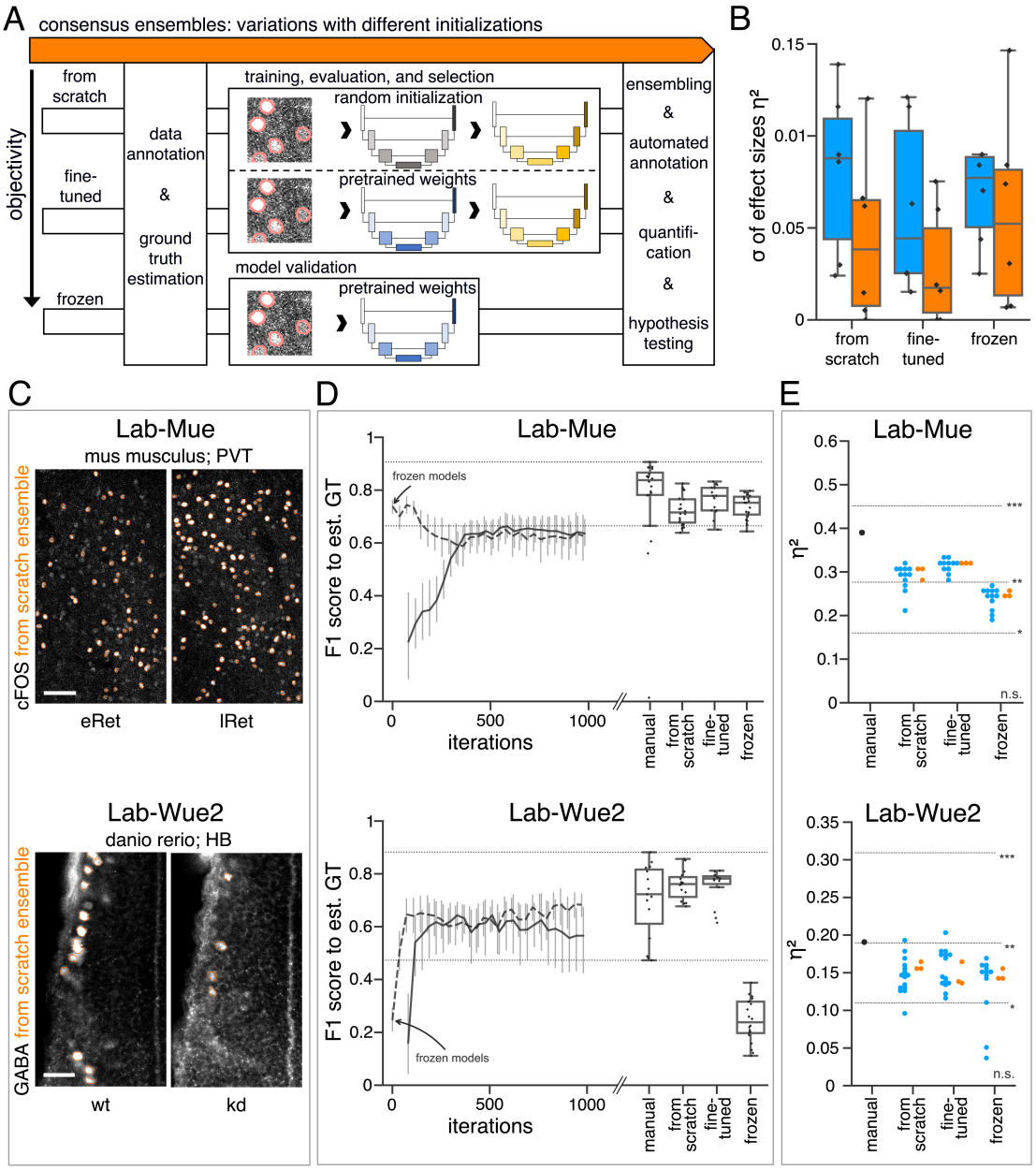


Figure 3.7: Consensus ensembles for DL-based feature annotation in external bioimage data sets. (**A**) Schematic overview depicting three initialization variants for creating consensus ensembles on new datasets. Data annotation by multiple human experts and subsequent ground truth estimation are required for all three initialization variants. In the *from scratch* variant, a U-Net model with randomly initialized weights is trained on pairs of microscopy images and estimated ground truth annotations. This variant was used to create consensus ensembles for the initial *Lab-Wue1* dataset. [*Caption continues on next page*]

Figure 3.7: [*continued*] Alternatively, the same training dataset can be used to adapt a U-Net model with pretrained weights by means of transfer learning (*fine-tuned*). In both variants, models are evaluated and selected on the basis of a validation set after model training. In a third variant, U-Net models with pretrained weights can be evaluated directly on a validation dataset, without further training (*frozen*). In all three variants, consensus ensembles of the respective models are then used for bioimage analysis. (**B**) Overall reliability of bioimage analysis results of each variant assessed as variation per effect. In all three strategies, consensus ensembles (orange) showed lower standard deviations than consensus models (blue). The *frozen* results need to be considered with caution as they are based on models that did not meet the selection criterion. **(C-E)** Detailed comparison of the two external datasets with highest (*Lab-Mue*) and lowest (*Lab-Wue2*) similarity to *Lab-Wue1*. (**C**) Representative microscopy images. Orange: representative annotations of a lab-specific *from scratch* consensus ensemble. PVT: para-ventricular nucleus of thalamus, eRet: early retrieval, lRet: late retrieval, HB: hindbrain, wt: wildtype, kd: *gad1b* knockdown. Scale bars: *Lab-Mue* 100 µm and *Lab-Wue2* 6 µm. (**D**) Mean $M_{\text{F1 score}}$ of *from scratch* (solid line) and *fine-tuned* (dashed line) consensus models on the validation dataset over the course of training (iterations). Mean $M_{\text{F1 score}}$ of *frozen* consensus models are indicated with arrows. Box plots show the $M_{\text{F1 score}}$ among the annotations of human experts as reference and the mean $M_{\text{F1 score}}$ of selected consensus models. Two dotted horizontal lines mark the whisker ends of the $M_{\text{F1 score}}$ among the human expert annotations. (**E**) Effect sizes of all individual bioimage analyses (black: manual experts, blue: consensus models, orange: consensus ensembles). Three horizontal lines separate the significance levels (n.s.: not sign., *: $0.05 \geqslant p > 0.01$, **: $0.01 \geqslant p > 0.001$, ***: $p \leqslant 0.001$ with Mann-Whitney-U tests).

fine-tuning or by freezing features (i.e., model weights) (Yosinski et al. 2014). The latter approach, if applied to the same task (e.g., image segmentation), does not require any further model training. These *out-of-the-box* models reduce time and hardware requirements and may further increase the objectivity of image analysis, by altogether excluding the need for any additional manual input.

Consequently, we hypothesized that transfer learning from pretrained model ensembles would substantially reduce the training efforts (Falk et al. 2019) and might even increase objectivity of bioimage analysis. To test this, we followed three different initialization variants of the consensus ensemble strategy (Figure 3.7A). In addition to starting the training of DL models with

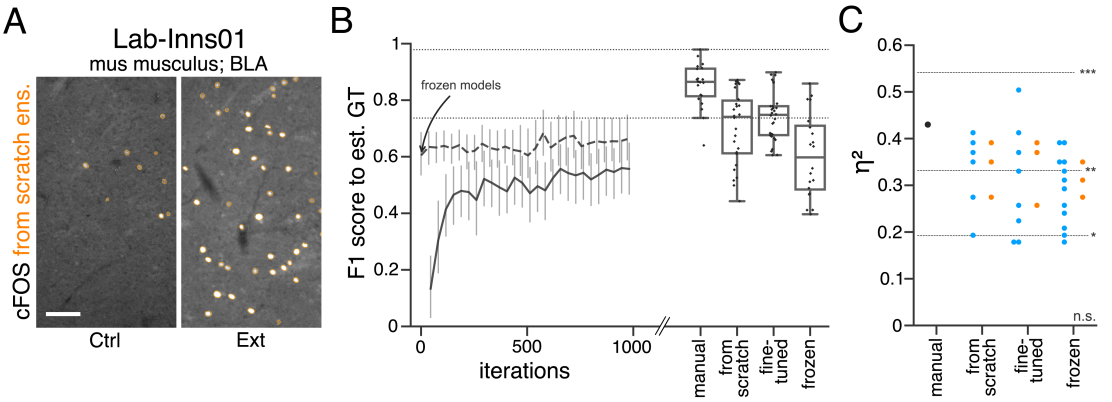


Figure 3.8: Performance of consensus ensembles on feature annotation in image dataset *Lab-Inns01*. (**A**) Representative microscopy images. Orange: representative annotations of a lab-specific *from scratch* consensus ensemble. BLA: basolateral amygdala, Ctrl: control, Ext: extinction. Scale bar: 80 µm. (**B**) Mean $M_{\text{F1 score}}$ of *from scratch* (solid line) and *fine-tuned* (dashed line) consensus models on the validation dataset over the course of training (iterations). Mean $M_{\text{F1 score}}$ of *frozen* consensus models are indicated with an arrow. Box plots show the $M_{\text{F1 score}}$ among the annotations of human experts as reference and the mean $M_{\text{F1 score}}$ of selected consensus models. Two dotted horizontal lines mark the whisker ends of the $M_{\text{F1 score}}$ among the human expert annotations. (**C**) Effect sizes of all individual bioimage analyses (black: manual experts, blue: consensus models, orange: consensus ensembles). Three horizontal lines separate four selected significance intervals (n.s.: not significant, *: $0.05 \geqslant p > 0.01$, **: $0.01 \geqslant p > 0.001$, ***: $p \leqslant 0.001$).

randomly initialized weights (Figure 3.7A - *from scratch*), we reused the consensus ensemble weights from the previous evaluation (*Lab-Wue1*) by means of fine-tuning (A). In addition to starting the training of DL models with randomly initialized weights (Figure 3.7A - *fine-tuned*) and freezing of all model layers (Figure 3.7A - *frozen*). Although no training of the *frozen* model is required, we tested and evaluated the performance of *frozen* models to ensure their validity. After performing the similarity analysis, we compared the full bioimage analyses, including quantification and hypothesis testing, of the different initialization variants. Finally, to establish a notion of external validity, we also compared these results with the manually and independently performed bioimage analysis of a lab-specific expert (Figures 3.7, 3.8, 3.9).

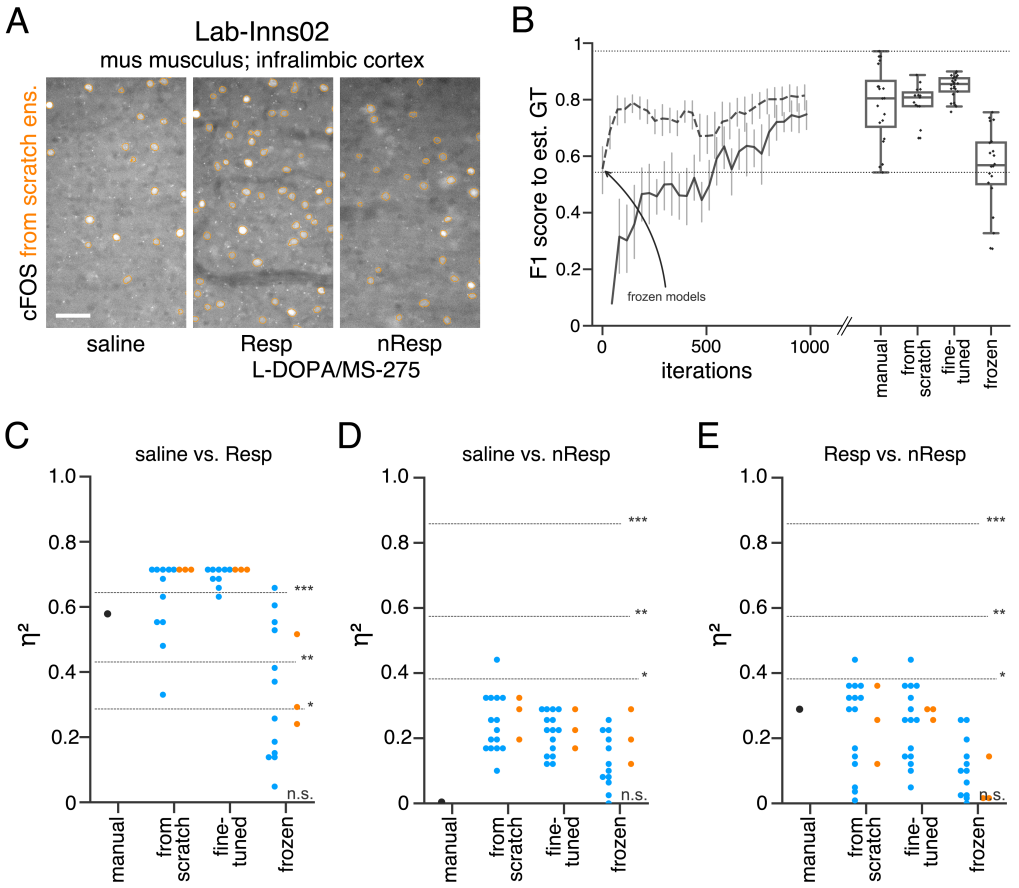


Figure 3.9: Performance of consensus ensembles on fluorescent feature annotation in image dataset *Lab-Inns02*. (**A**) Representative microscopy images. Orange: representative annotations of a lab-specific *from scratch* consensus ensemble. Resp: responders, nResp: non-responders. Scale bar: 40 µm. (**B**) Mean $M_{\text{F1 score}}$ of *from scratch* (solid line) and *fine-tuned* (dashed line) consensus models on the validation dataset over the course of training (iterations). Mean $M_{\text{F1 score}}$ of *frozen* consensus models are indicated with an arrow. Box plots show the $M_{\text{F1 score}}$ among the annotations of human experts as reference and the mean $M_{\text{F1 score}}$ of selected consensus models. Two dotted horizontal lines mark the whisker ends of the $M_{\text{F1 score}}$ among the human expert annotations. (**C-E**) Effect sizes of all individual bioimage analyses (black: manual experts, blue: consensus models, orange: consensus ensembles). Three horizontal lines separate four selected significance intervals (n.s.: not significant, *: $0.05 \geqslant p > 0.01$, **: $0.01 \geqslant p > 0.001$, ***: $p \leqslant 0.001$ after Bonferroni correction for multiple comparisons).

Dataset characteristics

The first dataset (*Lab-Mue*) represents very similar image parameters compared to our original *Lab-Wue1* dataset (Figure 3.7C - *Lab-Mue*). Mice experienced restraint stress and subsequent Pavlovian fear conditioning (cue-conditioning, tone-footshock association) and the number of cFOS-positive cells in the paraventricular thalamus (PVT) was compared between early (eRet) and late (lRET) phases of fear memory retrieval. In the context of transfer learning, this dataset originates from a very similar domain and requires the same task (image segmentation). Another two external datasets are focused on the quantification of cFOS abundance (similar domain), albeit showing less similarity in image parameters to our initial dataset (Figure 3.8, Figure 3.9). In *Lab-Inns1*, mice underwent Pavlovian fear conditioning and extinction in the same context. The image dataset of *Lab-Inns2* shows cFOS immunoreactivity in the infralimbic cortex (IL) following fear renewal, meaning return of extinguished fear in a context different from the extinction training context. Since heterogeneity in this behavioral response was observed, mice were classified as responders (Resp) or non-responders (nResp), based on freezing responses (see methods). The image dataset of *Lab-Wue2* shows the least similarity of image parameters to the dataset of *Lab-Wue1*. This dataset represents another commonly used model organism in neurobiology, the zebrafish. Here, cell bodies of specific neurons (GABAergic neurons) instead of nuclei were fluorescently labeled (Figure 3.8C - *Lab-Wue2*). Hence, this dataset originates from a different domain but was acquired using the same technique.

Similarity analysis

As only limited training data was available we executed the similarity analysis for all external datasets by means of *k-fold cross-validation*. We observed that the inter-rater variability differed between laboratories and different experts but remained comparable as previously for *Lab-Wue1* (Figure 3.7D). Both *from scratch* and *fine-tuned* initiation variants resulted in individual consensus models that reached human expert-level performance (Figures 3.7D, 3.8, 3.9). However, models adapted from pretrained weights yielded a higher validity in terms of similarity to the estimated ground truth. They either exceeded the maximal $M_{\text{F1 score}}$ reached by *from scratch* models (Figures 3.7D - *Lab-Mue*,

3.8, 3.9) or reached them after less training iterations (Figure 3.7D - *Lab-Wue2*). As expected, the performance of *frozen Lab-Wue1*-specific consensus models was highly dependent on the image similarity between the original and the new dataset. Consequently, the *out-of-the-box* segmentation performance of the *frozen Lab-Wue1* models was very poor on dissimilar images (Figure 3.7D - *Lab-Wue2*), but we found it to be on par with human experts and adapted models on images that are highly similar to the original dataset (Figure 3.7D - *Lab-Mue* - very similar domain and the same task).

Bioimage analysis results

To further strengthen the validity of our workflow, we compared all DL-based bioimage analyses to the manual analysis of a human expert from the individual laboratory (Figures 3.7E, 3.8, 3.9, and Table 3.1).

For *Lab-Mue*, the bioimage analyses of all DL-based approaches, including the *frozen* consensus models and ensembles pretrained on *Lab-Wue1*, revealed a significantly higher number of cFOS-positive cells in the PVT of mice 24h after fear conditioning (lRET), which was confirmed by the manual expert analysis (Figure 3.7E - *Lab-Mue*, Table 3.1). Yet again, the formation of model ensembles increased the reproducibility of results by yielding less or almost no variation in the effect sizes (Figure 3.7E - *Lab-Mue*).

The manual expert analysis of the *Lab-Inns1* dataset revealed a significantly higher number of cFOS-positive nuclei in the basolateral amygdala (BLA) after extinction of a previously learned fear, which was also reliably detected by all consensus ensembles, regardless of initiation variant (Figure 3.8, Table 3.1). However, this significant difference was only present in the analyses of most individual consensus models, both *from scratch* and *fine-tuned* (Figure 3.8). Again, this could be attributed to a higher variability between the effect sizes of individual models, compared to a higher homogeneity among ensembles (Figure 3.8).

For *Lab-Inns2*, the manual expert analysis as well as all deep-learning based approaches that were adapted to the *Lab-Inns2* dataset show increased numbers of cFOS-positive cells in the infralimbic cortex of L-DOPA/MS-275 responders (Resp) compared to control (Sal) mice (Figure 3.9, Table 3.1). However, in L-DOPA/MS-275 non-responders (nResp), we did not observe a significant

Table 3.1: Bioimage analyses results of external datasets. Data are based either on manual analysis or on annotations by a consensus ensemble. The results are given for the individual consensus ensemble initialization variants (*from scratch*, *fine-tuned*). P-values of *Lab-Inns2* are corrected for multiple comparisons using Bonferroni correction. μ_1: mean group 1, μ_2: mean group 2

Lab	groups	initialization variant	μ_1	μ_2	U	significance level (p)	η^2
Mue	eRet ∼ lRet	manual	1.00	1.65	19.0	** (0.002)	0.39
		from scratch	1.00	1.70	25.0	** (0.007)	0.31
		fine-tuned	1.00	1.68	24.0	** (0.006)	0.32
Inns1	Ctrl ∼ Ext	manual	1.00	3.92	10.0	** (0.005)	0.43
		from scratch	1.00	2.26	13.0	* (0.010)	0.35
		fine-tuned	1.00	1.85	14.0	* (0.013)	0.33
Inns2	Sal ∼ Resp	manual	1.00	1.83	5.0	** (0.002)	0.59
		from scratch	1.00	1.96	0.0	*** (<0.001)	0.71
		fine-tuned	1.00	2.07	0.0	*** (<0.001)	0.71
	Sal ∼ nResp	manual	1.00	1.05	27.0	n.s. (1.000)	0.00
		from scratch	1.00	1.63	8.0	n.s. (0.130)	0.29
		fine-tuned	1.00	1.42	12.0	n.s. (0.377)	0.16
	Res ∼ nRes	manual	1.83	1.05	42.0	n.s. (0.130)	0.29
		from scratch	1.96	1.63	41.0	n.s. (0.173)	0.26
		fine-tuned	2.07	1.42	42.0	n.s. (0.130)	0.29
Wue2	wt ∼ kd	manual	1.00	0.28	227.5	* (0.010)	0.19
		from scratch	1.00	0.45	220.0	* (0.021)	0.16
		fine-tuned	1.00	0.37	216.0	* (0.029)	0.14

increase of cFOS-positive nuclei (Figure 3.9, Table 3.1). Furthermore, the high effect sizes of the comparison between L-DOPA/MS-275 responders and non-responders further indicate that the differences observed in the behavioral responses of Resp and nResp mice were also reflected in the abundance of cFOS in the infralimbic cortex (Figure 3.9, Table 3.1).

Manual expert analysis of the fourth external dataset revealed a significantly lower amount of GABA-positive somata in *gad1b* knock-down zebrafish, compared to wildtypes (Figure 3.7E - *Lab-Wue2*, Table 3.1). Again, this effect was reliably detected by all deep-learning based approaches that included training on the *Lab-Wue2*-specific training dataset and the effect sizes of ensembles showed less variability (Figure 3.7E - *Lab-Wue2*). Despite its poor segmenta-

tion performance and hence, poor validity, this effect was also present in the bioimage analysis of the *frozen* consensus models and ensembles pretrained on *Lab-Wue1* (Figure 3.7E - *Lab-Wue2*).

As with our initial dataset, we assessed reliability by calculating the variation per effect as the standard deviation of the reported effect sizes within each group and pooled these results across all external datasets. Consistent with the higher reliability of *from scratch* and *fine-tuned* ensemble annotations, this analysis shows that the formation of model ensembles reduced the variation per effect in both variants, compared to the respective individual models (Figure 3.7B). The *frozen* models and ensembles exhibit a similar effect but need to be considered with caution as they are based on models that did not meet the selection criterion (reliably performing on par with human experts; see 3.2.3 for a detailed explanation).

In summary, we assessed the reproducibility of our consensus ensemble strategy by using four external datasets. These datasets were acquired with different image acquisition techniques, investigate two common model organisms, and analyze the two main cellular compartments (nuclei and somata) at varying resolutions. In line with the results obtained on our initial dataset, we observed an increased reproducibility for the consensus ensembles compared to individual consensus models after training on all four external datasets (Figure 3.7B).

Moreover, our data also suggests that pretrained consensus models can even be deployed *out-of-the-box*, but only when carefully validated. Thus, sharing pretrained model weights across different laboratories reduces lab-specific biases within the bioimage analysis and may further increase objectivity and validity.

Ultimately, we conclude that our proposed ensemble consensus workflow is reproducible for different data sets and laboratories and increases the objectivity, reliability, and validity of DL-based bioimage analyses.

3.4 Discussion

The present study contributes to bridging the gap between “methods” and “biology” oriented studies in image feature analysis (Meijering et al. 2016). We ex-

plored the potentials and limitations of DL models utilizing the general quality criteria for quantitative research: objectivity, reliability, and validity. Thereby, we put forward an effective but easily implementable strategy that aims to establish reproducible, DL-based bioimage analysis within the life science community.

The number of DL-based tools for bioimage annotations and their accessibility for non-AI specialists is gradually increasing (McQuin et al. 2018; Haberl et al. 2018; Falk et al. 2019). DL models can hold advantages over conventional algorithms (Caicedo et al. 2019) and have the potential to be commonly used for bioimage analysis tasks throughout the life sciences. Usually, the performance of new bioimage analysis tools or methods is assessed by means of similarity measures to a certain ground truth (Ronneberger, Fischer, and Brox 2015; McQuin et al. 2018; Haberl et al. 2018; Falk et al. 2019; Caicedo et al. 2019). However, this is rarely sufficient to establish trust in the use of DL models for bioimage analysis, as the vast amount of parameters and flexibility to adapt DL models to virtually any task renders them prone to internalize unintended, but subjective human biases (Chamier, Laine, and Henriques 2019). This is particularly true in the case of fluorescent feature analysis in bioimage datasets, as an objective ground truth is not available. In conjunction with the stochastic training process, this is a very critical point, because it holds the potential for intended or unintended tampering similar to p-hacking (Head et al. 2015), e.g., by training DL models until non-significant results become significant.

To investigate the effects of DL-based strategies on the bioimage analysis of fluorescent features, we acquired a typical bioimage dataset (*LabWue1*), and five experts manually annotated corresponding ROIs (here cFOS-positive nuclei) in a representative subset of images. Then, we tested three DL-based strategies for automatized feature segmentation. DL models were either trained on the manual annotations of a single expert (expert models) or on the input of multiple experts pooled by ground truth estimation (consensus models). In addition, we formed ensembles of consensus models (consensus ensembles).

3.4.1 Similarity Analysis of Fluorescent Feature Annotation

In accordance with previous studies, similarity analyses revealed a substantial level of inter-rater variability in the heuristic annotations of the single experts (Schmitz, Korr, and Heinsen 1999; Collier et al. 2003; Niedworok et al. 2016). Furthermore, we confirmed the concerns already put forward by others (Falk et al. 2019; Chamier, Laine, and Henriques 2019) that training of DL models solely on the input of a single human expert imposes a high risk of incorporating an individual human bias into the trained models. We therefore conclude that models trained on single expert annotations resemble an automation of manual image annotation, but cannot remove subjective biases from bioimage analyses. Importantly, only consensus ensembles led to a coincident significant increase also in the reliability and validity of fluorescent feature annotations. Our analyses also show that annotations of multiple experts are imperative for two reasons: first, they mitigate or even eliminate the bias of expert-specific annotations and, secondly, are essential for the assessment of the model performance.

3.4.2 Reproducibility and Validity of Bioimage Analyses

Our bioimage dataset from *Lab-Wue1* enabled us to look at the impact of different DL-based strategies on the results of bioimage analyses. This revealed a striking model-to-model variability as the main factor impairing the reproducibility of DL-based bioimage analyses. Convincingly, the majority votes for each effect were identical for all three strategies. However, the variance within the reported effect sizes differed significantly for each strategy. This entailed, for example, that no expert model was in full agreement with the congruent majority votes. On the contrary, consensus ensembles detected all effects with significantly higher reliability. Thus, our data indicates that bioimage analysis performed with a consensus ensemble significantly reduces the risk of obtaining irreproducible results.

3.4.3 Evaluation of consensus ensembles on external datasets

We then tested our consensus ensemble approach and three initialization variants on four external datasets with limited training data and varying similari-

ties in terms of image parameters to our original dataset (*Lab-Wue1*). In line with previous studies on transfer learning, we demonstrate that the adaptation of models from pretrained weights to new, yet similar data requires less training iterations, compared to the training of models *from scratch* (Falk et al. 2019). We extend these analyses and show that the reliability of *fine-tuned* ensembles was at least equivalent to *from scratch* ensembles, if not higher. Furthermore, we also provide initial evidence that pretrained ensembles can be used even without any adaptation, if task similarity is sufficiently high. Our data suggest that this component in the analysis pipeline could further increase the objectivity of bioimage analyses.

3.4.4 Potentials of Open Source Model Libraries

Sharing model weights from validated models in open-source libraries, similarly to *TensorFlow Hub*[23] or *PyTorch Hub*[24], offers a great opportunity to provide annotation experience across labs in an open science community. In this study, for instance, we used the nuclear label of cFOS, an activity-dependent transcription factor, as fluorescent feature of interest. This label is in its signature indistinguishable from a variety of other fluorescent labels, like those of transcription factors (CREB, phospho-CREB, Pax6, NeuroG2 or Brain3a), cell division markers (phospho-histone H3), apotposis markers (Caspase-3), and multiple others. Similarly to the pretrained and shared models of Falk et al. (2019), we surmise that the learned feature representations (i.e., model weights) of our cFOS consensus ensembles may serve as a good initialization for models that aim at performing nucleosomatic fluorescent label segmentation in brain slices.

In line with the results of the *Kaggle Data Science Bowl 2018* (Caicedo et al. 2019), however, our findings indicate that a model adapted to a specific data set usually outperforms a general model trained on different datasets from different domains. To use and share frozen *out-of-the-box* models across the science community, we therefore need to create a well-documented library that contains validated model weights for each specific task and domain (e.g., for each organism, marker type, image resolution, etc.). In conjunction with

[23] `www.TensorFlow.org/hub`
[24] `www.pytorch.org/hub/`

data repositories, this would also allow retrospective data analysis of prior studies.

In summary, open-source model libraries may contribute to better reproducibility of scientific experiments (Fanelli 2018) by improving the objectivity in bioimage analyses, by offering openness to analysis criteria, and by sharing pretrained models for (re-)evaluation.

3.4.5 Limitations

This paper describes a blueprint for the evaluation of DL models in biomedical imaging. Therefore, some of our methodological decisions were shaped by standardization considerations concerning the future deployment in bioimage analysis pipelines.

The project was triggered by segmentation tasks for fluorescent labels (cFOS) in the cell nucleus. These are rather simple features, and we could readily annotate data from different labs, which facilitated the evaluation. However, this limits the generalizability to more complex image segmentation tasks, where training data annotation is slow and tedious. In particular human perceptive capabilities for richer graphical features, such as area, volume, or density, is much worse than for regular, linear image features (Cleveland and McGill 1985; Feldman-Stewart et al. 2000). A case in point is the annotation of images showing ramified neurons or astrocytes. Such tasks would cause an enormous workload rendering complete human annotation virtually impossible. In this respect, we concur with prior research asserting that DL models based on human annotations will not be an option in these settings (Driscoll et al. 2019).

The characteristics of our examined strategies are based on best practices in the field of DL and derived from the extant literature (Meijering et al. 2016; Falk et al. 2019; Caicedo et al. 2019). The focus on the U-Net model architecture (Ronneberger, Fischer, and Brox 2015) is a direct consequence of this standardization idea. Yet, it is also an important limitation of our study. Unlike more conventional studies that introduce a new method and provide a comprehensive performance comparison to the state of the art, we rely on U-net as the widely studied de facto standard for biomedical image segmentation purposes (McQuin et al. 2018; Falk et al. 2019; Caicedo et al. 2019). Similarly, we chose to use (STAPLE, Warfield, Zou, and Wells 2004) as the benchmark procedure

for ground truth estimation. Thereby, we forwent considering alternatives and variants (Lampert, Stumpf, and Gançarski 2016). In addition, we tried different ways to incorporate the single expert annotations into one DL model. For instance, we followed the approach of Guan et al. (2018) by modeling individual experts in a multi-head DL model instead of pooling them in the first place. However, we decided to discard the approach as our tests did not improve the results but increased complexity.

3.4.6 Accessibility

To enable other researchers to easily access, interact with, and reproduce our results and to share our trained models, we provide an open-source *Python* library that is easily accessible for both local installation and cloud-based deployment.

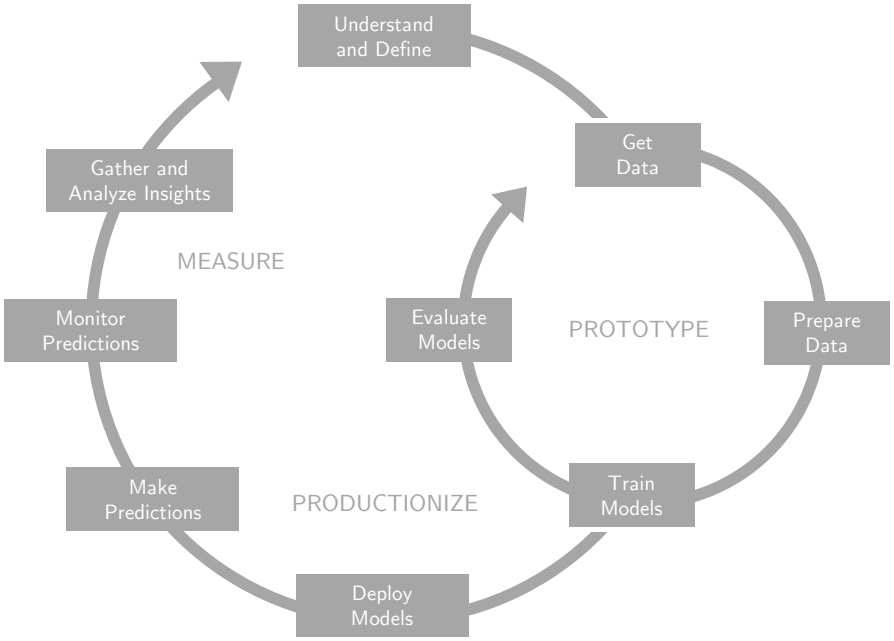
With *Jupyter Notebooks* becoming the computational notebook of choice for data scientists (Perkel 2018) we also implemented a training pipeline for non-AI experts in a *Jupyter Notebook* optimized for Google Colab, providing free access to the required computational resources (e.g., GPUs and TPUs). In summary, we recommend using the annotations of multiple human experts to train and evaluate DL consensus model ensembles. In such a way, DL can be used to increase the objectivity, reliability, and validity of bioimage analyses and pave the way for higher reproducibility in science.

4 Deep learning in the bioimaging wild: Handling ambiguous data with *deepflash2*

This chapter is adapted from the article of Griebel, M., Segebarth, D., Stein, N., Schukraft, N., Tovote, P., Blum, R., & Flath, C. M. *Deep-learning in the bioimaging wild: Handling ambiguous data with deepflash2* published as a preprint[25]. The tool can easily be accessed in Google Colab[26]. All data and source code are available on GitHub[27]. The repository also contains Jupyter notebooks with instructions to reproduce the paper's analyses and benchmark methods easily. Additionally, the documentation[28] provides walkthrough tutorials and videos for using the GUI as well as information on the *deepflash2* Python API. Please refer to the original article for detailed information on animal experiments and data acquisition.

This chapter covers all elements of the ML life cycle. It particularly emphasizes the *Monitor Predictions* and *Gather and Analyze Insights* phases as the presented DL solution, *deepflash2*, provides capabilities for integrated quality assurance and out-of-distribution detection during inference. This chapter follows the typical structure of a life science article similar to Chapter 3.

[25] `https://arxiv.org/abs/2111.06693`; currently under review at Nature Methods
[26] `https://colab.research.google.com/github/matjesg/deepflash2/blob/master/deepflash2_GUI.ipynb`
[27] `https://github.com/matjesg/deepflash2`
[28] `https://matjesg.github.io/deepflash2/`



Machine learning life cycle stages covered in Chapter 4.

Summary. *deepflash2* is a deep learning solution that facilitates the objective and reliable segmentation of ambiguous bioimages through multi-expert annotations and integrated quality assurance. Thereby, *deepflash2* addresses typical challenges that arise during training, evaluation, and application of deep learning models in bioimaging. The tool is embedded in an easy-to-use graphical user interface and offers best-in-class predictive performance for semantic and instance segmentation under economical usage of computational resources.

4.1 Introduction

Partitioning images into meaningful segments (e.g., cells, cellular compartments, or other anatomical structures) is one of the most ubiquitous tasks in bioimage analysis (Meijering 2020). Segmentation facilitates downstream tasks such as (3D) detection, tracking, quantification, and statistical evaluation of image features. Performing segmentation tasks manually is tedious and time-consuming. Conversely, its automation promises additional insights, more precise analyses, and more rigorous statistics (Falk et al. 2019). DL has

proven to be a flexible method to analyze large amounts of bioimage data (Ronneberger, Fischer, and Brox 2015) and numerous solutions for automated segmentation have been proposed (Falk et al. 2019; Haberl et al. 2018; Berg et al. 2019; Chamier et al. 2021; Bannon et al. 2021; Isensee et al. 2021; Stringer et al. 2021; Lucas et al. 2021). Depending on annotated training data, these tools and analysis pipelines are well suited for settings where the observable phenomena exhibit a high signal-to-noise ratio (SNR), for instance, in monodispersed cell cultures. However, the SNR in bioimages is often low, influenced by experimental conditions, sample characteristics, and imaging trade-offs. Such image material is inherently ambiguous which hampers a reliable analysis. A case in point is the analysis of fluorescent images of complex brain tissue – a core technique in modern neuroscience – which is frequently subject to various sources of ambiguity such as cellular and structural diversity, heterogeneous staining conditions, and challenging image acquisition processes.

With *deepflash2*, we introduce a DL-based analysis tool for fast and reliable segmentation of ambiguous microscopy images. By integrating annotations from multiple experts and providing quality assurance for the analysis of new images, the tool bridges key challenges during model training, evaluation, and application. *Training and evaluation challenges* commence with the manual annotation process. Here, human experts rely on heuristic criteria (e.g., morphology, size, signal intensity) to cope with low SNRs. Relying on a single human expert's annotations for training can result in biased DL models (Segebarth et al. 2020a). At the same time, inter-expert agreement suffers in such settings, which, in turn, leads to ambiguous training annotations (Falk et al. 2019; Niedworok et al. 2016). Without reliable annotations, there is no obvious ground truth, which complicates both model training and evaluation. The *application challenge* emerges when DL models are deployed for analyzing large numbers of bioimages. This scaling-up step is a crucial leap of faith for users as it effectively means delegating control over the study to a black box system. DL models will generate segmentations for any image. However, the segmentation quality is unknown as the reliability of model generalizations beyond the training data cannot be guaranteed. Selecting a representative subset of images for training and evaluation in a single experiment is already challenging. Maintaining a representative training set across multiple experiments with possibly varying conditions compounds these problems and

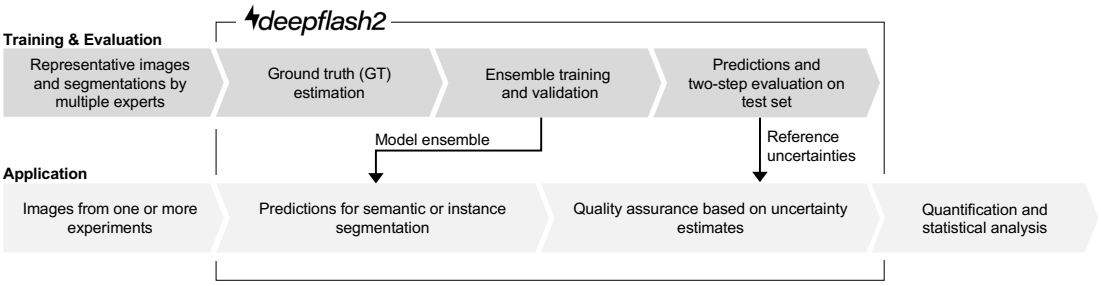


Figure 4.1: *deepflash2* pipelines. Proposed integration into the bioimage analysis workflow.

may eventually prevent reliable automation. For this reason, a viable deployment needs effective quality assurance, or as Ribeiro et al. (Ribeiro, Singh, and Guestrin 2016, p. 1135) put it "if the users do not trust [...] a prediction, they will not use it."

We address these challenges in two consecutive pipelines for *training & evaluation* and *application* that seamlessly integrate into the bioimage analysis workflow (Figure 4.1). We illustrate the capabilities of *deepflash2* using five representative fluorescence microscopy datasets of mouse brain tissue with varying degrees of ambiguity (Figure 4.1, Section 4.2.4). We benchmark the tool against other common analysis tools (Figures 4.2, 4.4, Table 4.1), achieving best-in-class predictive performance under economical usage of computational resources. The *deepflash2* pipelines are embedded in a lightweight and easy-to-use graphical user interface (GUI).

4.2 Methods

The *deepflash2* code library is implemented in Python 3, using *numpy*, *scipy*, and *opencv* for the base operations. The ground truth estimation functionalities are based on the *simpleITK* (Lowekamp et al. 2013). The DL related part is built upon the rich ecosystem of *PyTorch* (Paszke et al. 2019) libraries, comprising *fastai* (Howard and Gugger 2020) for the training procedure, *segmentation models pytorch* (Yakubovskiy 2020) for segmentation architectures, *timm* (Wightman 2019) for pre-trained encoders, and *albumentations* (Buslaev et al. 2020) for data augmentations. Instance segmentation capabilities are complemented using the *cellpose* library (Stringer et al. 2021). The GUI addi-

tionally leverages the Jupyter Notebook environment and *ipywidgets* (Kluyver et al. 2016).

4.2.1 Ground Truth Estimation

To train reproducible and unbiased models *deepflash2* relies on GT estimation from the annotations of multiple experts. *deeepflash2* offers GT estimation via simultaneous truth and performance level estimation (STAPLE) (Warfield, Zou, and Wells 2004) (default in our analyses) or majority voting. This GT estimation is the basis for achieving predictions as close as possible to the unobservable GT. In contrast, recent work on the segmentation of ambiguous data focuses on explicitly modeling the disagreement between the experts (Kohl et al. 2018; Ji et al. 2021), which is not the main objective of our study. Note that, due to the ambiguities in the data, GT estimation can yield biologically implausible results (e.g., by merging the areas of two cells). We corrected such artifacts in our test sets.

4.2.2 Training

Network architecture. Powered by the *segmentation models pytorch* package (Yakubovskiy 2020), *deepflash2* allows the user to select from various architectures, such as U-Net (Ronneberger, Fischer, and Brox 2015), U-Net++ (Zhou et al. 2018), or DeepLabV3+ (Chen et al. 2018b). It also supports a wide range of encoders, e.g., ResNet (He et al. 2016), EfficientNet (Tan and Le 2019), or ResNeSt (Zhang et al. 2020). During the development of *deepflash2* we evaluated the performance of different architecture and encoder combinations. We then chose a U-Net architecture with a ResNet-34 encoder as a baseline for all experiments of this study. There was no combination that outperformed this baseline on all datasets in a stable training regime. However, we found that switching to more current encoders such as ResNeSt can improve the results on some datasets. If available, the encoders can be initialized with pre-trained weights to allow better feature extraction and fast training convergence. The remaining weights are initialized from a truncated normal distribution (He et al. 2015). This approach combines the desirable properties of pretraining and

random initialization that facilitate diversity in model ensembles. We used imagenet (Deng et al. 2009) pre-trained weights in all our experiments.

Training procedure. Each model is trained using the *fine-tune* policy of the *fastai* library (Howard and Gugger 2020). This entails freezing of the encoder weights, *one-cycle training* (Smith 2018) of one epoch, unfreezing the weights, and again *one-cycle-training*. The epochs of the second training cycle depend on the number of training images and are computed such that a fixed number of training iterations is reached. During each epoch, we sample equally sized patches from each image in the training data. To address the issue of class imbalances, we use a weighted random sampling approach that ensures that the center points of the patches are sampled equally from each class. This kind of sampling also contributes to the data augmentation pipeline, along with other random augmentations such as rotating, flipping, and gamma correction. Users can adjust these augmentations or add more augmentations (e.g., contrast limited adaptive histogram equalization or grid distortions). We use the mean of the cross-entropy and Dice loss (Drozdzal et al. 2016) as learning objective. *deepflash2* also provides options for common segmentation loss functions such as Focal (Lin et al. 2017b), Tversky (Salehi, Erdogmus, and Gholipour 2017), or Lovasz (Berman, Triki, and Blaschko 2018). We trained each model with 100 iterations in the first (frozen encoder weights) cycle and 2500 iterations in the second cycle using a mini-batch size of four (patch size 512×512), the Adam optimizer (Kingma and Ba 2015) with decoupled weight decay (0.001) (Loshchilov and Hutter 2019), and a base learning rate of 0.001. The training and validation data for the different models are shuffled by means of a k-fold cross-validation (with $k = 5$ in our experiments). Users can customize all training settings, for example, by opting for a different optimizer or setting a dataset-specific learning rate using the learning rate finder.

4.2.3 Prediction

Semantic segmentation. For the semantic segmentation of a new image with features $\mathbf{X} \in \mathbb{R}^{d \times c}$ *deepflash2* predicts a semantic segmentation map $\mathbf{y} \in \{1, \ldots, K\}^d$, with K being the number of classes, d the dimensions of the input and c the input channels. Without loss of generality class 1 is defined as back-

ground. We use the trained ensemble of M deep neural networks to model the probabilistic predictive distribution $p_\theta(\mathbf{y} \mid \mathbf{X})$, where $\theta = (\theta_1, \ldots, \theta_M)$ are the parameters of the ensemble. Here, we leverage a sliding window approach with overlapping borders and Gaussian importance weighting (Isensee et al. 2021). We improve the prediction accuracy and robustness using T deterministic test-time augmentations (rotating and flipping the input image). Each augmentation $t \in \{1, ..., T\}$ applied to an input image creates an augmented feature matrix $\mathbf{X_t}$. To combine all predictions we follow Lakshminarayanan, Pritzel, and Blundell (2017) and treat the ensemble as a uniformly-weighted mixture model to derive

$$p(\mathbf{y} \mid \mathbf{X}) = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{M} \sum_{m=1}^{M} p_{\theta_m}(\mathbf{y} \mid \mathbf{X_t}, \theta_m) \tag{4.1}$$

with $p_{\theta_m}(\mathbf{y} \mid \mathbf{X_t}, \theta_m) = \mathrm{Softmax}(\mathbf{f}_{\theta_m}(\mathbf{X_t}))$ and $\mathbf{f}_{\theta_m}$ representing the neural network parametrized with θ_m. We use $M = 5$ models and $T = 4$ augmentations in all our experiments. Finally, we obtain the predicted segmentation map

$$\hat{\mathbf{y}} = \underset{\mathbf{k} \in \{1,\ldots,K\}^d}{\arg\max}\ p(\mathbf{y} = \mathbf{k} \mid \mathbf{X}). \tag{4.2}$$

Uncertainty quantification. The uncertainty is typically categorized into *aleatoric* (statistical or per-measurement) uncertainty and *epistemic* (systematic or model) uncertainty (Der Kiureghian and Ditlevsen 2009). To approximate the uncertainty maps of the predicted segmentations we follow the approach of Kwon et al. (2020). Here, we replace the Monte-Carlo dropout approach of Gal and Ghahramani (2016) with deep ensembles, which have proven to produce well-calibrated uncertainty estimates and a more robust out-of-distribution detection (Lakshminarayanan, Pritzel, and Blundell 2017). In combination with test-time augmentations (inspired by Wang et al. 2019) we approximate the predictive (hybrid) uncertainty the for each class $k \in \{1, \ldots, K\}$ as

$$\begin{aligned}\mathrm{Var}_{p(\mathbf{y}=k|\mathbf{X})} \approx & \underbrace{\frac{1}{T}\sum_{t=1}^{T}\frac{1}{M}\sum_{m=1}^{M}\left[p_{\theta_m}\left(\mathbf{y}=k \mid \mathbf{X}_t,\theta_m\right)-p_{\theta_m}\left(\mathbf{y}=k \mid \mathbf{X}_t,\theta_m\right)^2\right]}_{\text{epistemic uncertainty}} \\ & + \underbrace{\frac{1}{T}\sum_{t=1}^{T}\frac{1}{M}\sum_{m=1}^{M}\left[p_{\theta_m}\left(\mathbf{y}=k \mid \mathbf{X}_t,\theta_m\right)-p\left(\mathbf{y}=k \mid \mathbf{X}\right)\right]^2}_{\text{aleatoric uncertainty}}\end{aligned} \tag{4.3}$$

where $p\left(\mathbf{y}=k \mid \mathbf{X}\right)$ denotes probabilities of a single class k. To allow an intuitive visualization and efficient calculation in multi-class settings, we aggregate the results of the single classes to retrieve the final predictive uncertainty map:

$$\mathrm{Var}_{p(\mathbf{y}|\mathbf{X},\theta)} = \frac{1}{K}\sum_{k=1}^{K}\mathrm{Var}_{p(\mathbf{y}=k|\mathbf{X},\theta)} \tag{4.4}$$

Note that, due to symmetries, the results may only differ from the general formulation in Kwon et al. (2020) for $K > 2$.

The *aleatoric* uncertainty is low for probabilities close to zero or one, and high for probabilities around 0.5. This results, for instance, in high *aleatoric* uncertainties for the border regions of the segmented cell nuclei or somata. In contrast, *epistemic* (model) uncertainty foremost captures the uncertainty of planar areas that are entirely ambiguous. In these areas, the models' predictions may differ considerably as a clear distinction between the foreground and background classes is not feasible.

For the heuristic sorting and out-of-distribution detection, we define an aggregated uncertainty metric on image level. Let $\hat{y}_i$ be the predicted segmentation of pixel i, $\mathbf{x}_i$ the feature vector of pixel i and N the total number of pixels defined by d. We define the scalar valued foreground uncertainty score for all predicted $N_f = \left\{i \in \{1,...,N\} \mid \hat{y}_i > 1\right\}$ as

$$U_{p(\mathbf{y}|\mathbf{X},\theta)} := \frac{1}{|N_f|}\sum_{i\in N_f}\mathrm{Var}_{p(y_i|\mathbf{x}_i,\theta)}\,. \tag{4.5}$$

Instance segmentation. If the segmented image contains touching objects (e.g., cells that are in close proximity), *deepflash2* offers an option for reliable

instance segmentation. For this, we use the combined predictions of each class $p(\mathbf{y} = k \mid \mathbf{X})$ to predict the flow representations using the *cellpose* library (Stringer et al. 2021). We then leverage the post-processing pipeline of *cellpose* to derive instance segmentations by combining the flow representations with the predicted segmentation maps $\hat{y}$. This procedure scales to an arbitrary number of classes and is, in contrast to the original *cellpose* implementation, not limited to one (or two) of input channels.

4.2.4 Evaluation

Evaluation metrics. For semantic segmentation, we calculate the similarity of two segmentation masks $\mathbf{y}_a$ and $\mathbf{y}_b$ using the dice score. For binary masks, this metric is defined as

$$DS := \frac{2TP}{2TP + FP + FN}, \tag{4.6}$$

where the *true positives* (TP) are the sum of all matching positive (pixels) elements of $\mathbf{y}_a$ and $\mathbf{y}_b$, and the *false positives* (FP) and *false negatives* (FN) the sum of positive elements that only appear in $\mathbf{y}_a$ or $\mathbf{y}_b$, respectively. In multi-class settings we use *macro* averaging, i.e., we calculate the metrics for each class and then find their unweighted mean. The dice score is commonly used for semantic segmentation tasks but is unaware of different instances (sets of pixels belonging to a class and instance).

For instance segmentation, let $\mathbf{y}_a^I$ and $\mathbf{y}_b^I$ be two instance segmentation masks that contain a finite number of instances I_a and I_b, respectively. An instance I_a is considered a match (*true positive* – TP_η) if an instance I_b exists with an *Intersection of Union* (also known as *Jaccard index*) $IoU(I_a, I_b) = \frac{I_a \cap I_b}{I_a \cup I_b}$ exceeding a threshold $\eta \in [0, 1]$. Unmatched instances I_a are considered as *false positives* (FP_η), unmatched instances I_b as *false negatives* (FN_η). We define the Average Precision at a fixed threshold η as $AP_\eta := \frac{TP_\eta}{TP_\eta + FN_\eta + FP_\eta}$. To become independent of fixed values for η it is common to average the results over different η. The resulting metric is known as *mean Average Precision* and defined as

$$mAP := \frac{1}{|H|} \sum_{\eta \in H} AP_\eta. \tag{4.7}$$

We use a set of 10 thresholds $H = \{\eta \in [0.50, \ldots, 0.95] \mid \eta \equiv 0 \mod 0.05\}$ for all evaluations. This corresponds to the metric used in the COCO object detection challenge (Lin et al. 2014a). Additionally, we exclude all instances I that are below a biologically viable size from the analysis. The minimum size is derived from the smallest area annotated by a human expert: 61 pixel (*PV in HC*), 30 pixel (*cFOS in HC*), 385 pixel (*mScarlet in PAG*), and 193 pixel (*YFP in CTX*).

Evaluation datasets. We evaluate our pipeline on five datasets that represent common bioimage analysis settings. The datasets exemplify a range of fluorescently labeled (sub-)cellular targets in mouse brain tissue with varying degrees of data ambiguity.

The *PV in HC* dataset published by Segebarth et al. (2020a) describes indirect immunofluorescence labeling of Parvalbumin-positive (PV-positive) interneurons in the hippocampus. Morphological features are widely ramified axons projecting to neighbored neurons for soma-near inhibition of excitatory neuronal activity (Hu, Gan, and Jonas 2014). The axonal projections densely wrap around the somata of target cells. This occasionally causes data ambiguities when the somata of the PV-positive neurons need to be separated from the PV-positive immunofluorescent signal around the soma of neighboured cells. Thresholding approaches such as Otsu's method (see Figure 4.4a) typically fail at this task as it requires to differentiate between rather brightly labeled somata that express PV in the cytosol vs. brightly labeled PV-positive axon bundles that can appear in the neighborhood.

The publicly available *cFOS in HC* dataset (Segebarth et al. 2020b) describes indirect immunofluorescent labeling of the transcription factor cFOS in different subregions of the hippocampus after behavioral testing of the mice (Segebarth et al. 2020a). The counting or segmentation of cFOS-positive nuclei is an often used experimental paradigm in the neurosciences. The staining is used to investigate information processing in neural circuits (Ruediger et al. 2011). The low SNR of cFOS labels for most but not all image features renders its heuristic segmentation a very challenging task. This results in a very high inter-expert variability after manual segmentation (see (Segebarth et al. 2020a).We use 280 additional images of this dataset to demonstrate the out-of-distribution detection capabilities of *deepflash2*. There are no expert annotations available for the additional images, however, 24 images com-

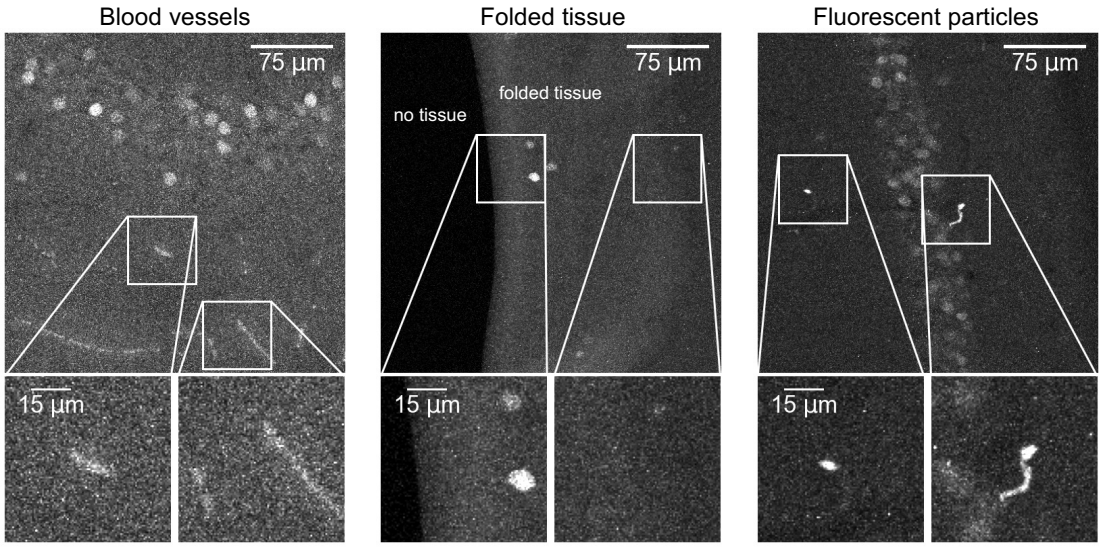


Figure 4.1: Partly out-of-distribution images. Image crops and zoom-ins of three error categories in the extended *cFOS in HC* dataset.

prise characteristics that do not occur in the training data. Such partly out-of-distribution images exhibit elements that may distort the analysis, e.g., blood vessels, folded tissue, and fluorescent particles. Representative samples of these categories are depicted in Figure 4.1.

The optogenetic *mScarlet in PAG* dataset shows an indirect immunofluorescent post-labeling of the red-fluorescent protein mScarlet, after viral expression in the peri-aqueductal gray. Here, microscopy images visualize mScarlet, tagged to the light-sensitive inhibitory opsin OPN3. The recombinant protein was delivered via stereotactic injection of an adeno-associated viral vector to the PAG. Recombinant opsins are often used for optogenetics, a key technology in the neurosciences (Rost et al. 2017). This allows control of neuronal activity in selected neuron populations. Due to a plethora of factors (e.g. virus injection, virus titer at the locus of injection, experiment-specific success), the resulting (sub-)cellular distribution and intensity of the fluorescent signals results in highly ambiguous images, such as those depicted in Figure 4.1. This renders the segmentation of infected neurons in optogenetics challenging and tedious (Falk et al. 2019).

The *YFP in CTX* dataset shows direct fluorescence of yellow fluorescent protein (YFP) in the cortex of so-called thy1-YFP mice. In thy1-YFP mice, a fluores-

Table 4.1: Comparison of datasets

	PV in HC	cFOS in HC	mScarlet in PAG	YFP in CTX	GFAP in HC
Annotation target	somata	nuclei	somata	somata	morphology
Semantic segmentation	yes	yes	yes	yes	yes
Instance segmentation	yes	yes	yes	yes	no
Train images	36	36	12	12	12
Test images	8	8	8	8	8
Experts	5	5	4-5	4-5	3
Additional images	–	280	–	–	–
Fluorescence Microsc.	confocal	confocal	light	light	light
Size (pixel)	1024×1024	1024×1024	2752×2208	2752×2208	580×580
Resolution (px/µm)	1.61	1.61	3.7	3.7	3.7

cent protein is expressed in the cytosol of neuronal subtypes with the help of promoter elements from the thy1 gene (Feng et al. 2000). This provides a fluorescent Golgi-like vital stain that can be used to investigate disease-related changes in neuron numbers or neuron morphology, for instance for hypothesis-generating research in neurodegenerative diseases (e.g. Alzheimers disease). Here, computational bioimage analysis is aggravated by the pure intensity of the label that causes strong background signals by light scattering or out-of-focus light. Both can blur the signal borders in the image plane.

Finally, the *GFAP in HC* dataset shows indirect immunofluorescence signals of glial acidic fibrillary protein (GFAP) in the hippocampus. Anti-GFAP labeling is one of the most commonly used stainings in the neurosciences and is also used for histological examination of brain tumor tissue. Here, the extensions of the GFAP-labeled astrocytic skeleton cannot be separated from parts of neighboring astrocytes, rendering a reliable instance separation and thus instance segmentation impossible. Albeit the signal is typically bright and very clear around the center of the cell, the signal borders of the radial fibers become ambiguous due to the 3D-ball-like structure, low SNR at the end of the fibers, and out-of-focus light interference. Table 4.1 provides a high-level comparison of the key dataset characteristics.

Performance benchmarks. We benchmark the predictive performance of *deepflash2* against a select group of well-established algorithms and tools. These comprise the U-Net of (Falk et al. 2019) and *nnunet* (Isensee et al. 2021) for both semantic and instance segmentation as well as two out-of-the-box

baselines. We utilize Otsu's method (Otsu 1979) as a simple baseline for semantic segmentation and *cellpose* (Stringer et al. 2021) as a generic baseline for (cell) instance segmentation. Additionally, we benchmark *deepflash2* against fine-tuned *cellpose* models and ensembles, showing superior performance of our method (Table 4.1). *cellpose* has previously proven to outperform other well-known methods for instance segmentation, e.g., Mask-RCNN (He et al. 2017a) or StarDist (Schmidt et al. 2018).

For each dataset, we apply the tools as described by their developers to render the comparison as fair as possible. We train the U-Net of (Falk et al. 2019) on a 90/10 train-validation-split for 10,000 iterations (learning rate of 0.00001 and the Adam optimizer (Kingma and Ba 2015) using the authors' *TensorFlow 1.x* implementation. This includes all relevant features such as overlapping tile strategy and border-aware loss function. We derive the parameter values for the loss function (border weight factor (λ), border weight sigma (σ), and foreground-background-ratio (v) by means of Bayesian hyperparameter tuning: PV in HC: λ=25, σ=10, v=0.66; cFOS in HC: λ=44, σ=2, v=0.23; mScarlet in PAG: λ=15, σ=10, v=0.66; YFP in CTX: λ=15, σ=5, v=0.85; GFAP in HC: λ=1, σ=1, v=0.85.

We train the self-configuring *nnunet* (version 1.6.6) model ensemble (Isensee et al. 2021) following the authors' instructions provided on GitHub.

cellpose provides three pretrained model ensembles (*nuclei*, *cyto*, and *cyto2*) for out-of-the-box usage (Stringer et al. 2021). We select the ensemble with the highest score on the training data: *cyto* for PV in HC and YFP in CTX; *cyto2* for cFOS in HC and mScarlet in PAG. During inference we fix the cell diameter (in pixel) for each dataset: PV in HC: 24; cFOS in HC: 15; mScarlet in PAG: 55; YFP in CTX: 50. In addition to the out-of-the-box *cellpose** approach, (Stringer et al. 2021) we include fine-tuned *cellpose* models and ensembles in our comparison. We train the *cellpose* models via five-fold cross-validation with the default training settings from the command line interface (500 epochs, 0.2 learning rate, batch size of 8). The resulting *cellpose* ensemble consists of five models similar to the *deepflash2* model ensembles. As the *cellpose* command-line interface does not implement training via cross-validation, we also include models trained on a single train-validation split into our analysis. These models are simply selected from the trained model ensembles. We use

the *cellpose* GitHub version with commit hash 316927e (August 26, 2021) for our experiments.

We repeat our experiments with different seeds to ensure that our results are robust and reproducible. The experiments for training duration comparison are executed on the *free* platform Google Colaboratory (Nvidia Tesla K80 GPU, 2 vCPUs; times were extrapolated when the 12-hour limit was reached) and the *paid* Google Cloud Platform (Nvidia A100 GPU, 12 vCPUs). The remaining experiments are executed locally (Nvidia GeForce RTX 3090) or in the cloud (Google Cloud Platform on Nvidia Tesla K40 GPUs).

4.2.5 Quality Assurance

Once the *deepflash2* model ensemble is deployed for predictions on new data, the quality assurance process helps the user prioritize the review of more ambiguous or out-of-distribution images. The predictions on such images are typically error-prone and exhibit a higher uncertainty score U. Thus, *deepflash2* automatically sorts the predictions by decreasing uncertainty score. Depending on the ambiguities in the data and the expected prediction quality (inferred from the hold-out test set), a conservative protocol could require scientists to verify all images with an uncertainty score exceeding a threshold U_{min}. Given the the hold-out test set $Q = \{(\mathbf{X}_1, \mathbf{y}_1), \ldots, (\mathbf{X}_L, \mathbf{y}_L)\}$ where L is the number of samples we define

$$U_{min} := \min \left\{ U_{p(\mathbf{y}|\mathbf{X},\theta)} \mid (\mathbf{y}, \mathbf{X}) \in Q, S(y, \hat{y}) < \tau \right\}. \tag{4.8}$$

with $S(y, \hat{y})$ being an arbitrary evaluation metric (e.g., DS or mAP) and $\tau \in [0, 1]$ a threshold that satisfies the prediction quality requirements. From a practical perspective, this means selecting all predictions from the test set with a score below the pre-defined threshold (e.g., $DS = 0.8$) and taking their minimum uncertainty score value U as U_{min}. The verification process of a single image is simplified by the uncertainty maps that allow the user to quickly find difficult or ambiguous areas within the image.

4.3 Results

Training and evaluation build upon a representative sample of the bioimage dataset under analysis, annotated by multiple experts (the annotations can be performed with any tool). Depending on the biological analysis setting, we distinguish between *semantic* and *instance* segmentation. Semantic segmentation means subdividing the image into meaningful categories (Falk et

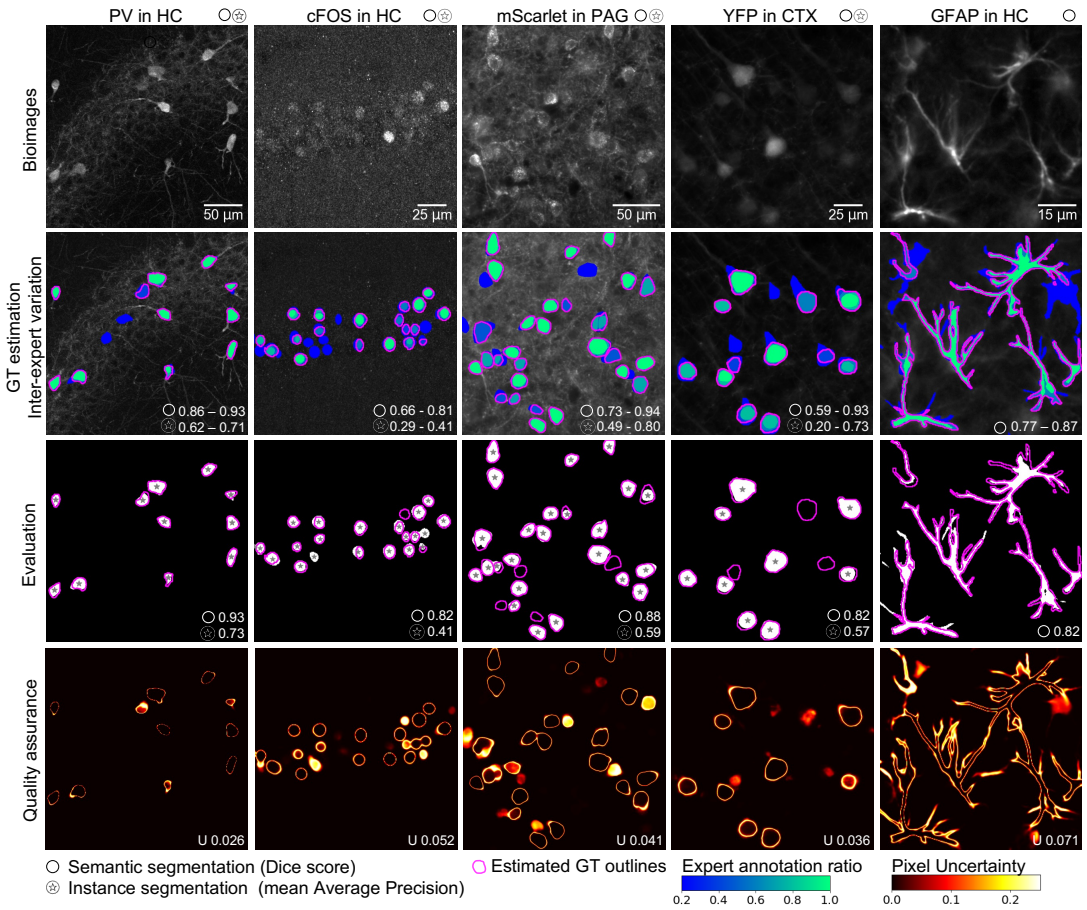


Figure 4.1: Ambiguous bioimaging datasets. Representative image sections from the test sets of five immunofluorescence imaging datasets (first row) with corresponding expert annotations and ground truth (GT) estimation (second row). The predicted segmentations and the similarity to the estimated GT are depicted in the third row, the corresponding uncertainty maps, and uncertainty scores U for quality assurance in the fourth row. Areas with a low expert agreement (blue) or differences between the predicted segmentation and the estimated GT typically exhibit high uncertainties. The maximum pixel uncertainty is limited to 0.25.

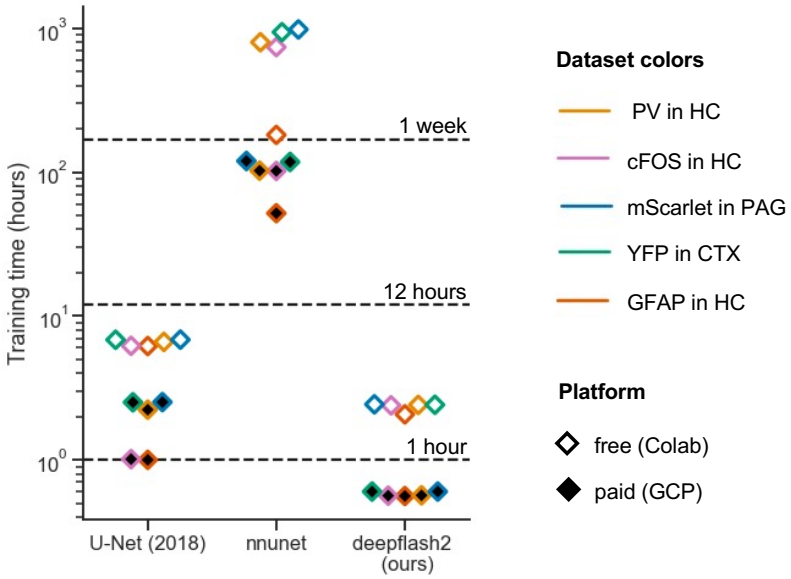


Figure 4.2: Training duration comparison. Training durations on different platforms: Google Colaboratory (Colab, free Nvidia Tesla K80 GPU) and Google Cloud Platform (GPC, paid Nvidia A100 GPU).

al. 2019). Instance segmentation further differentiates between multiple instances of the same category by assigning the segmented structures to unique entities (e.g., cell 1, cell 2, ...). To derive objective training annotations from multi-annotator data *deepflash2* estimates the ground truth (Section 4.2.1) via STAPLE. *deepflash2* subsequently computes the similarity scores (Section 4.2.4) between expert segmentations and the estimated GT. These measures of inter-expert variation serve as a proxy for data ambiguity as illustrated in Figure 4.1 (first and second row). Well-defined fluorescent labels are typically unanimously annotated (green), whereas more ambiguous signals are marked by fewer experts (blue).

DL model training in *deepflash2* capitalizes on model ensembles to ensure high accuracy and reproducibility in the light of data ambiguity (Segebarth et al. 2020a). Furthermore, it facilitates reliable uncertainty quantification (Lakshminarayanan, Pritzel, and Blundell 2017). To ensure training efficiency *deepflash2* leverages pre-trained models and advanced training strategies (Section 4.2.2). This approach yields very competitive training durations (Figure 4.2).

The model ensemble then predicts semantic segmentation maps which are evaluated on a hold-out test set (Figure 4.1, third row). If required, *deepflash2*

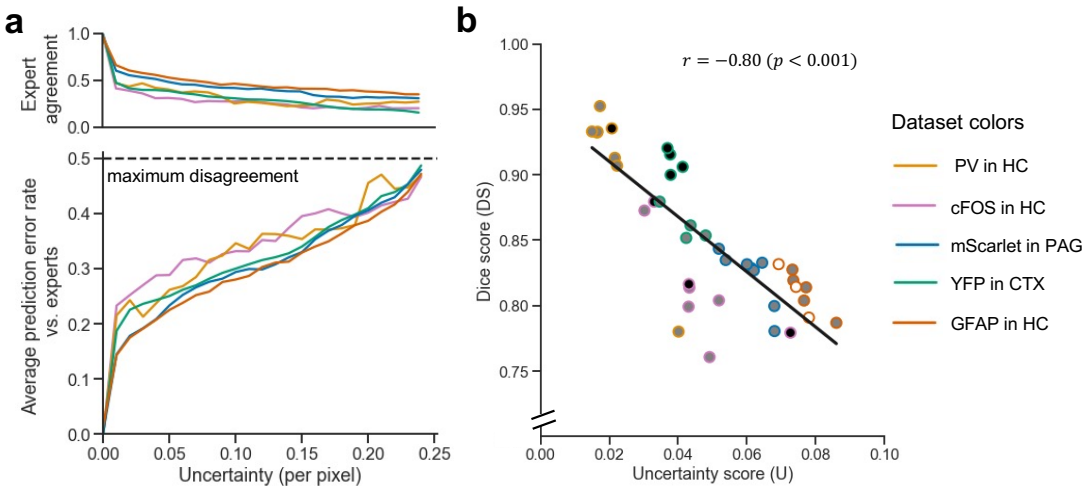


Figure 4.3: Uncertainty evaluation. (**a**) Relationship between pixel-wise uncertainty and expert agreement (at least one expert with differing annotation; upper plot) and average prediction error rate (relative frequency of deviations between different expert segmentations and the predicted segmentation; lower plot) on the test set. (**b**) Correlation between dice scores and uncertainties on the test set (right). We quantify the linear correlation using Pearson's r and a two-tailed p-value for testing non-correlation.

combines these maps with the flow representations of the *cellpose* library to derive reliable instance segmentations (i.e., separation of touching objects). Each segmentation is accompanied by a predictive uncertainty map and the average foreground uncertainty score U (Figure 4.1, fourth row; Section 4.2.3). Note that the model ensembles are solely trained on the estimated GT, that is, there is no longer a concept of ambiguous annotations. However, Figure 4.3a confirms that the uncertainty maps reliably capture expert disagreement: Low pixel uncertainty is indicative of high expert agreement, whereas high pixel uncertainty arises in settings where experts submitted ambiguous annotations.

To assess the model validity for bioimage analysis, *deepflash2* implements the following two-step evaluation process:

1. *Calculate the similarity scores* between the predicted segmentations and the estimated GT on the test set.

2. *Relate the performance scores to data ambiguity*. The experts' performance scores are used to establish the desired performance range.

4.3.1 Segmentation Performance

Across all evaluation datasets, we find that *deepflash2* achieves best-in-class performance vis-a-vis state-of-the-art benchmark tools for both semantic (Figure 4.4a) and instance segmentation (Figure 4.4b) tasks. To ensure that our results are robust and reproducible we repeat our experiments with different seeds. This changes the train-validation splits and weight initialization for each repetition. The results of three experiment repetitions for all methods are depicted in Table 4.1.

The results show that the ensemble-based methods *nnunet* and *deepflash2* yield very stable results (low std. deviations) across all datasets, while the U-Net of (Falk et al. 2019), based on a single model, is subject to higher performance variability. We also report the results of the detection task, which is commonly measured by the $AP_{IoU=0.50}$. In contrast to the mAP that provides

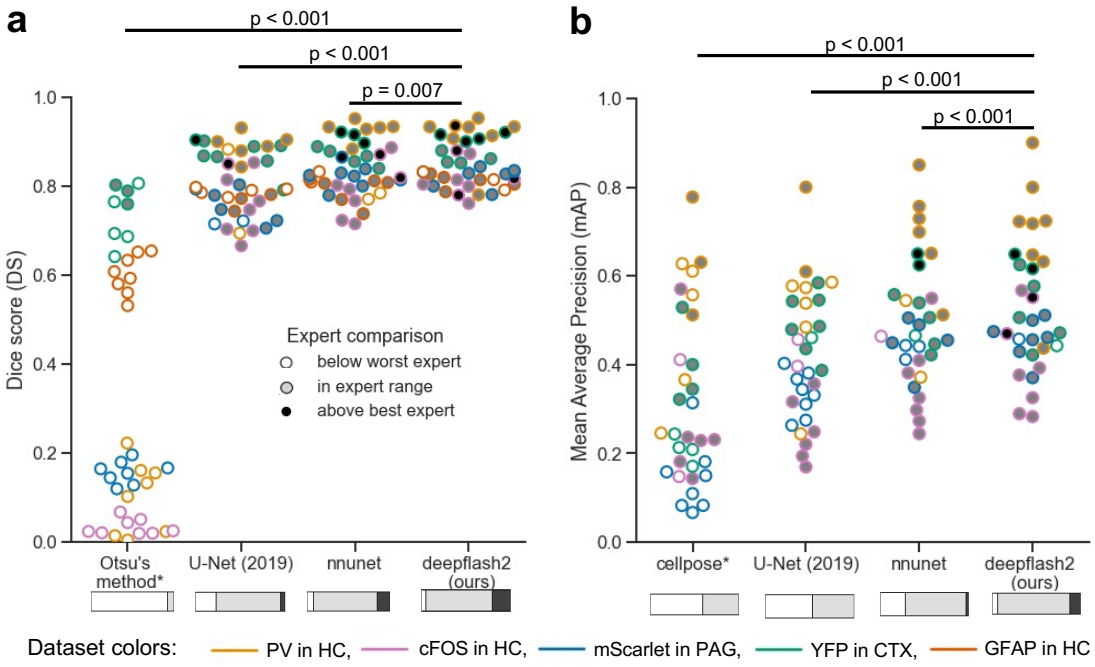


Figure 4.4: Segmentation performance comparison. Predictive performance on the test sets for (**a**) semantic segmentation (N=40) and (**b**) instance segmentation (N=32), measured by similarity to the estimated GT. The grayscale filling depicts the comparison against the expert annotation scores. The p-values result from a two-sided Wilcoxon signed-rank test. *indicates out-of-the-box methods (not fine-tuned to the respective dataset)

Table 4.1: Detailed performance comparison. Average predictive performance measured by similarity to the estimated ground truth (STAPLE) on the hold-out test sets (N=8 images for each dataset) over three repetitions. *indicates out-of-the-box methods (not fine-tuned to the respective dataset)

	PV in HC	cFOS in HC	mScarlet in PAG	YFP in CTX	GFAP in HC
Method	Semantic Segmentation - Mean DS (std. deviation)				
Otsu*	0.101 (–)	0.033 (–)	0.156 (–)	0.743 (–)	0.600 (–)
U-Net (2019)	0.863 (0.010)	0.769 (0.010)	0.756 (0.008)	0.850 (0.037)	0.762 (0.015)
nnunet	0.891 (0.002)	0.797 (0.000)	0.821 (0.002)	0.883 (0.000)	0.799 (0.000)
deepflash2 (ours)	**0.914** (0.003)	**0.813** (0.002)	**0.824** (0.002)	**0.886** (0.002)	**0.811** (0.001)
	Instance Segmentation - Mean mAP (std. deviation)				
cellpose*	0.541 (–)	0.268 (–)	0.148 (–)	0.304 (–)	– (–)
cellpose (single)	0.610 (0.012)	0.329 (0.010)	0.415 (0.004)	0.499 (0.014)	– (–)
cellpose (ensemble)	0.628 (0.028)	0.350 (0.004)	0.432 (0.002)	0.511 (0.010)	– (–)
U-Net (2019)	0.548 (0.024)	0.305 (0.016)	0.337 (0.003)	0.455 (0.059)	– (–)
nnunet	0.643 (0.004)	0.368 (0.002)	0.443 (0.002)	0.527 (0.003)	– (–)
deepflash2 (ours)	**0.696** (0.007)	**0.404** (0.004)	**0.460** (0.003)	**0.538** (0.001)	– (–)
	Detection - $AP_{IoU=0.50}$ (std. deviation)				
cellpose*	0.701 (–)	0.404 (–)	0.237 (–)	0.536 (–)	– (–)
cellpose (single)	0.844 (0.025)	0.662 (0.008)	0.666 (0.011)	0.805 (0.027)	– (–)
cellpose (ensemble)	0.851 (0.031)	0.688 (0.015)	0.686 (0.009)	0.823 (0.018)	– (–)
U-Net (2019)	0.844 (0.016)	0.566 (0.017)	0.573 (0.004)	0.755 (0.044)	– (–)
nnunet	0.825 (0.003)	0.647 (0.011)	0.670 (0.003)	0.807 (0.005)	– (–)
deepflash2 (ours)	**0.857** (0.008)	**0.695** (0.003)	**0.713** (0.008)	**0.824** (0.004)	– (–)

a measure for the quality of the segmentation, the $AP_{IoU=0.50}$ metric measures the "counting" performance of a method. That is, for instance, whether the same cell is annotated or not. The *cellpose* ensemble performs on par with the *deepflash2* model on the detection task on the *YFP in CTX* dataset. Fine-tuned *cellpose* models yield similar results to *deepflash2* at low IoU-thresholds η but constantly perform worse for higher thresholds (see Figure 4.5).

To disentangle the difficulty of the prediction task (driven by data ambiguity) from the predictive performance we scrutinize the "gross" performance by comparing it against the underlying expert annotation scores (Figure 4.4). We find that *deepflash2* reliably achieves human expert performance and in some cases even outperforms the *best* available expert annotation. The U values of the test set serve as a reference for the quality assurance procedure for the application step (Section 4.2.5). We find that the uncertainty score U is a strong predictor for the obtained predictive performance as measured by the dice score (Figure 4.3b). Consequently, U can be used as a proxy for the expected performance on unlabeled data.

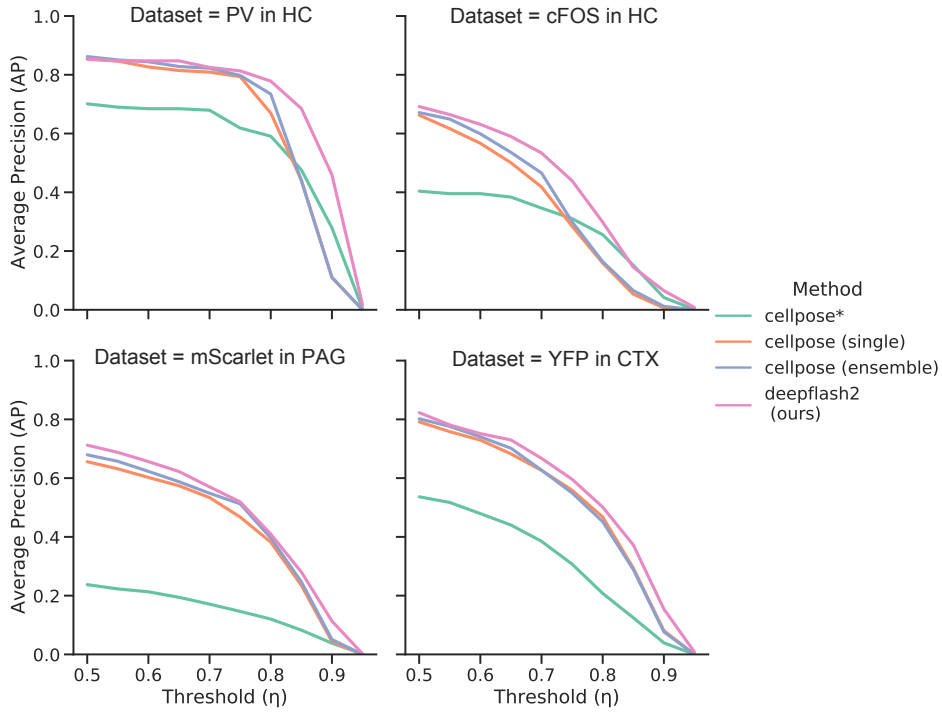


Figure 4.5: *deepflash2* vs. *cellpose* The lines depict the instance segmentation performance using the mean of the Average Precision (N=8 hold-out test images for each dataset) at a certain IoU-threshold η over 3 repetitions. The fine-tuned *cellpose* models (single and ensemble) yield similar results to *deepflash2* at low η but constantly perform worse at higher η.

4.3.2 Quality Assurance

During application, scientists typically aim to analyze a large number of bioimages without ground truth information. To establish trust in its predictions, *deepflash2* enables quality assurance in the following manner: First, the predictions are sorted by decreasing uncertainty score. In situations with high uncertainty scores, scientists may want to check predictions through manual inspection using the provided uncertainty maps. Examples of the different uncertainty types are depicted in Figure 4.6. Also, *deepflash2* facilitates a single click export-import to ImageJ ROIs (regions of interest), with ROIs sorted by uncertainty. This quality assurance process helps the user prioritize the review of more ambiguous instances. Moreover, it facilitates the detection of out-of-distribution images, i.e., images that differ from the training data and are thus prone to erroneous predictions. We showcase the out-of-distribution

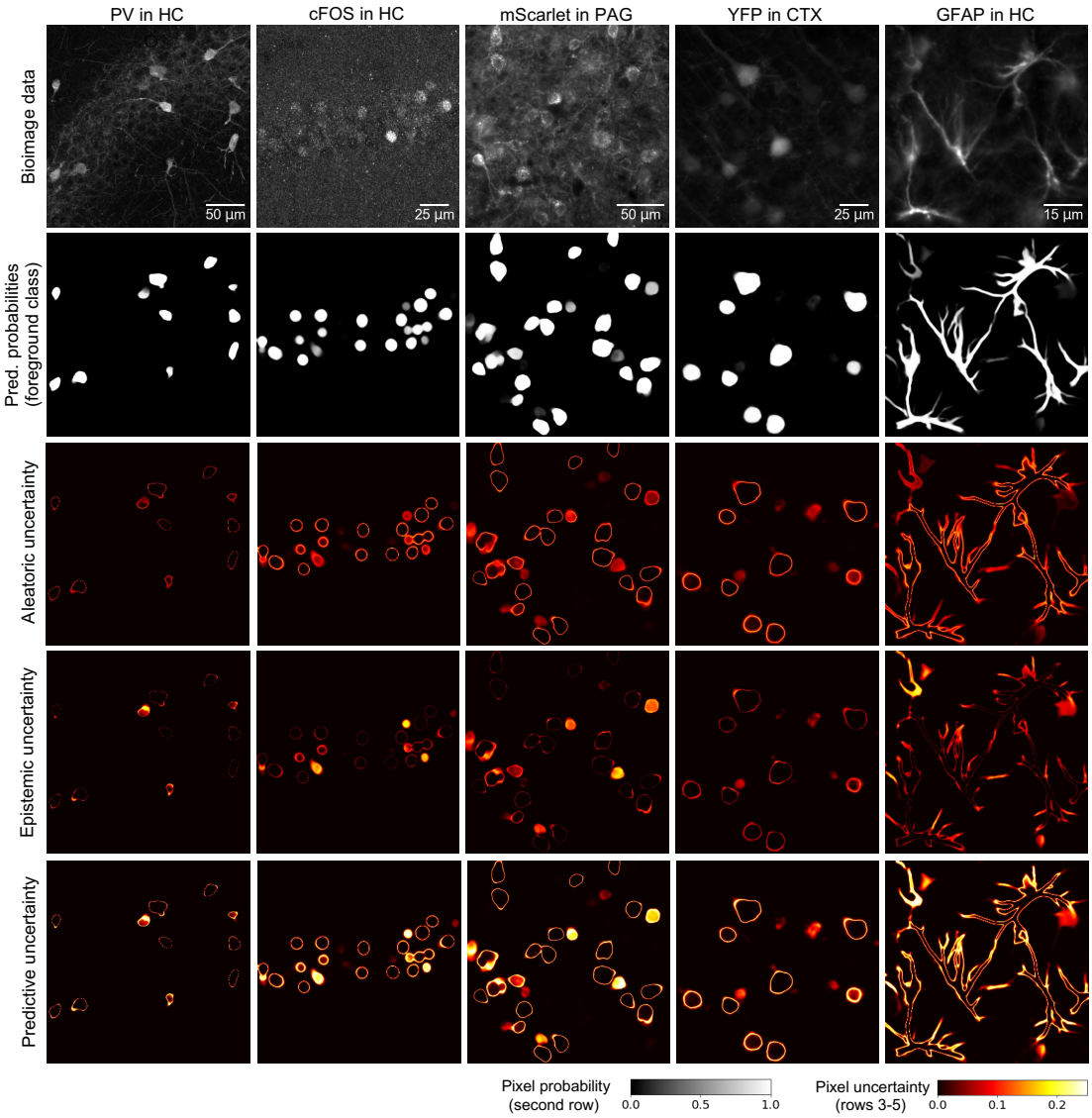


Figure 4.6: Visualization of uncertainty types. Representative image sections from the test sets of five immunofluorescence imaging datasets (first row) and corresponding ensemble probability map $p(\mathbf{y} = k \mid \mathbf{X})$ for the foreground class k (second row). The *aleatoric* (third row) and *epistemic* (fourth row) uncertainty maps are combined into the *predictive* uncertainty map $Var_{p(\mathbf{y}|\mathbf{X},\theta)}$ (fifth row). *Aleatoric* uncertainties foremost emerge in the border regions of the segmented cell nuclei or somata. High *epistemic* uncertainties typically occur in planar areas where a clear distinction between the foreground and background classes is not feasible. *deepflash2* computes the *predictive* uncertainty by default. The maximum pixel uncertainty is limited to 0.25.

detection on a large bioimage dataset comprising 256 in-distribution images (same properties as training images, Figure 4.7b), 24 partly out-of-distribution images (same properties with previously unseen structures such as blood vessels, Figure 4.7c), and 32 fully out-of-distribution images (different immunofluorescent labels, Figure 4.7d). Using the uncertainty score for sorting, all fully

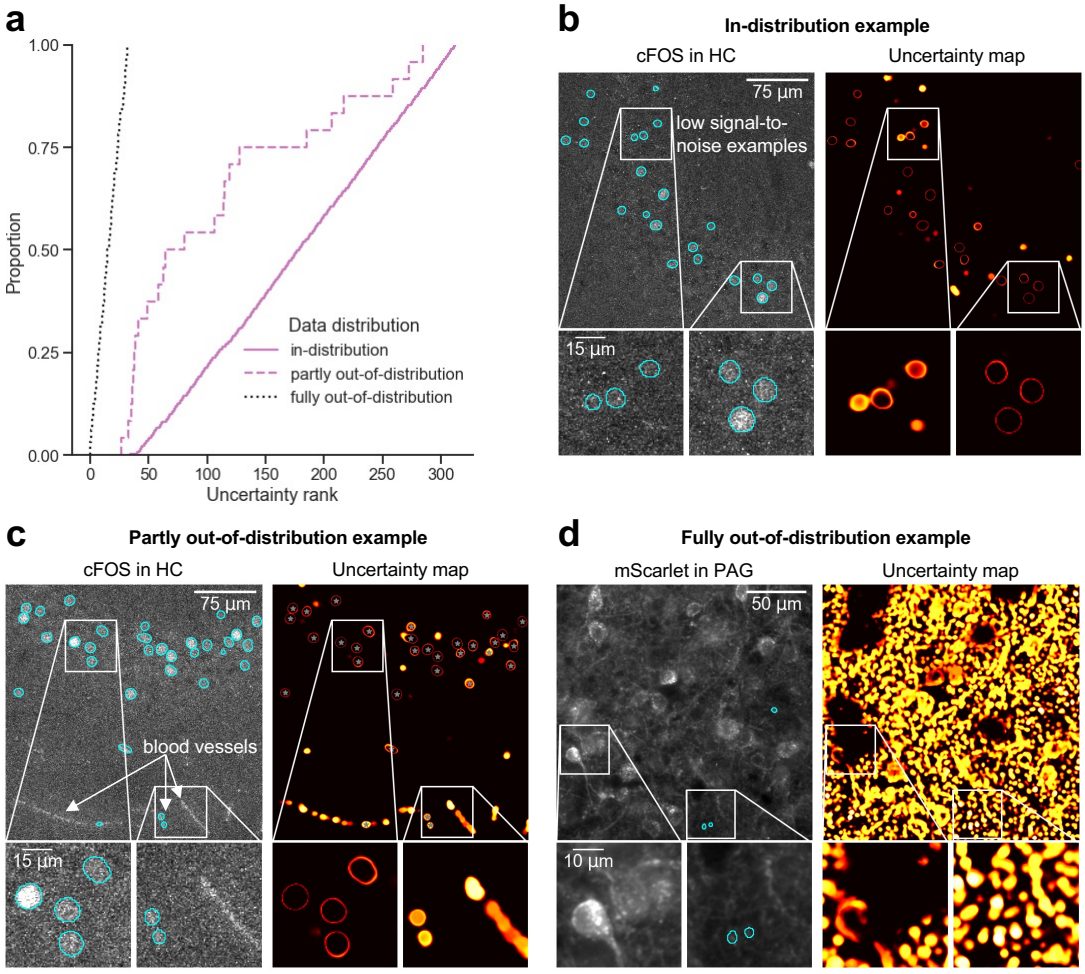


Figure 4.7: Out-of-distribution detection. (**a**) Out-of-distribution (*ood*) detection performance using heuristic ranking via uncertainty score. Starting the manual verification of the predictions at the lowest rank all images with deviant fluorescence labels (fully *ood*, N=32 images) are detected first. The partly *ood* images (with previously unseen structures, N=24) are mostly located in the lower ranks and the in-distribution images (similar to training data of cFOS in HC, N=264) in the upper ranks. (**b**, **c**, **d**) Representative image crops of the three categories used in **a**.

out-of-distribution images are ranked within the first 32 ranks, and most partly out-of-distribution images are ranked within the first 150 ranks (Figure 4.7a). A conservative protocol could require scientists to verify all images with an uncertainty score exceeding the reference uncertainty scores (Section 4.2.5). Out-of-distribution images may then be excluded from the analysis or annotated for re-training in an active learning manner (Gal, Islam, and Ghahramani 2017).

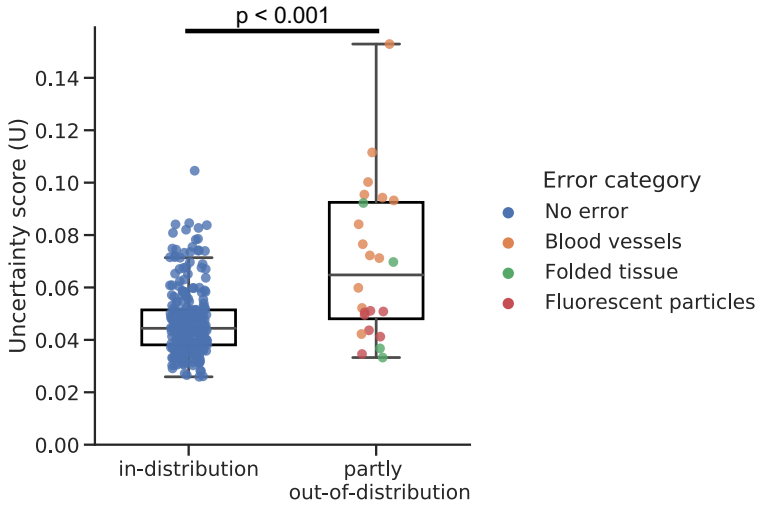


Figure 4.8: Uncertainty scores and out-of-distribution error categories. Uncertainty score comparison for the *cFOS in HC* out-of-distribution dataset. The p-value results from a two-sided non-parametric Mann–Whitney *U* test.

A more detailed analysis in Figure 4.8 reveals that the uncertainty scores U of the partly out-of-distribution images are significantly higher than the uncertainty scores of the in-distribution images (No error). However, the error categories are not distributed evenly. On the one hand, blood vessels and folded tissue images cover a high and relatively wide range of uncertainty scores. On the other hand, fluorescent particle images exhibit uncertainty scores close to the median of the in-distribution images. A possible explanation is the small proportion of unseen structures (a single strongly fluorescent particle unrelated to the actual fluorescent label) in these images. Using the proposed heuristic search strategy (Figure 4.7) such images would be detected at a later stage. However, the inclusion of such images into the bioimage analysis would possibly not impair the results.

4.4 Discussion

Our deep learning solution facilitates the objective and reliable segmentation of ambiguous bioimages through multi-expert annotations and integrated quality assurance. The GUI of *deepflash2* runs as a web application inside a Jupyter Notebook, the de-facto standard of computational notebooks in the scientific community (Perkel 2018). *deepflash2* can be installed locally or in cloud environments such as Google Colaboratory (Colab), enabling quick setup in less than a minute and providing free access to graphics processing units (GPUs) for fast training. In contrast to other tools that are only optimized for Colab (e.g., *ZeroCostDL4Mic* Chamier et al. 2021), the *deepflash2* GUI is based on interactive HTML widgets (Kluyver et al. 2016), providing a clean interface and limiting cognitive overload by leveraging tool-tips and links to the documentation[29]. *deepflash2* allows users to execute all analysis steps directly in the GUI or use the export functionality to Fiji or spreadsheet software for a more detailed analysis. The GUI is built on top of the *deepflash2 Python API* and leverages established open-source libraries in the *PyTorch* (Paszke et al. 2019) ecosystem (Section 4.2). While our evaluation data focuses on ambiguous fluorescent images, *deepflash2* can be used for 2D images with an arbitrary number of input channels. For example, the *deepflash2 Python API* was successfully used on periodic acid–Schiff stained 3-channel images. The implementation won a Gold Medal and the Innovation Award in a Kaggle data science competition hosted by the HuBMAP consortium (Section 2.5).

In summary, *deepflash2* offers an end-to-end integration of DL pipelines for bioimage analysis of ambiguous data. An easy-to-use GUI allows researchers without programming experience to rapidly train performant and robust DL model ensembles and monitor their predictions on new data. We are confident that *deepflash2* can help establish more objectivity and reproducibility in natural sciences while lowering the overall workload for human annotators.

[29] `https://matjesg.github.io/deepflash2/`

5 Conclusion and Outlook

Novel DL architectures, better data availability, and significant increases in computing power have enabled AI researchers to solve problems that were considered unassailable for many years. And yet, the 2020's McKinsey Global Survey on AI (Balakrishnan et al. 2020) revealed that still only 16 percent of the responding companies had adopted DL models in at least one business function, and most projects never got beyond the piloting stage (see Chapter 1). In light of these observations, I defined the guiding research objective of this thesis as *solving business and societal problems by developing rigorous deep learning models* (Section 1.1). Consequently, this thesis examines the challenges encountered throughout the life cycle of DL projects and proposes solutions to develop objective and reliable DL models.

5.1 Prototype, Productionize, Measure

The three main concepts of the ML life cycle – Prototype, Productionize, Measure – form the backbone of this thesis (Figure 1.2). Following these concepts, the main contributions and key findings are summarized below.

5.1.1 Deep Learning Prototypes

Chapter 2 is primarily concerned with prototyping DL solutions for problems from different domains. The prototypes are all based on DL architectures with convolutional layers and mainly address different kinds of image recognition tasks. Therefore, I conceptualize guidelines for applied image recognition in Section 2.1. These guidelines span task definition, deep neural network configuration, and training procedures. I showcase the guidelines by means of a biomedical research project that aims to automate the segmentation of fluo-

rescent neurons in microscopy images. The case study demonstrates the potential of DL in biomedical applications, which I further explore in Chapters 3 and 4. Section 2.2 illustrates the bottom-up development of a DL backend for an augmented intelligence system in the manufacturing sector. A wearable device equipped with highly sensitive sensors is paired with a convolutional neural network to monitor connector system assembly processes in real-time.

Turning to the fashion domain in Section 2.3, I present an artificial curation system for individual outfit recommendations that leverages DL techniques and unstructured data from social media and fashion blogs. Here, I lay out the artifact design and provide a comprehensive evaluation strategy to assess the system's utility. I also demonstrate the capabilities of a DL prototype for fashion image segmentation. The study's findings may inspire the design of DL solutions for other use cases, such as fashion style assessment, recommendations in online shops, or curated design of new collections. In Section 2.4 I map (primarily DL-enabled) AI solutions against typical challenges in creative processes and highlight the unparalleled capabilities of GANs in content creation. On the one hand, AI promotes diverging thinking by creating various design options. On the other hand, AI approaches offer tools for informative summarization and visualization, which in turn can inform decision-making processes. To further emphasize the potential of AI-augmented creativity processes, I showcase several possible applications facilitating and improving the traditional fashion design process. Finally, I present my award-winning solution for the segmentation of glomeruli in human kidney tissue images (Section 2.5). The DL prototype was developed for the Kaggle data science competition *HuBMAP - Hacking the Kidney*. The main contributions comprise a model agnostic sampling strategy that enables fast and reliable model training and energy-based uncertainty scores that facilitate a semi-automated annotation.

To date, only one of the presented prototypes from Chapter 2 has been productionized. This situation somehow illustrates the fact that many promising DL projects never get deployed, in spite of their potential.

5.1.2 From Prototype to Production

Chapter 3 continues the development path of the biomedical research project of Section 2.1 beyond the prototyping phase. Here, I investigate how different

data annotation and training strategies affect the objectivity, reliability, and validity of DL-based bioimage analysis. Training DL models on subjective annotations may be unstable or yield biased models. Consequently, these models may be unable to detect biological effects reliably. An analysis pipeline integrating data annotation, ground truth estimation, and model training can mitigate this risk. To evaluate this integrated process, I compare different DL-based analysis approaches. With data from two model organisms and five laboratories, I show that ground truth estimation from multiple human annotators helps to establish objectivity in fluorescent feature annotations. Furthermore, ensembles of multiple models trained on the estimated ground truth establish reliability and validity.

In terms of the ML life cycle, Chapter 3 covers both the development of various prototypes and their deployment. The predictions of these models are used for further statistical analysis of biological effects.

5.1.3 Considering the entire Machine Learning Life Cycle

Based on the findings described in Chapters 2 and 3, I present a DL solution in Chapter 4 that addresses typical challenges encountered throughout the entire ML life cycle. The DL pipeline, named *deepflash2*, facilitates the objective and reliable segmentation of ambiguous bioimages through multi-expert annotations. Notably, it emphasizes the life cycle phases *Monitor Predictions* and *Gather and Analyze Insights* as it provides capabilities for integrated quality assurance and out-of-distribution detection during inference. *deepflash2* is embedded in an easy-to-use graphical user interface and offers best-in-class predictive performance for semantic and instance segmentation under economical usage of computational resources.

5.2 Future Research Directions

In the rapidly expanding field of DL, the potential for future research remains extensive. As a major part of this thesis is concerned with biomedical image segmentation, I first outline the most promising research directions in this field and continue with an outlook on GANs in the creative process.

5.2.1 Deep Learning based Bioimage Analysis

Integration of transformers. The transformer architecture (Vaswani et al. 2017) is presumably the most impactful DL architecture of the past years. Based on transformers, BERT (Devlin et al. 2019) and its successors continue to set new standards in natural language processing (NLP) tasks. Transformers are also increasingly being applied in fields such as speech, vision, and reinforcement learning. For biomedical image segmentation, traditional DL architectures (e.g., variants of the U-Net) have long prevailed and still deliver excellent results. However, novel architectures, such as the SegFormer (Xie et al. 2021), are challenging their position. To be prepared for future developments, I integrated the *transformers* library of Hugging Face[30] into a development version of deepflash2[31]. This enables easy integration of the pre-trained SegFormer (or other models from the Hugging Face Model Hub) into the bioimage analysis workflow. However, preliminary experiments with the SegFormer have not yielded competitive results on the data from Chapter 4.

Self-supervised learning. As outlined in Chapter 3, DL models should be trained on data annotations that are as objective as possible. The acquisition of such annotations is, however, tedious and eventually expensive. The self-supervised learning (SSL) paradigm bypasses the data annotation requirements by predicting parts of the input from other parts of the input. This concept has profoundly impacted NLP as it allows pre-training on large corpora of unlabeled data (e.g., training of BERT). New SSL methods are also closing the performance gap to supervised models in the computer vision domain (Goyal et al. 2021). Based on the vision transformer (Dosovitskiy et al. 2021), BERT Pre-Training of Image Transformers (BEIT, Bao, Dong, and Wei 2021) shows promising results for training transformers using the SSL paradigm.

Active learning. The concept of active learning is another way of addressing the hurdles of expensive data annotation. Once an initial ML model is trained, active learning focuses on labeling examples with high uncertainty to improve the predictive performance. There are a variety of active learning approaches,

[30] `https://huggingface.co/docs/transformers/index`
[31] `https://github.com/matjesg/deepflash2/tree/transformers`

and some specifically address DL-based active learning for image data (Gal, Islam, and Ghahramani 2017). As mentioned in Chapter 4.3.2, the uncertainty estimates of *deepflash2* can be used in an active learning manner. However, the exact procedure has yet to be developed and evaluated carefully in further studies. As outlined by Karamcheti et al. (2021) in the context of a visual question answering task, active learning can possibly impair the predictive performance if out-of-distribution images are not excluded from the analysis and annotated for retraining. Another interesting finding of Karamcheti et al. (2021) should also be examined in the context of biomedical image segmentation: simply removing difficult examples from the training set can improve the overall performance.

5.2.2 Generative Adversarial Networks in the Creative Process

In Section 2.4, I establish a promising field at the confluence of design and AI and showcase how GANs and other DL technologies can assist fashion designers. However, the presented GAN prototypes generate rather low-quality images (see Figures 2.4.6 and 2.4.7). In recent years, researchers have been working on removing the characteristic artifacts and developing better DL architectures to achieve photorealistic results. A case in point is Nvidia's style-based generator for unconditional image modeling (StyleGAN, Karras, Laine, and Aila 2019), which has been continuously improved with StyleGAN2 (Karras et al. 2020) and StyleGAN3 (Karras et al. 2021) to redefine the state-of-the-art in terms of distribution quality metrics as well as perceived image quality.

Moreover, OpenAI's CLIP (Radford et al. 2021) opens up opportunities for a more "targeted" (conditional) image modeling by learning visual concepts from natural language supervision. VQGAN (Esser, Rombach, and Ommer 2021) in combination with CLIP creates high-resolution images from text prompts (see Figure 5.1). Further studies may explore how these technologies can be leveraged to tap into the best of human and computational creativity.

Figure 5.1: Image generated by VQGAN and CLIP[32] with input text *space collection* and *fashion show*.

5.3 Implications for Practice

To conclude this thesis, I would like to touch upon a few topics that are often neglected in scientific studies yet play an essential role in the day-to-day work of ML or DL projects.

The importance of data. The collection of representative and unbiased data is one of the most critical aspects of a DL project. Moreover, correctly understanding the data is essential to creating reliable DL models and often requires domain knowledge. Many prototypes presented in Chapter 2 did not proceed beyond the prototyping stage because of data issues. The DL system for the classification of structure-borne noise signals (Section 2.2), for instance, would have required data from more complex scenarios using different plugs and additional test persons. The prototype for the semantic segmentation of tissue in whole slide images (Section 2.5) would have required more clinical data, at least for validation. The style and matching engines of the fashion curation sys-

[32] `https://colab.research.google.com/drive/1ZAus_gn2RhTZWzOWUpPERNC0Q8OhZRTZ`. The video capture of the training iteration results is available at `https://vimeo.com/664750002`.

tem (Section 2.3) remained at a conceptual stage as no appropriate data were available. In contrast, the biomedical project for the segmentation of fluorescent neurons in microscopy images (Section 2.1) could continue successfully (Chapter 3) because data from many different laboratories were available, and the objectivity issues of data annotations could be resolved. However, most DL research focuses on algorithmic advances and neglects a systematic investigation of data-related aspects. The data-centric AI movement[33] presents a potential remedy as it breaks with the algorithm-centered paradigm and focuses on data instead of model architectures and training procedures.

Deep learning frameworks. When I started my first DL project – the segmentation of fluorescent neurons in microscopy images from Section 2.1 – I chose a rather new combination of Python libraries, *Keras* and *Tensorflow*, as my first toolset to create and train a U-Net from scratch. The models from Chapter 3 are also based on these frameworks, however, I had already started using *Tensorflow 2* during this phase. Hereafter, I switched to *PyTorch* and *fastai*. During the development of *deepflash2* (Chapter 4), I stopped building custom architectures and utilized libraries such as *timm* and *segmentation-models.pytorch* to fine-tune models from pretrained weights. Currently, it is also possible to fine-tune task-specific DL pipelines, e.g., using the *transformers* library of Hugging Face or the *detectron2* library of Meta Research. These advances are opening up DL to a much wider audience. It remains exciting to see how DL frameworks and development paradigms continue to evolve.

[33] https://landing.ai/data-centric-ai/

List of Figures

List of Tables

Bibliography

Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. "Tensorflow: A System for Large-Scale Machine Learning". In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265–283.

Abbasi, Ahmed, Suprateek Sarker, and Roger HL Chiang. 2016. "Big Data Research in Information Systems: Toward an Inclusive Research Agenda". *Journal of the association for information systems* 17 (2): 3.

Abdulla, Waleed. 2017. "Mask R-CNN for Object Detection and Instance Segmentation on Keras and TensorFlow". `https://github.com/matterport/Mask_RCNN`, *GitHub repository*.

Adomavicius, Gediminas, and Alexander Tuzhilin. 2005. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions". *IEEE Transactions on Knowledge & Data Engineering*, no. 6: 734–749.

Agarwal, Ritu, and Vasant Dhar. 2014. "Big Data, Data Science, and Analytics: The Opportunity and Challenge for IS Research". *Information Systems Research* 25 (3): 443–448.

Amed, Imran, Achim Berg, Sara Kappelmark, Saskia Hedrich, Johanna Andersson, Martine Drageset, and Robb Young. 2017. *The State of Fashion 2018*. Visited on 11/25/2018. `https://cdn.businessoffashion.com/reports/The_State_of_Fashion_2018_v2.pdf`.

Amershi, Saleema, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. 2019. "Software Engineering for Machine Learning: A Case Study". In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 291–300.

Andriluka, Mykhaylo, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2014. "2d Human Pose Estimation: New Benchmark and State of the Art Analysis". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3686–3693.

Baker, Monya. 2016. "Reproducibility Crisis?" *Nature* 533 (26): 353–66.

Balakrishnan, Tara, Michael Chui, Bryce Hall, and Nicolaus Henke. 2020. *Global Survey: The State of AI in 2020*. Visited on 11/25/2021. https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/McKinsey%20Analytics/Our%20Insights/Global%20survey%20The%20state%20of%20AI%20in%202020/Global-survey-The-state-of-AI-in-2020.pdf.

Bannon, Dylan, Erick Moen, Morgan Schwartz, Enrico Borba, Takamasa Kudo, Noah Greenwald, Vibha Vijayakumar, Brian Chang, Edward Pao, Erik Osterman, et al. 2021. "DeepCell Kiosk: Scaling Deep Learning–Enabled Cellular Image Analysis With Kubernetes". *Nature Methods* 18 (1): 43–45.

Bao, Hangbo, Li Dong, and Furu Wei. 2021. "BEiT: BERT Pre-Training of Image Transformers". *CoRR* abs/2106.08254. arXiv: 2106.08254.

Basili, Victor R. 1996. "The Role of Experimentation in Software Engineering: Past, Current, and Future". In *Proceedings of IEEE 18th International Conference on Software Engineering*, 442–449.

Baskerville, Richard, Abayomi Baiyere, Shirley Gregor, Alan Hevner, and Matti Rossi. 2018. "Design Science Research Contributions: Finding a Balance Between Artifact and Theory". *Journal of the Association for Information Systems* 19 (5): 358–376.

Bayramoglu, Neslihan, and Janne Heikkilä. 2016. "Transfer Learning for Cell Nuclei Classification in Histopathology Images". In *European Conference on Computer Vision*, 532–539.

Berg, Stuart, Dominik Kutra, Thorben Kroeger, Christoph N Straehle, Bernhard X Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, et al. 2019. "Ilastik: Interactive Machine Learning for (Bio) Image Analysis". *Nature Methods* 16 (12): 1226–1232.

Bergstra, James, and Yoshua Bengio. 2012. "Random Search for Hyper-Parameter Optimization." *Journal of machine learning research* 13 (2).

Berman, Maxim, Amal Rannen Triki, and Matthew B. Blaschko. 2018. “The Lovász-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-Over-Union Measure in Neural Networks”. In *Conference on Computer Vision and Pattern Recognition*, 4413–4421.

Beucher, Serge. 1979. “Use of Watersheds in Contour Detection”. In *Proceedings of the International Workshop on Image Processing*.

Bharadwaj, Anandhi, Omar A El Sawy, Paul A Pavlou, and N Venkatraman. 2013. “Digital Business Strategy: Toward a Next Generation of Insights”. *MIS quarterly*: 471–482.

Bono, Bernard de, Pierre Grenon, Richard Baldock, and Peter Hunter. 2013. “Functional Tissue Units and Their Primary Tissue Motifs in Multi-Scale Physiology”. *Journal of biomedical semantics* 4 (1): 1–13.

Bowman, Samuel R., Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Józefowicz, and Samy Bengio. 2016. “Generating Sentences from a Continuous Space”. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, 10–21.

Bracewell, Ronald Newbold, and Ronald N Bracewell. 1986. *The Fourier Transform and Its Applications*. Vol. 31999. McGraw-Hill New York.

Breck, Eric, Shanqing Cai, Eric Nielsen, Michael Salib, and D Sculley. 2017. “The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction”. In *2017 IEEE International Conference on Big Data (Big Data)*, 1123–1132.

Breiman, Leo. 2001. “Random Forests”. *Machine learning* 45 (1): 5–32.

Brock, Andrew, Jeff Donahue, and Karen Simonyan. 2019. “Large Scale GAN Training for High Fidelity Natural Image Synthesis”. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Burbidge, R., M. Trotter, B. Buxton, and S. Holden. 2001. “Drug Design by Machine Learning: Support Vector Machines for Pharmaceutical Data Analysis”. *Computers & Chemistry* 26 (1): 5–14.

Buslaev, Alexander, Vladimir I Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A Kalinin. 2020. “Albumentations: Fast and Flexible Image Augmentations”. *Information* 11 (2): 125.

Caicedo, Juan C, Allen Goodman, Kyle W Karhohs, Beth A Cimini, Jeanelle Ackerman, Marzieh Haghighi, CherKeng Heng, Tim Becker, Minh Doan, Claire McQuin, et al. 2019. "Nucleus Segmentation Across Imaging Experiments: The 2018 Data Science Bowl". *Nature methods* 16 (12): 1247–1253.

Campeau, S, WA Falls, WE Cullinan, DL Helmreich, M Davis, and SJ Watson. 1997. "Elicitation and Reduction of Fear: Behavioural and Neuroendocrine Indices and Brain Induction of the Immediate-Early Gene C-Fos". *Neuroscience* 78 (4): 1087–1104.

Cha, Hoon Sang, and Soeun You. 2018. "The Value and Risk of Curated Shopping: Online Consumer's Choice". In *Proceedings of the 51st Hawaii International Conference on System Sciences*.

Chamier, Lucas von, Romain F Laine, and Ricardo Henriques. 2019. "Artificial Intelligence for Microscopy: What You Should Know". *Biochemical Society Transactions* 47 (4): 1029–1040.

Chamier, Lucas von, Romain F Laine, Johanna Jukkala, Christoph Spahn, Daniel Krentzel, Elias Nehme, Martina Lerche, Sara Hernández-Pérez, Pieta K Mattila, Eleni Karinou, et al. 2021. "Democratising Deep Learning for Microscopy With ZeroCostDL4Mic". *Nature communications* 12 (1): 1–18.

Chan, Tony F, Gene Howard Golub, and Randall J LeVeque. 1982. "Updating Formulae and a Pairwise Algorithm for Computing Sample Variances". In *COMPSTAT 1982 5th Symposium Held at Toulouse 1982*, 30–41.

Chapman, Pete, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, Rudiger Wirth, et al. 2000. "CRISP-DM 1.0: Step-by-Step Data Mining Guide". *SPSS inc* 9:13.

Checco, Alessandro, Gianluca Demartini, Alexander Loeser, Ines Arous, Mourad Khayati, Matthias Dantone, Richard Koopmanschap, Svetlin Stalinov, Martin Kersten, and Ying Zhang. 2017. "FashionBrain Project: A Vision for Understanding Europe's Fashion Data Universe". In *KDD Fashion 2017*.

Chen, Hsinchun, Roger HL Chiang, and Veda C Storey. 2012. "Business Intelligence and Analytics: From Big Data to Big Impact". *MIS quarterly*: 1165–1188.

Chen, Huizhong, Andrew Gallagher, and Bernd Girod. 2012. “Describing Clothing by Semantic Attributes”. In *European Conference on Computer Vision*, 609–623.

Chen, Liang-Chieh, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. 2018a. “MaskLab: Instance Segmentation by Refining Object Detection With Semantic and Direction Features”. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 4013–4022.

Chen, Liang-Chieh, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018b. “Encoder-Decoder With Atrous Separable Convolution for Semantic Image Segmentation”. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 833–851.

Chen, Tianqi, and Carlos Guestrin. 2016. “Xgboost: A Scalable Tree Boosting System”. In *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 785–794.

Chen, Xiaoli, Sünje Dallmeier-Tiessen, Robin Dasler, Sebastian Feger, Pamfilos Fokianos, Jose Benito Gonzalez, Harri Hirvonsalo, Dinos Kousidis, Artemis Lavasa, Salvatore Mele, et al. 2019. “Open Is Not Enough”. *Nature Physics* 15 (2): 113–119.

Chodera, John, Alpha A. Lee, Nir London, and Frank von Delft. 2020. “Crowdsourcing Drug Discovery for Pandemics”. *Nature Chemistry* 12 (7): 581–581.

Chollet, François, et al. 2015. *Keras*. `https://keras.io`. `https://keras.io`.

Christensen, Kasper, Sladjana Nørskov, Lars Frederiksen, and Joachim Scholderer. 2017. “In Search of New Product Ideas: Identifying Ideas in Online Communities by Machine Learning and Text Mining”. *Creativity and Innovation Management* 26 (1): 17–30.

Christiansen, Eric M, Samuel J Yang, D Michael Ando, Ashkan Javaherian, Gaia Skibinski, Scott Lipnick, Elliot Mount, Alison O’Neil, Kevan Shah, Alicia K Lee, et al. 2018. “In Silico Labeling: Predicting Fluorescent Labels in Unlabeled Images”. *Cell* 173 (3): 792–803.

Cleveland, William S, and Robert McGill. 1985. “Graphical Perception and Graphical Methods for Analyzing Scientific Data”. *Science* 229 (4716): 828–833.

Collier, Dawn C., Stuart S.C. Burnett, Mayankkumar Amin, Stephen Bilton, Christopher Brooks, Amanda Ryan, Dominique Roniger, Danny Tran, and George Starkschall. 2003. "Assessment of Consistency in Contouring of Normal-Tissue Anatomic Structures." *Journal of applied clinical medical physics / American College of Medical Physics* 4 (1): 17–24.

Cortes, Corinna, and Vladimir Vapnik. 1995. "Support-Vector Networks". *Machine learning* 20 (3): 273–297.

Couger, J Daniel, Lexis F Higgins, and Scott C McIntyre. 1993. "(Un)structured Creativity in Information Systems Organizations". *Mis Quarterly*: 375–397.

Cropley, Arthur. 2006. "In Praise of Convergent Thinking". *Creativity research journal* 18 (3): 391–404.

David, H. 2015. "Why Are There Still So Many Jobs? The History and Future of Workplace Automation". *Journal of economic perspectives* 29 (3): 3–30.

David, H, and David Dorn. 2013. "The Growth of Low-Skill Service Jobs and the Polarization of the US Labor Market". *American Economic Review* 103 (5): 1553–97.

Davis, Jason V, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. 2007. "Information-Theoretic Metric Learning". In *Proceedings of the 24th International Conference on Machine Learning*, 209–216.

DeGrave, Alex J, Joseph D Janizek, and Su-In Lee. 2021. "AI for Radiographic COVID-19 Detection Selects Shortcuts Over Signal". *Nature Machine Intelligence*: 1–10.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. "ImageNet: A Large-Scale Hierarchical Image Database". In *Conference on Computer Vision and Pattern Recognition*, 248–255.

Der Kiureghian, Armen, and Ove Ditlevsen. 2009. "Aleatory or Epistemic? Does It Matter?" *Structural safety* 31 (2): 105–112.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186.

Dietterich, Thomas G. 2000. “Ensemble Methods in Machine Learning”. In *International Workshop on Multiple Classifier Systems*, 1–15.

Doherty, Neil F, and Fiona Ellis-Chadwick. 2010. “Internet Retailing: The Past, the Present and the Future”. *International Journal of Retail & Distribution Management* 38 (11/12): 943–965.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Driscoll, Meghan K, Erik S Welf, Andrew R Jamieson, Kevin M Dean, Tadamoto Isogai, Reto Fiolka, and Gaudenz Danuser. 2019. “Robust and Automated Detection of Subcellular Morphological Motifs in 3D Microscopy Images”. *Nature methods*: 1–8.

Drozdzal, Michal, Eugene Vorontsov, Gabriel Chartrand, Samuel Kadoury, and Chris Pal. 2016. “The Importance of Skip Connections in Biomedical Image Segmentation”. In *International Workshop on Deep Learning in Medical Image Analysis*, 179–187.

Du Sautoy, Marcus. 2019. *The Creativity Code: How AI Is Learning to Write, Paint and Think*. HarperCollins UK.

Dürr, Alexander, Matthias Griebel, Giacomo Welsch, and Frédéric Thiesse. 2020. “Predicting Fraudulent Initial Coin Offerings Using Information Extracted From Whitepapers”. In *28th European Conference on Information Systems - Liberty, Equality, and Fraternity in a Digitizing World, ECIS 2020, Marrakech, Morocco, June 15-17, 2020*.

Dvorak, Paul. 1998. “Poka-Yoke Designs Make Assemblies Mistake-Proof”. *Machine design* 70 (4): 181–4.

Eck, Douglas, and Juergen Schmidhuber. 2002. “A First Look at Music Composition Using LSTM Recurrent Neural Networks”. *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale* 103.

Esser, Patrick, Robin Rombach, and Björn Ommer. 2021. “Taming Transformers for High-Resolution Image Synthesis”. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, Virtual, June 19-25, 2021*, 12873–12883.

Falk, Thorsten, Dominic Mai, Robert Bensch, Özgün Çiçek, Ahmed Abdulkadir, Yassine Marrakchi, Anton Böhm, Jan Deubner, Zoe Jäckel, Katharina Seiwald, et al. 2019. “U-Net: Deep Learning for Cell Counting, Detection, and Morphometry”. *Nature Methods* 16 (1): 67–70.

Fanelli, Daniele. 2018. “Opinion: Is Science Really Facing a Reproducibility Crisis, and Do We Need It To?” *Proceedings of the National Academy of Sciences* 115 (11): 2628–2631.

Feldman-Stewart, Deb, Nancy Kocovski, Beth A McConnell, Michael D Brundage, and William J Mackillop. 2000. “Perception of Quantitative Information for Treatment Decisions”. *Medical Decision Making* 20 (2): 228–238.

Feng, Guoping, Rebecca H Mellor, Michael Bernstein, Cynthia Keller-Peck, Quyen T Nguyen, Mia Wallace, Jeanne M Nerbonne, Jeff W Lichtman, and Joshua R Sanes. 2000. “Imaging Neuronal Subsets in Transgenic Mice Expressing Multiple Spectral Variants of GFP”. *Neuron* 28 (1): 41–51.

Fleiss, Joseph L, and Jacob Cohen. 1973. “The Equivalence of Weighted Kappa and the Intraclass Correlation Coefficient as Measures of Reliability”. *Educational and psychological measurement* 33 (3): 613–619.

Frambach, Janneke M, Cees PM van der Vleuten, and Steven J Durning. 2013. “AM Last Page: Quality Criteria in Qualitative and Quantitative Research”. *Academic Medicine* 88 (4): 552.

Frome, Andrea, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc’Aurelio Ranzato, and Tomás Mikolov. 2013. “DeViSE: A Deep Visual-Semantic Embedding Model”. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a Meeting Held December 5-8, 2013, Lake Tahoe, Nevada, United States*, 2121–2129.

Al-Fuqaha, Ala, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. 2015. “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications”. *IEEE communications surveys & tutorials* 17 (4): 2347–2376.

Gabale, Vijay, and Anand Prabhu Subramanian. 2018. “How To Extract Fashion Trends From Social Media? A Robust Object Detector With Support For Unsupervised Learning”. *CoRR* abs/1806.10787. arXiv: 1806.10787.

Gal, Yarin, and Zoubin Ghahramani. 2016. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning”. In *Proceedings of the 33nd International Conference on Machine Learning, ICML*, 1050–1059.

Gal, Yarin, Riashat Islam, and Zoubin Ghahramani. 2017. “Deep Bayesian Active Learning With Image Data”. In *Proceedings of the 34th International Conference on Machine Learning*, 70:1183–1192. Proceedings of Machine Learning Research.

Gallo, Francisco T, Cynthia Katche, Juan F Morici, Jorge H Medina, and Noelia V Weisstaub. 2018. “Immediate Early Genes, Memory and Psychiatric Disorders: Focus on C-Fos, Egr1 and Arc”. *Frontiers in behavioral neuroscience* 12:79.

Gatys, Leon A, Alexander S Ecker, and Matthias Bethge. 2016. “Image Style Transfer Using Convolutional Neural Networks”. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2414–2423.

Gibbs, Samuel. 2016. “Mercedes-Benz Swaps Robots for People on Its Assembly Lines”. *The Guardian* 26.

Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580–587.

Golovin, Daniel, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, and David Sculley. 2017. “Google Vizier: A Service for Black-Box Optimization”. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1487–1495.

Goodfellow, Ian J., Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. "Generative Adversarial Nets". In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2672–2680.

Goyal, Priya, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, and Piotr Bojanowski. 2021. "Self-supervised Pretraining of Visual Features in the Wild". *CoRR* abs/2103.01988.

Gregor, Shirley, and Alan R Hevner. 2013. "Positioning and Presenting Design Science Research for Maximum Impact". *MIS quarterly*: 337–355.

Gregor, Shirley, David Jones, et al. 2007. "The Anatomy of a Design Theory".

Griebel, Matthias, Alexander Dürr, and Nikolai Stein. 2019. "Applied Image Recognition: Guidelines for Using Deep Learning Models in Practice". In *Human Practice. Digital Ecologies. Our Future. 14. Internationale Tagung Wirtschaftsinformatik (WI 2019), February 24-27, 2019, Siegen, Germany*, 393–406.

Griebel, Matthias, Christoph Flath, and Sascha Friesike. 2020. "Augmented Creativity: Leveraging Artificial Intelligence for Idea Generation in the Creative Sphere". In *28th European Conference on Information Systems - Liberty, Equality, and Fraternity in a Digitizing World, ECIS 2020, Marrakech, Morocco, June 15-17, 2020*.

Griebel, Matthias, Dennis Segebarth, Nikolai Stein, Nina Schukraft, Philip Tovote, Robert Blum, and Christoph M. Flath. 2021. "Deep-Learning in the Bioimaging Wild: Handling Ambiguous Data With Deepflash2". *CoRR* abs/2111.06693, currently under review at Nature Methods. arXiv: 2111.06693.

Griebel, Matthias, Giacomo Welsch, Toni Greif, and Christoph Flath. 2019. "A Picture Is Worth More Than a Thousand Purchases: Designing an Image-Based Fashion Curation System". In *27th European Conference on Information Systems - Information Systems for a Sharing Society, ECIS 2019, Stockholm and Uppsala, Sweden, June 8-14, 2019*.

Guan, Melody Y, Varun Gulshan, Andrew M Dai, and Geoffrey E Hinton. 2018. "Who Said What: Modeling Individual Labelers Improves Classification". In *Thirty-Second AAAI Conference on Artificial Intelligence*, 3109–3118.

Gubbi, Jayavardhana, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. 2013. "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions". *Future generation computer systems* 29 (7): 1645–1660.

Guo, Yanming, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. 2016. "Deep Learning for Visual Understanding: A Review". *Neurocomputing* 187:27–48.

Guzowski, John F, Barry Setlow, Edward K Wagner, and James L McGaugh. 2001. "Experience-Dependent Gene Expression in the Rat Hippocampus After Spatial Learning: A Comparison of the Immediate-Early genesArc, C-Fos, and Zif268". *Journal of Neuroscience* 21 (14): 5089–5098.

Haberl, Matthias G., Christopher Churas, Lucas Tindall, Daniela Boassa, Sébastien Phan, Eric A. Bushong, Matthew Madany, Raffi Akay, Thomas J. Deerinck, Steven T. Peltier, and Mark H. Ellisman. 2018. "CDeep3M—Plug-and-Play Cloud-Based Deep Learning for Image Segmentation". *Nature Methods* 15 (9): 677–680.

Hadi Kiapour, M, Xufeng Han, Svetlana Lazebnik, Alexander C Berg, and Tamara L Berg. 2015. "Where to Buy It: Matching Street Clothing Photos in Online Shops". In *Proceedings of the IEEE International Conference on Computer Vision*, 3343–3351.

Al-Halah, Ziad, Rainer Stiefelhagen, and Kristen Grauman. 2017. "Fashion Forward: Forecasting Visual Style in Fashion". In *Proceedings of the IEEE International Conference on Computer Vision*, 388–397.

Han, Xintong, Zuxuan Wu, Phoenix X. Huang, Xiao Zhang, Menglong Zhu, Yuan Li, Yang Zhao, and Larry S. Davis. 2017a. "Automatic Spatially-Aware Fashion Concept Discovery". *Proceedings of the IEEE International Conference on Computer Vision* 2017-October:1472–1480.

Han, Xintong, Zuxuan Wu, Yu-Gang Jiang, and Larry S. Davis. 2017b. “Learning Fashion Compatibility with Bidirectional LSTMs”. In *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*, 1078–1086.

Haralick, Robert M, Karthikeyan Shanmugam, and Its’ Hak Dinstein. 1973. “Textural Features for Image Classification”. *IEEE Transactions on systems, man, and cybernetics*, no. 6: 610–621.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017a. “Mask R-CNN”. In *Proceedings of the IEEE International Conference on Computer Vision*, 2980–2988.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. “Deep Residual Learning for Image Recognition”. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

— . 2015. “Delving Deep Into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In *International Conference on Computer Vision*, 1026–1034.

He, Ruidan, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017b. “An Unsupervised Neural Attention Model for Aspect Extraction”. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*: 388–397.

Head, Megan L, Luke Holman, Rob Lanfear, Andrew T Kahn, and Michael D Jennions. 2015. “The Extent and Consequences of P-Hacking in Science”. *PLoS biology* 13 (3).

Heinemann, Gerrit. 2017. *Der Neue Online-Handel: Geschäftsmodelle, Geschäftssysteme Und Benchmarks Im E-Commerce*. Springer-Verlag.

Hermann, Jeremy, and Mike Del Balso. 2018. *Scaling Machine Learning at Uber With Michelangelo*. Visited on 01/06/2022. `https://eng.uber.com/scaling-michelangelo/`.

Hevner, Alan R., Salvatore T. March, Jinsoo Park, and Sudha Ram. 2004. “Design Science in Information Systems Research”. *MIS Quarterly* 28 (1): 75–105.

Howard, Jeremy, and Sylvain Gugger. 2020. “Fastai: A Layered API for Deep Learning”. *Information* 11 (2): 108.

Hsiao, Wei-Lin, and Kristen Grauman. 2017. “Learning the Latent “Look”: Unsupervised Discovery of a Style-Coherent Embedding From Fashion Images”. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 4213–4222.

Hu, Hua, Jian Gan, and Peter Jonas. 2014. “Fast-Spiking, Parvalbumin+ GABAergic Interneurons: From Cellular Design to Microcircuit Function”. *Science* 345 (6196).

Huang, Junshi, Rogerio S Feris, Qiang Chen, and Shuicheng Yan. 2015. “Cross-Domain Image Retrieval With a Dual Attribute-Aware Ranking Network”. In *Proceedings of the IEEE International Conference on Computer Vision*, 1062–1070.

Huang, Xun, and Serge J. Belongie. 2017. “Arbitrary Style Transfer in Real-Time With Adaptive Instance Normalization.” In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 1510–1519.

Huff, Nicole C, Matthew Frank, Karli Wright-Hardesty, David Sprunger, Patricia Matus-Amat, Emily Higgins, and Jerry W Rudy. 2006. “Amygdala Regulation of Immediate-Early Gene Expression in the Hippocampus Induced by Contextual Fear Conditioning”. *Journal of Neuroscience* 26 (5): 1616–1623.

Hutson, Matthew. 2018. “Artificial Intelligence Faces Reproducibility Crisis”. *Science* 359 (6377): 725–726.

Hwang, Jyh-Jing, Sergei Azernikov, Alexei A. Efros, and Stella X. Yu. 2018. “Learning Beyond Human Expertise with Generative Models for Dental Restorations”. *CoRR* abs/1804.00064. arXiv: 1804.00064.

Ioannidis, John PA. 2016. “Why Most Clinical Research Is Not Useful”. *PLoS medicine* 13 (6): e1002049.

Ioffe, Sergey, and Christian Szegedy. 2015. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 37:448–456. JMLR Workshop and Conference Proceedings.

Isensee, Fabian, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. 2021. “nnU-Net: A Self-Configuring Method for Deep Learning-Based Biomedical Image Segmentation”. *Nature Methods* 18 (2): 203–211.

Ji, Wei, Shuang Yu, Junde Wu, Kai Ma, Cheng Bian, Qi Bi, Jingjing Li, Hanruo Liu, Li Cheng, and Yefeng Zheng. 2021. "Learning Calibrated Medical Image Segmentation via Multi-Rater Agreement Modeling". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12341–12351.

Karamcheti, Siddharth, Ranjay Krishna, Li Fei-Fei, and Christopher D. Manning. 2021. "Mind Your Outliers! Investigating the Negative Impact of Outliers on Active Learning for Visual Question Answering": 7265–7281.

Karras, Tero, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. "Alias-Free Generative Adversarial Networks". *CoRR* abs/2106.12423. arXiv: 2106.12423.

Karras, Tero, Samuli Laine, and Timo Aila. 2019. "A Style-Based Generator Architecture for Generative Adversarial Networks". In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 4401–4410.

Karras, Tero, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. "Analyzing and Improving the Image Quality of StyleGAN". In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 8107–8116.

Kato, Natsumi, Osone Hiroyuki, Daitetsu Sato, and Naoya Muramatsu. 2017. "Crowd Sourcing Clothes Design Directed by Adversarial Neural Networks". In *NIPS 2017 Workshop: Machine Learning for Creativity and Design*, 1–4. Nips.

Keiser, Ashley A, Lacie M Turnbull, Mara A Darian, Dana E Feldman, Iris Song, and Natalie C Tronson. 2017. "Sex Differences in Context Fear Generalization and Recruitment of Hippocampus and Amygdala During Retrieval". *Neuropsychopharmacology* 42:397–407.

Kiapour, M Hadi, Kota Yamaguchi, Alexander C Berg, and Tamara L Berg. 2014. "Hipster Wars: Discovering Elements of Fashion Styles". In *European Conference on Computer Vision*, 472–488.

Kim, Hongki, Kil-Soo Suh, and Un-Kon Lee. 2013. "Effects of Collaborative Online Shopping on Shopping Experience Through Social and Relational Perspectives". *Information & Management* 50 (4): 169–180.

Kingma, Diederik P., and Jimmy Ba. 2015. “Adam: A Method for Stochastic Optimization”. In *3rd International Conference on Learning Representations*.

Kluyver, Thomas, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing, and Jupyter development team. 2016. “Jupyter Notebooks-a Publishing Format for Reproducible Computational Workflows”. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 87–90.

Kohavi, Ron, et al. 1995. “A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection”. In *Ijcai*, 14:1137–1145. 2.

Kohl, Simon, Bernardino Romera-Paredes, Clemens Meyer, Jeffrey De Fauw, Joseph R. Ledsam, Klaus H. Maier-Hein, S. M. Ali Eslami, Danilo Jimenez Rezende, and Olaf Ronneberger. 2018. “A Probabilistic U-Net for Segmentation of Ambiguous Images”. In *Advances in Neural Information Processing Systems 31*, 6965–6975.

Kong, Qiuqiang, Turab Iqbal, Yong Xu, Wenwu Wang, and Mark D. Plumbley. 2018. “DCASE 2018 Challenge Surrey Cross-Task Convolutional Neural Network Baseline”. In *Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events, DCASE 2018, Surrey, UK, November 19-20, 2018*, 217–221.

Krenzer, Adrian, Nikolai Stein, Matthias Griebel, and Christoph Flath. 2019. “Augmented Intelligence for Quality Control of Manual Assembly Processes Using Industrial Wearable Systems”. In *Proceedings of the 40th International Conference on Information Systems, ICIS 2019, Munich, Germany, December 15-18, 2019*.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. “Imagenet Classification With Deep Convolutional Neural Networks”, *Advances in neural information processing systems* 25:1097–1105.

Kuhn, Harold W. 1955. “The Hungarian Method for the Assignment Problem”. *Naval research logistics quarterly* 2 (1-2): 83–97.

Kwon, Yongchan, Joong-Ho Won, Beom Joon Kim, and Myunghee Cho Paik. 2020. “Uncertainty Quantification Using Bayesian Neural Networks in Classification: Application to Biomedical Image Segmentation”. *Computational Statistics & Data Analysis* 142:106816.

Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. 2017. “Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles”. In *Advances in Neural Information Processing Systems 30*, 6402–6413.

Lampert, Thomas A, André Stumpf, and Pierre Gançarski. 2016. “An Empirical Study Into Annotator Agreement, Ground Truth Estimation, and Algorithm Evaluation”. *IEEE Transactions on Image Processing* 25 (6): 2557–2572.

Landis, J Richard, and Gary G Koch. 1977. “The Measurement of Observer Agreement for Categorical Data”. *biometrics*: 159–174.

LeCun, Yann A, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. 2012. “Efficient Backprop”. In *Neural Networks: Tricks of the Trade*, 9–48.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. “Deep Learning”. *Nature* 521 (7553): 436–444.

Lederman, Reeva, Greg Wadley, John Gleeson, Sarah Bendall, and Mario Álvarez-Jiménez. 2014. “Moderated Online Social Therapy: Designing and Evaluating Technology for Mental Health”. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21 (1): 1–26.

Li, Anan, H. Gong, B. Zhang, Q. Wang, C. Yan, J. Wu, L. Qian, S. Zeng, and Q. Luo. 2010. “Micro-Optical Sectioning Tomography to Obtain a HIgh-Resolution Atlas of the Mouse Brain”. *Science* 1404 (2010): 1404–1408.

Lin, T., P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. 2017a. “Feature Pyramid Networks for Object Detection”. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 936–944.

Lin, Tsung-Yi, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017b. “Focal Loss for Dense Object Detection”. In *IEEE International Conference on Computer Vision*, 2999–3007.

Lin, Tsung-Yi, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014a. “Microsoft COCO: Common Objects in Context”. In *European Conference on Computer Vision*, 740–755.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014b. "Microsoft COCO: Common Objects in Context". In *Computer Vision – ECCV 2014*, 740–755.

Lin, Zhijie. 2014. "An Empirical Investigation of User and System Recommendations in E-Commerce". *Decision Support Systems* 68:111–124.

Lindberg, Tilmann, Christoph Meinel, and Ralf Wagner. 2011. "Design Thinking: A Fruitful Concept for IT Development?" In *Design Thinking*, 3–18.

Liu, Shu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. 2018. "Path Aggregation Network for Instance Segmentation". In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 8759–8768.

Liu, Weitang, Xiaoyun Wang, John D. Owens, and Yixuan Li. 2020. "Energy-Based Out-of-Distribution Detection". In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual*.

Liu, Ziwei, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. 2016a. "DeepFashion: Powering Robust Clothes Recognition and Retrieval With Rich Annotations". In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1096–1104.

Liu, Ziwei, Sijie Yan, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2016b. "Fashion Landmark Detection in the Wild". In *European Conference on Computer Vision*, 229–245.

Loebbecke, Claudia, and Arnold Picot. 2015. "Reflections on Societal and Business Model Transformation Arising From Digitization and Big Data Analytics: A Research Agenda". *The Journal of Strategic Information Systems* 24 (3): 149–157.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell. 2015. "Fully Convolutional Networks for Semantic Segmentation". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440.

Loshchilov, Ilya, and Frank Hutter. 2019. "Decoupled Weight Decay Regularization". In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Lowekamp, Bradley Christopher, David T Chen, Luis Ibáñez, and Daniel Blezek. 2013. "The Design of SimpleITK". *Frontiers in neuroinformatics* 7:45.

Lucas, Alice M, Pearl V Ryder, Bin Li, Beth A Cimini, Kevin W Eliceiri, and Anne E Carpenter. 2021. "Open-Source Deep-Learning Software for Bioimage Segmentation". *Molecular Biology of the Cell* 32 (9): 823–829.

Maška, Martin, Vladimir Ulman, David Svoboda, Pavel Matula, Petr Matula, Cristina Ederra, Ainhoa Urbiola, Tomás España, Subramanian Venkatesan, Deepak MW Balak, et al. 2014. "A Benchmark for Comparison of Cell Tracking Algorithms". *Bioinformatics* 30 (11): 1609–1617.

Mathwick, Charla, and Edward Rigdon. 2004. "Play, Flow, and the Online Search Experience". *Journal of consumer research* 31 (2): 324–332.

Matzen, Kevin, Kavita Bala, and Noah Snavely. 2017. "StreetStyle: Exploring World-Wide Clothing Styles From Millions of Photos". *CoRR* abs/1706.01869. arXiv: 1706.01869.

McAuley, Julian J., Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. "Image-Based Recommendations on Styles and Substitutes". In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, ed. by Ricardo Baeza-Yates, Mounia Lalmas, Alistair Moffat, and Berthier A. Ribeiro-Neto, 43–52.

McDole, Katie, Léo Guignard, Fernando Amat, Andrew Berger, Grégoire Malandain, Loïc A. Royer, Srinivas C. Turaga, Kristin Branson, and Philipp J. Keller. 2018. "In Toto Imaging and Reconstruction of Post-Implantation Mouse Development at the Single-Cell Level". *Cell*: 1–18.

McKelvey, Kathryn, and Janine Munslow. 2011. *Fashion Design: Process, Innovation and Practice*. John Wiley & Sons.

McQuin, Claire, Allen Goodman, Vasiliy Chernyshev, Lee Kamentsky, Beth A. Cimini, Kyle W. Karhohs, Minh Doan, Liya Ding, Susanne M. Rafelski, Derek Thirstrup, Winfried Wiegraebe, Shantanu Singh, Tim Becker, Juan C. Caicedo, and Anne E. Carpenter. 2018. "CellProfiler 3.0: Next-Generation Image Processing for Biology". *PLOS Biology* 16 (7): 1–17.

Meijering, Erik. 2020. "A Bird's-Eye View of Deep Learning in Bioimage Analysis". *Computational and Structural Biotechnology Journal* 18:2312–2325.

Meijering, Erik, Anne E. Carpenter, Hanchuan Peng, Fred A. Hamprecht, and Jean-Christophe Olivo-Marin. 2016. “Imagining the Future of Bioimage Analysis”. *Nature Biotechnology* 34 (12): 1250–1255.

Miller, Arthur I. 2019. *The Artist in the Machine: The World of AI-powered Creativity*. Mit Press.

Milletari, Fausto, Nassir Navab, and Seyed-Ahmad Ahmadi. 2016. “V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation”. In *International Conference on 3D Vision (3DV)*, 565–571.

Mirza, Mehdi, and Simon Osindero. 2014. “Conditional Generative Adversarial Nets”. *CoRR* abs/1411.1784. arXiv: 1411.1784.

Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar. 2012. *Foundations of Machine Learning*. MIT press.

Morana, Stefan, Silvia Schacht, Ansgar Scherp, and Alexander Maedche. 2017. “A Review of the Nature and Effects of Guidance Design Features”. *Decision Support Systems* 97:31–42.

Müller-Wienbergen, Felix, Oliver Müller, Stefan Seidel, and Jörg Becker. 2011. “Leaving the Beaten Tracks in Creative Work-a Design Theory for Systems That Support Convergent and Divergent Thinking”. *Journal of the Association for Information Systems* 12 (11): 714.

Müller, Oliver, Iris Junglas, Jan Vom Brocke, and Stefan Debortoli. 2016. “Utilizing Big Data Analytics for Information Systems Research: Challenges, Promises and Guidelines”. *European Journal of Information Systems* 25 (4): 289–302.

Muller, Sune Dueholm, and Frank Ulrich. 2013. “Creativity and Information Systems in a Hypercompetitive Environment: A Literature Review”. *Communications of the Association for Information Systems* 32 (1): 7.

Murawski, Nathen J, Anna Y Klintsova, and Mark E Stanton. 2012. “Neonatal Alcohol Exposure and the Hippocampus in Developing Male Rats: Effects on Behaviorally Induced CA1 C-Fos Expression, CA1 Pyramidal Cell Number, and Contextual Fear Conditioning”. *Neuroscience* 206:89–99.

Nair, Vinod, and Geoffrey E. Hinton. 2010. “Rectified Linear Units Improve Restricted Boltzmann Machines”. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, 807–814.

Nakamura, Takuma, and Ryosuke Goto. 2018. “Outfit Generation and Style Extraction via Bidirectional LSTM and Autoencoder”. *CoRR* abs/1807.03133. arXiv: 1807.03133.

Nguyen, Dat Quoc, Thanh Vu, Afshin Rahimi, Mai Hoang Dao, Linh The Nguyen, and Long Doan. 2020. “WNUT-2020 Task 2: Identification of Informative COVID-19 English Tweets”. In *Proceedings of the Sixth Workshop on Noisy User-Generated Text, W-Nut at EMNLP 2020 Online, November 19, 2020*, 314–318.

Niedworok, Christian J., Alexander P.Y. Brown, M. Jorge Cardoso, Pavel Osten, Sebastien Ourselin, Marc Modat, and Troy W. Margrie. 2016. “AMAP Is a Validated Pipeline for Registration and Segmentation of High-Resolution Mouse Brain Data”. *Nature Communications* 7 (May): 1–9.

Noh, Hyeonwoo, Seunghoon Hong, and Bohyung Han. 2015. “Learning Deconvolution Network for Semantic Segmentation”. In *Proceedings of the IEEE International Conference on Computer Vision*, 1520–1528.

Oberdorf, Felix, Nikolai Stein, Nicolas Walk, Matthias Griebel, and Christoph Flath. 2020. “ADR for Big-Data IT Artifact Development: An Escalation Management Example”. In *Proceedings of the 41st International Conference on Information Systems, ICIS 2020, Making Digital Inclusive: Blending the Locak and the Global, Hyderabad, India, December 13-16, 2020*.

Oquab, Maxime, Leon Bottou, Ivan Laptev, and Josef Sivic. 2014. “Learning and Transferring Mid-Level Image Representations Using Convolutional Neural Networks”. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1717–1724.

Osborn, Alex F. 1953. “Applied Imagination.”

Osten, Pavel, and Troy W. Margrie. 2013. “Mapping Brain Circuitry With a Light Microscope”. *Nature Methods* 10 (6): 515–523.

Otsu, Nobuyuki. 1979. “A Threshold Selection Method From Gray-Level Histograms”. *IEEE transactions on systems, man, and cybernetics* 9 (1): 62–66.

Ounkomol, Chawin, Sharmishtaa Seshamani, Mary M Maleckar, Forrest Collman, and Gregory R Johnson. 2018. “Label-Free Prediction of Three-Dimensional Fluorescence Images From Transmitted-Light Microscopy”. *Nature methods* 15 (11): 917–920.

Padmanabhan, Balaji, Xiao Fang, and Nachiketa Sahoo. 2022. “Machine Learning in Information Systems Research”. *MIS quarterly* forthcoming.

Pan, Yunhe. 2016. “Heading Toward Artificial Intelligence 2.0”. *Engineering* 2 (4): 409–413.

Parker, Karen, James A. Stone, Ross Arena, Debra Lundberg, Sandeep Aggarwal, David Goodhart, and Mouhieddin Traboulsi. 2018. *Notes From the AI Frontier: Insights From Use Cases*. `https://www.mckinsey.com/featured-insights/artificial-intelligence/notes-from-the-ai-frontier-applications-and-value-of-deep-learning`.

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In *Advances in Neural Information Processing Systems 32*, 8024–8035.

Patel, Pankesh, Muhammad Intizar Ali, and Amit Sheth. 2017. “On Using the Intelligent Edge for IoT Analytics”. *IEEE Intelligent Systems* 32 (5): 64–69.

Patwa, Parth, Mohit Bhardwaj, Vineeth Guptha, Gitanjali Kumari, Shivam Sharma, Srinivas PYKL, Amitava Das, Asif Ekbal, Md. Shad Akhtar, and Tanmoy Chakraborty. 2021. “Overview of CONSTRAINT 2021 Shared Tasks: Detecting English COVID-19 Fake News and Hindi Hostile Posts”. In *Combating Online Hostile Posts in Regional Languages during Emergency Situation - First International Workshop, CONSTRAINT 2021, Collocated with AAAI 2021, Virtual Event, February 8, 2021, Revised Selected Papers*, 1402:42–53.

Pavlou, Paul A. 2018. “Internet of Things – Will Humans Be Replaced or Augmented?” *Marketing Intelligence Review* 10 (2): 42–47.

Peffers, Ken, Tuure Tuunanen, Marcus A. Rothenberger, and Samir Chatterjee. 2007. “A Design Science Research Methodology for Information Systems Research”. *Journal of Management Information Systems* 24 (3): 45–77.

Perkel, Jeffrey M. 2018. “Why Jupyter Is Data Scientists’ Computational Notebook of Choice”. *Nature* 563 (7732): 145–147.

Pfeiffer, Sabine. 2016. “Robots, Industry 4.0 and Humans, or Why Assembly Work Is More Than Routine Work”. *Societies* 6 (2): 16.

Powers, David Martin. 2011. “Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation”. *Journal of Machine Learning Technologies* 2 (1): 37–63.

Provost, Foster, and Tom Fawcett. 2013. “Data Science and Its Relationship to Big Data and Data-Driven Decision Making”. *Big data* 1 (1): 51–59.

Radford, Alec, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. “Learning Transferable Visual Models From Natural Language Supervision”. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, 139:8748–8763.

Radford, Alec, Luke Metz, and Soumith Chintala. 2016. “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Rai, Arun. 2017. “Editor’s Comments: Diversity of Design Science Research”. *MIS quarterly* 41 (1): iii–xviii.

Rajnai, Zoltán, and István Kocsis. 2017. “Labor Market Risks of Industry 4.0, Digitization, Robots and AI”. In *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, 343–346.

Ramamoorthi, Kartik, Robin Fropf, Gabriel M Belfort, Helen L Fitzmaurice, Ross M McKinney, Rachael L Neve, Tim Otto, and Yingxi Lin. 2011. “Npas4 Regulates a Transcriptional Program in CA3 Required for Contextual Memory Formation”. *Science* 334 (6063): 1669–1675.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You? Explaining the Predictions of Any Classifier". In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144.

Roberts, Adam, Jesse Engel, and Douglas Eck. 2017. "Hierarchical Variational Autoencoders for Music". In *NIPS Workshop on Machine Learning for Creativity and Design*.

Roberts, Michael, Derek Driggs, Matthew Thorpe, Julian Gilbey, Michael Yeung, Stephan Ursprung, Angelica I Aviles-Rivero, Christian Etmann, Cathal McCague, Lucian Beer, et al. 2021. "Common Pitfalls and Recommendations for Using Machine Learning to Detect and Prognosticate for COVID-19 Using Chest Radiographs and CT Scans". *Nature Machine Intelligence* 3 (3): 199–217.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 2015. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234–241.

Rosenthal, Robert, and M Robin DiMatteo. 2002. "Meta-Analysis". In *Stevens extquotesingle Handbook of Experimental Psychology*, 391–428.

Rost, Benjamin R, Franziska Schneider-Warme, Dietmar Schmitz, and Peter Hegemann. 2017. "Optogenetic Tools for Subcellular Applications in Neuroscience". *Neuron* 96 (3): 572–603.

Ruediger, Sarah, Claudia Vittori, Ewa Bednarek, Christel Genoud, Piergiorgio Strata, Benedetto Sacchetti, and Pico Caroni. 2011. "Learning-Related Feedforward Inhibitory Connectivity Growth Required for Memory Precision". *Nature* 473 (7348): 514–518.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. "Imagenet Large Scale Visual Recognition Challenge". *International journal of computer vision* 115 (3): 211–252.

Salehi, Seyed Sadegh Mohseni, Deniz Erdogmus, and Ali Gholipour. 2017. "Tversky Loss Function for Image Segmentation Using 3D Fully Convolutional Deep Networks". In *International Workshop on Machine Learning in Medical Imaging*, 379–387.

Schmidt, Uwe, Martin Weigert, Coleman Broaddus, and Gene Myers. 2018. "Cell Detection With Star-Convex Polygons". In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 265–273.

Schmitz, Christoph, Hubert Korr, and Helmut Heinsen. 1999. "Design-Based Counting Techniques: The Real Problems". *Trends in neurosciences* 22 (8): 345.

Sebald, Anna Kathrin, and Frank Jacob. 2018. "Help Welcome or Not: Understanding Consumer Shopping Motivation in Curated Fashion Retailing". *Journal of Retailing and Consumer Services* 40:188–203.

Segebarth, Dennis, Matthias Griebel, Alexander Dürr, Cora R. von Collenberg, Corinna Martin, Dominik Fiedler, Lucas B Comeras, Anupam Sah, Nikolai Stein, Rohini Gupta, et al. 2018. "DeepFLaSh, a Deep Learning Pipeline for Segmentation of Fluorescent Labels in Microscopy Images". *bioRxiv*.

Segebarth, Dennis, Matthias Griebel, Nikolai Stein, Cora R. von Collenberg, Corinna Martin, Dominik Fiedler, Lucas B Comeras, Anupam Sah, Victoria Schoeffler, Teresa Lüffe, et al. 2020a. "On the Objectivity, Reliability, and Validity of Deep Learning Enabled Bioimage Analyses". *eLife* 9:e59780.

Segebarth, Dennis, Matthias Griebel, Nikolai Stein, Cora R. von Collenberg, Corinna Martin, Dominik Fiedler, Lucas B. Comeras, Anupam Sah, Victoria Schoeffler, Theresa Lüffe, Alexander Dürr, Rohini Gupta, Manju Sasi, Christine Lillesaar, Maren D. Lange, Ramon O. Tasan, Nicolas Singewald, Hans-Christian Pape, Christoph M. Flath, and Robert Blum. 2020b. *Data From: On the Objectivity, Reliability, and Validity of Deep Learning Enabled Bioimage Analyses*. Dataset. `https://doi.org/10.5061/dryad.4b8gtht9d`.

Seidel, Stefan, Felix Müller-Wienbergen, and Jörg Becker. 2010. "The Concept of Creativity in the Information Systems Discipline: Past, Present, and Prospects". *Communications of the Association for Information Systems* 27 (1): 14.

Sejdić, Ervin, Igor Djurović, and Jin Jiang. 2009. “Time–Frequency Feature Representation Using Energy Concentration: An Overview of Recent Advances”. *Digital signal processing* 19 (1): 153–183.

Sezgin, Mehmet, and Bülent Sankur. 2004. “Survey Over Image Thresholding Techniques and Quantitative Performance Evaluation”. *Journal of Electronic imaging* 13 (1): 146–166.

Shankar, Devashish, Sujay Narumanchi, H. A. Ananya, Pramod Kompalli, and Krishnendu Chaudhury. 2017. “Deep Learning based Large Scale Visual Recommendation and Search for E-Commerce”. *CoRR* abs/1703.02344. arXiv: 1703.02344.

Shuvaev, Sergey A, Alexander A Lazutkin, Alexander V Kedrov, Konstantin V Anokhin, Grigori N Enikolopov, and Alexei A Koulakov. 2017. “Dalmatian: An Algorithm for Automatic Cell Detection and Counting in 3d”. *Frontiers in neuroanatomy* 11:117.

Siebert, Sabina, Laura M. Machesky, and Robert H. Insall. 2015. “Point of View: Overflow in science and its implications for trust”. *eLife* 4:e10825.

Simo-Serra, Edgar, Sanja Fidler, Francesc Moreno-Noguer, and Raquel Urtasun. 2015. “Neuroaesthetics in Fashion: Modeling the Perception of Fashionability”. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 869–877.

Simonyan, Karen, and Andrew Zisserman. 2015. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In *International Conference on Learning Representations*.

Sirinukunwattana, Korsuk, Josien PW Pluim, Hao Chen, Xiaojuan Qi, Pheng-Ann Heng, Yun Bo Guo, Li Yang Wang, Bogdan J Matuszewski, Elia Bruni, Urko Sanchez, et al. 2017. “Gland Segmentation in Colon Histology Images: The Glas Challenge Contest”. *Medical image analysis* 35:489–502.

Smith, Leslie N. 2018. “A Disciplined Approach to Neural Network Hyper-Parameters: Part 1–Learning Rate, Batch Size, Momentum, and Weight Decay”. *CoRR* abs/1803.09820. arXiv: 1803.09820.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. “Dropout: A Simple Way to Prevent Neural Networks From Overfitting”. *The Journal of Machine Learning Research* 15 (1): 1929–1958.

Stringer, Carsen, Tim Wang, Michalis Michaelos, and Marius Pachitariu. 2021. “Cellpose: A Generalist Algorithm for Cellular Segmentation”. *Nature Methods* 18 (1): 100–106.

Studer, Stefan, Thanh Binh Bui, Christian Drescher, Alexander Hanuschkin, Ludwig Winkler, Steven Peters, and Klaus-Robert Müller. 2021. “Towards CRISP-ML (Q): A Machine Learning Process Model With Quality Assurance Methodology”. *Machine Learning and Knowledge Extraction* 3 (2): 392–413.

Sutskever, Ilya, James Martens, and Geoffrey E Hinton. 2011. “Generating Text With Recurrent Neural Networks”. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 1017–1024.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. “Going Deeper With Convolutions”. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.

Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. “Rethinking the Inception Architecture for Computer Vision”. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2818–2826.

Tan, Mingxing, and Quoc V. Le. 2019. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. In *Proceedings of the 36th International Conference on Machine Learning*, 97:6105–6114.

Tassoul, Marc, and Jan Buijs. 2007. “Clustering: An Essential Step From Diverging to Converging”. *Creativity and Innovation Management* 16 (1): 16–26.

Tautkute, Ivona, Tomasz Trzcinski, Aleksander Skorupa, Lukasz Brocki, and Krzysztof Marasek. 2019. “DeepStyle: Multimodal Search Engine for Fashion and Interior Design”. *IEEE Access* 7:84613–84628.

Tayler, Kaycie K, Kazumasa Z Tanaka, Leon G Reijmers, and Brian J Wiltgen. 2013. “Reactivation of Neural Ensembles During the Retrieval of Recent and Remote Memory”. *Current Biology* 23 (2): 99–106.

Taylor, Barry N, and Chris E Kuyatt. 1994. *Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results*. Tech. rep. US Department of Commerce, Technology Administration, National Institute of Standards and Technology.

Ullman, Jeffrey. 2021. "On the Nature of Data Science". In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.

Valenti, Michele, Stefano Squartini, Aleksandr Diment, Giambattista Parascandolo, and Tuomas Virtanen. 2017. "A Convolutional Neural Network Approach for Acoustic Scene Classification". In *2017 International Joint Conference on Neural Networks (IJCNN)*, 1547–1554.

Van der Maaten, Laurens, and Geoffrey Hinton. 2008. "Visualizing Data Using T-Sne." *Journal of machine learning research* 9 (11).

Vasileva, Mariya I., Bryan A. Plummer, Krishna Dusad, Shreya Rajpal, Ranjitha Kumar, and David Forsyth. 2018. "Learning Type-Aware Embeddings for Fashion Compatibility". In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XVI*, 11220:405–421.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need". In *Advances in Neural Information Processing Systems*, 5998–6008.

Veit, Andreas, Serge Belongie, and Theofanis Karaletsos. 2017. "Conditional Similarity Networks". In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:1781–1789.

Veit, Andreas, Balazs Kovacs, Sean Bell, Julian McAuley, Kavita Bala, and Serge Belongie. 2015. "Learning Visual Clothing Style With Heterogeneous Dyadic Co-Occurrences". In *2015 IEEE International Conference on Computer Vision (ICCV)*, 4642–4650.

Voigt, Matthias, Björn Niehaves, and Jörg Becker. 2012. "Towards a Unified Design Theory for Creativity Support Systems". In *International Conference on Design Science Research in Information Systems*, 152–173.

Wager, Karen A, Frances W Lee, and John P Glaser. 2017. *Health Care Information Systems: A Practical Approach for Health Care Management*. John Wiley & Sons.

Wang, Guotai, Wenqi Li, Michael Aertsen, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren. 2019. "Aleatoric Uncertainty Estimation With Test-Time Augmentation for Medical Image Segmentation With Convolutional Neural Networks". *Neurocomputing* 338:34–45.

Wang, Jiang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. 2014. "Learning Fine-Grained Image Similarity With Deep Ranking". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1386–1393.

Wang, Wenguan, Yuanlu Xu, Jianbing Shen, and Song-Chun Zhu. 2018. "Attentive Fashion Grammar Network for Fashion Landmark Detection and Clothing Category Classification". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4271–4280.

Wang, Xianwang, and Tong Zhang. 2011. "Clothes Search in Consumer Photos via Color Matching and Attribute Learning". In *Proceedings of the 19th ACM International Conference on Multimedia*, 1353–1356.

Warfield, Simon K, Kelly H Zou, and William M Wells. 2004. "Simultaneous Truth and Performance Level Estimation (STAPLE): An Algorithm for the Validation of Image Segmentation". *IEEE transactions on medical imaging* 23 (7): 903–921.

Wightman, Ross. 2019. "PyTorch Image Models". `https://github.com/rwightman/pytorch-image-models`, *GitHub repository*.

Won Jeong, So, Ann Marie Fiore, Linda S Niehm, and Frederick O Lorenz. 2009. "The Role of Experiential Value in Online Shopping: The Impacts of Product Presentation on Consumer Responses Towards an Apparel Web Site". *Internet Research* 19 (1): 105–124.

Wynants, Laure, Ben Van Calster, Gary S Collins, Richard D Riley, Georg Heinze, Ewoud Schuit, Marc MJ Bonten, Darren L Dahly, Johanna A Damen, Thomas PA Debray, et al. 2020. "Prediction Models for Diagnosis and Prognosis of Covid-19: Systematic Review and Critical Appraisal". *bmj* 369.

Xiao, Bo, and Izak Benbasat. 2007. "E-Commerce Product Recommendation Agents: Use, Characteristics, and Impact". *MIS quarterly* 31 (1): 137–209.

Xie, Enze, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. 2021. "SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers". *CoRR* abs/2105.15203. arXiv: 2105.15203.

Xie, Jiaheng, Daniel Dajun Zeng, Xiao Liu, and Xiao Fang. 2017. "Understanding Reasons for Medication Nonadherence: An Exploration in Social Media Using Sentiment-Enriched Deep Learning Approach". In *38th International Conference on Information Systems: Transforming Society with Digital Innovation, ICIS 2017*.

Yakubovskiy, Pavel. 2020. "Segmentation Models Pytorch". https://github.com/qubvel/segmentation_models.pytorch, *GitHub repository*.

Yamaguchi, Kota, M Hadi Kiapour, Luis E Ortiz, and Tamara L Berg. 2015. "Retrieving Similar Styles to Parse Clothing". *IEEE transactions on pattern analysis and machine intelligence* 37 (5): 1028–1040.

— . 2012. "Parsing Clothing in Fashion Photographs". In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, 3570–3577.

Yang, Wei, Ping Luo, and Liang Lin. 2014. "Clothing Co-Parsing by Joint Image Segmentation and Labeling". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3182–3189.

Yaraghi, Niam, Anna Ye Du, Raj Sharman, Ram D Gopal, and Ram Ramesh. 2015. "Health Information Exchange as a Multisided Platform: Adoption, Usage, and Practice Involvement in Service Co-Production". *Information Systems Research* 26 (1): 1–18.

Yildirim, Gökhan, Calvin Seward, and Urs Bergmann. 2018. "Disentangling Multiple Conditional Inputs in GANs". *CoRR* abs/1806.07819. arXiv: 1806.07819.

Yong, Hongwei, Jianqiang Huang, Xiansheng Hua, and Lei Zhang. 2020. "Gradient Centralization: A New Optimization Technique for Deep Neural Networks". In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, 12346:635–652.

Yosinski, Jason, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. "How Transferable Are Features in Deep Neural Networks?" In *Advances in Neural Information Processing Systems*, 3320–3328.

Zhang, Hang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, and Alexander J. Smola. 2020. "ResNeSt: Split-Attention Networks". *CoRR* abs/2004.08955. arXiv: 2004.08955.

Zheng, Shuai, Fan Yang, M. Hadi Kiapour, and Robinson Piramuthu. 2018. "ModaNet: A Large-Scale Street Fashion Dataset With Polygon Annotations". *2018 ACM Multimedia Conference on Multimedia Conference - MM '18*.

Zhou, Zongwei, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. 2018. "UNet++: A Nested U-Net Architecture for Medical Image Segmentation". In *International Workshop on Deep Learning in Medical Image Analysis*, 3–11.

Zou, Xingxing, Wai Keung Wong, and Dongmei Mo. 2018. "Artificial Intelligence on Fashion and Textiles". In *Proceedings of the Artificial Intelligence on Fashion and Textiles (AIFT) Conference 2018*, 255–267.

Appendix

A List of Publications

A.1 Publications in this Thesis

Griebel, Matthias, Alexander Dürr, and Nikolai Stein. 2019. “Applied Image Recognition: Guidelines for Using Deep Learning Models in Practice”. In *Human Practice. Digital Ecologies. Our Future. 14. Internationale Tagung Wirtschaftsinformatik (WI 2019), February 24-27, 2019, Siegen, Germany*, 393–406.

Griebel, Matthias, Giacomo Welsch, Toni Greif, and Christoph Flath. 2019. “A Picture Is Worth More Than a Thousand Purchases: Designing an Image-Based Fashion Curation System”. In *27th European Conference on Information Systems - Information Systems for a Sharing Society, ECIS 2019, Stockholm and Uppsala, Sweden, June 8-14, 2019*.

Krenzer, Adrian, Nikolai Stein, Matthias Griebel, and Christoph Flath. 2019. “Augmented Intelligence for Quality Control of Manual Assembly Processes Using Industrial Wearable Systems”. In *Proceedings of the 40th International Conference on Information Systems, ICIS 2019, Munich, Germany, December 15-18, 2019*.

Griebel, Matthias, Christoph Flath, and Sascha Friesike. 2020. “Augmented Creativity: Leveraging Artificial Intelligence for Idea Generation in the Creative Sphere”. In *28th European Conference on Information Systems - Liberty, Equality, and Fraternity in a Digitizing World, ECIS 2020, Marrakech, Morocco, June 15-17, 2020*.

Segebarth, Dennis, Matthias Griebel, Nikolai Stein, Cora R. von Collenberg, Corinna Martin, Dominik Fiedler, Lucas B Comeras, Anupam Sah, Victoria Schoeffler, Teresa Lüffe, et al. 2020a. “On the Objectivity, Reliability, and Validity of Deep Learning Enabled Bioimage Analyses”. *eLife* 9:e59780.

Griebel, Matthias, Dennis Segebarth, Nikolai Stein, Nina Schukraft, Philip Tovote, Robert Blum, and Christoph M. Flath. 2021. “Deep-Learning in the Bioimaging Wild: Handling Ambiguous Data With Deepflash2”. *CoRR* abs/2111.06693, currently under review at Nature Methods. arXiv: 2111.06693.

A.2 Other Publications

Segebarth, Dennis, Matthias Griebel, Alexander Dürr, Cora R. von Collenberg, Corinna Martin, Dominik Fiedler, Lucas B Comeras, Anupam Sah, Nikolai Stein, Rohini Gupta, et al. 2018. “DeepFLaSh, a Deep Learning Pipeline for Segmentation of Fluorescent Labels in Microscopy Images”. *bioRxiv*.

Dürr, Alexander, Matthias Griebel, Giacomo Welsch, and Frédéric Thiesse. 2020. “Predicting Fraudulent Initial Coin Offerings Using Information Extracted From Whitepapers”. In *28th European Conference on Information Systems - Liberty, Equality, and Fraternity in a Digitizing World, ECIS 2020, Marrakech, Morocco, June 15-17, 2020*.

Oberdorf, Felix, Nikolai Stein, Nicolas Walk, Matthias Griebel, and Christoph Flath. 2020. “ADR for Big-Data IT Artifact Development: An Escalation Management Example”. In *Proceedings of the 41st International Conference on Information Systems, ICIS 2020, Making Digital Inclusive: Blending the Locak and the Global, Hyderabad, India, December 13-16, 2020*.

B Kaggle Kernels

Table B.1: Kernels to reproduce the results of Section 2.5 on the competition dataset.

Step	URL
Overview	`https://www.kaggle.com/matjes/hubmap-deepflash2-judge-price`
File Conversion	`https://www.kaggle.com/matjes/hubmap-zarr`
Sampling Preparation	`https://www.kaggle.com/matjes/hubmap-efficient-sampling-ii-deepflash2`
Ensemble Training	`https://www.kaggle.com/matjes/hubmap-deepflash2-train`
Validation	`https://www.kaggle.com/matjes/hubmap-deepflash2-validation`
Inference	`https://www.kaggle.com/matjes/hubmap-deepflash2-scaled-ensemble-sumbission`

Table B.2: Kernels to reproduce the results of Section 2.5 on the pancreas dataset.

Step	URL
File Conversion	`https://www.kaggle.com/matjes/cptac-pda-to-zarr`
Sampling Preparation	`https://www.kaggle.com/matjes/cptac-pda-pancreas-efficient-sampling-deepflash2`
Ensemble Training	`https://www.kaggle.com/matjes/cptac-pda-train-deepflash2`
Validation	`https://www.kaggle.com/matjes/cptac-pda-deepflash2-validation/output`