TÉCNICO LISBOA

High-throughput *in silico* characterization of 3D astrocyte-neuron cell cultures

Maria Beatriz Sereno Fernandes

2nd Cycle Integrated Project to obtain the Master of Science Degree in

Biomedical Engineering

Supervisor(s): Prof. Rui Miguel Carrasqueiro Henriques
Dr. Thomas Distler

Examination Committee

Chairperson:
Supervisor: Prof. Rui Miguel Carrasqueiro Henriques
Member of the Committee: Prof. Daniel Faria

January 2024

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

Preliminary confocal microscopy images of astrocytes inside 3D hydrogels were kindly provided by Thomas Distler and Giacomo Masserdotti, Institute of Stem Cell Research, Götz Lab, Helmholtz Zentrum München.

Resumo

Estudos recentes associaram a regulação dos mecanismos de neurogénese como uma das múltiplas funções dos astrócitos no sistema nervoso central. No entanto, devido à vasta complexidade morfológica e heterogeneidade da população de astrócitos, a caraterização completa destas células e a elucidação do seu papel em numerosos mecanismos reguladores permanece um desafio.

O principal objetivo deste projeto é estabelecer uma *pipeline* para a caraterização de culturas 3D de astrócitos-neurónios por meio de análise computacional usando algoritmos de última geração para segmentação e classificação de imagens. Este projeto explora os fundamentos teóricos da imagiologia biológica, assim como técnicas de aprendizagem automática e aprendizagem profunda. Além disso, detalha também uma revisão da literatura relevante abrangendo tarefas de *computer vision* no domínio biomédico, com ênfase na análise de culturas celulares.

Adicionalmente, o Mask R-CNN foi aplicado com sucesso para segmentar núcleos de células 2D a partir de projeções de máxima intensidade de um conjunto de dados de culturas 3D de astrócitos, fornecido pelo Laboratório Götz do Helmholtz Zentrum München. Após a anotação manual e a caracterização das imagens, o Mask R-CNN produziu resultados iniciais promissores com um TPR de 88.91%, PPV de 94.02%, AP médio de 85.04% e IoU médio de 62.72%. O Stardist 2D e o Cellpose 2D foram utilizados para validação.

O êxito desta análise impulsiona-nos a alargar os modelos para o contexto tridimensional. Os próximos esforços focar-se-ão em caracterizar de forma abrangente as culturas de astrócitos em hidrogéis 3D com especial foco na quantificação de células e astrócitos, bem como na realização de uma caracterização detalhada dos astrócitos. Potencialmente, esta caracterização também poderá ser extendida para co-culturas de astrócitos-neurónios.

Palavras-chave: culturas astrócitos-neurónios, segmentação de imagem, deteção de objectos, redes neuronais profundas, Mask R-CNN.

Abstract

Recent research has linked the regulation of neurogenesis mechanisms as one of the multiple functions of astrocytes in the central nervous system. However, due to the immense morphological complexity and heterogeneity within the astrocyte population, thoroughly characterizing these cells and elucidating their roles in numerous regulatory mechanisms remains challenging.

This project's primary objective is to establish a tailored pipeline for the comprehensive characterization of 3D astrocyte-neuron cell cultures through computational analysis using cutting-edge algorithms for image segmentation and classification. Our work builds a strong foundation by exploring essential biological imaging techniques and delving into machine learning and deep learning methodologies. A thorough review of relevant literature extensively covers computer vision tasks in the biomedical domain, especially in cell culture analysis.

Furthermore, we have successfully fine-tuned Mask R-CNN for segmenting 2D cell nuclei from maximum intensity projections of a 3D astrocyte culture dataset kindly provided by the Götz Lab at the Helmholtz Zentrum München. After manual annotation and extensive profiling, Mask R-CNN yield promising initial results with a TPR of 88.91%, PPV of 94.02%, mean AP of 85.04%, and mean IoU of 62.72%. Stardist 2D and Cellpose 2D further validated these outcomes.

Building upon successful 2D image analysis, our forthcoming efforts will concentrate on extending these models to a 3D setting. The goal is to comprehensively characterize astrocyte cultures cultivated in 3D hydrogels focusing on cell and astrocyte quantification, as well as in-depth astrocyte characterization. Potentially, this characterization can also be extended to astrocyte-neuron co-cultures.

Keywords: astrocyte-neuron cultures, image segmentation, object detection, deep, neural networks, Mask R-CNN.

Contents

List of Tables

List of Figures

Nomenclature

Greek symbols

η	Learning rate
λ	Regularization hyperparameter
$\phi(x)$	Activation function

Roman symbols

$\mathcal{X}$	Observation domain
$\mathcal{Z}$	Target domain
$\mathcal{L}(\mathbf{w})$	Loss function
$\hat{z}$	Estimated target
b	Bias
d	Data dimensionality
K	Cardinality of the classification problem
N	Total number of observations
net	Net value of a network layer
w	Weight value of a neural unit
z	Target value
$\tilde{X}$	Design matrix of the observations set
X	Observations set
$\hat{\mathbf{z}}$	Estimated target vector
$\mathbf{w}$	Weights vector
$\mathbf{x}$	Observation
$\mathbf{z}$	Target vector

Subscripts

i, j, k Computational indexes

Superscripts

t Iteration index

Glossary

2D	Bidimensional
3D	Three-dimensional
AI	Aritificial Intelligence
ANN	Artificial Neural Network
AP	Average Precision
ASPP	Atrous Spatial Pyramid Pooling
CNN	Convolutional Neural Network
CRF	Conditional Random Field
DAPI	4',6'-diamino-2-fenil-indol
DIC	Differential Interference Contrast (Microscopy)
DIV	Days *in Vitro*
DL	Deep Learning
DSB	Data Science Bowl Challenge
DT	Decision Tree
EM	Electron Microscopy
FCN	Fully Connected Network
FN	False Negative
FP	False Positive
GFAP	Glial Fibrilary Acidic Protein
IHC	Immunohistochemistry
IoU	Intersection over Union
MAE	Mean Absolute Error
ML	Machine Learning
MSE	Mean Squared Error
NPC	Neural Progenitor Cells
NSC	Neural Stem Cells
PPV	Positive Predictive Value
R-CNN	Region Proposal Convolutional Neural Network
RAFI	Relative Average Flourescence Intensity
RMSE	Root Mean Squared Error

ROI	Region of Interest
RPN	Region Proposal Network
ResNet	Residual Neural Network
SEM	Scanning Electron Microscopy
SGD	Stochastic Gradient Descent
SGZ	Subgranular Zone
SSE	Sum of Squared Residual Errors
SVM	Support Vector Machine
SVZ	Subventricular Zone
TEM	Transmission Electron Microscopy
TL	Transmitted-Light (Microscopy)
TPR	True Positive Ratio
TP	True Positive
VGG	Very Deep Convolutional Network
YOLO	You Only Look Once (Network)

Chapter 1

Introduction

Aldult neurogenesis is currently in the scientific spotlight, with researchers passionately unraveling its intricacies. Until the 1960s, prevailed the belief that neurogenesis was limited to the prenatal and postnatal periods, and once the neural circuits were established, the production of new neurons would cease. The established premise was revised in the sixties giving rise to the hypothesis that new neurons could be generated during adulthood. However, it was not until the late 1990s that evidence of adult neurogenesis was reported in mammals such as rodents, monkeys, and humans (Araki et al., 2021).

While there is some debate about which telencephalon areas retain neurogenic abilities, it is generally accepted that neurogenesis persists in the brain's subgranular (SGZ) and subventricular zones (SVZ). Neurogenesis on the SGZ is mainly responsible for modulating the functions of the hippocampus, and it is believed to play an active role in the individual's learning, memory, and behaviour. On the other hand, the new neurons originated in the SVZ are incorporated into the olfactory bulb through the rostral migratory stream and are responsible for odorant discrimination, olfactory learning, and long-term memory (Pérez-Rodríguez et al., 2021).

The process of neurogenesis remains a topic of ongoing research, as its full mechanisms remain to be completely comprehended. Recent scientific efforts have focused on unravelling this process, given its significant implications for embryonic development, cognitive function, mental health, ageing, and neurological disorders. The three consensual origins of the new neurons are: neural stem cells (NSCs), neural progenitor cells (NPCs), and non-neural glial cells, namely astrocytes (Götz & Huttner, 2005). The latter is particularly intriguing since it involves a process of *dediferentiation* in which a cell returns to a less-specialised cell fate. The ability of astrocytes to be epigenetically reprogrammed to generate neurons has been confirmed both naturally (*in vivo*) and artificially (*in vitro*) (Grade & Götz, 2017; Griffiths et al., 2020).

Traditionally, neuroscience research has predominantly centered on neurons, the essential units responsible for transmitting messages throughout the body via chemical or electrical signals (Kandel et al., 2000). Neurons are often regarded as the foundational components of the nervous system due to their ability to process and relay information. However, more recently, the roles of glial cells have been gaining widespread recognition for carrying indispensable functions in the brain during its development, mainte-

nance, and response to disease and trauma (Araki et al., 2021). The glial cell population is traditionally subdivided into microglia, astrocytes, and oligodendrocytes (Araki et al., 2021; Pérez-Rodríguez et al., 2021). Microglia serve as the immunocompetent and phagocytic cells of the nervous system, resembling the functions of peripheral macrophages/monocytes and originating from the yolk-sac progenitors that populate the brain during development (Jäkel & Dimou, 2017). Conversely, astrocytes and oligodendrocytes derive from NSCs, identicaly to neurons. Astrocytes, in particular, exhibit many similarities with NSCs, which can hinder the distinction of functions between the two (Cassé et al., 2018).

Regarding adult neurogenesis, the role of microglia and astrocytes must be emphasized in regulating the microenvironment conditions in the neurogenic niches. Firstly, given the immunity facet of microglia, they are responsible for maintaining the environment through their phagocytic capacity, the interaction with neurons through the fractalkine-CX3CR1 signalling pathway, the release of growth factors, and cytokines (Araki et al., 2021). On the other hand, astrocytes release soluble factors, such as adenosine triphosphate (ATP) and lactate, to promote the development and differentiation of adult-born neurons. Furthermore, they play an active role in maintaining homeostasis and integrating the new neurons into the existing neural circuits. A single astrocyte is capable of regulating up to 2.000.000 synapses in the human brain (Oberheim et al., 2009), functionally creating a tripartite synapse by controlling the neural activity by uptaking or releasing neurotransmitters into the synapsis' interstitial space (Araki et al., 2021).

1.1 Motivation

Considering astrocytes' pivotal role in neurogenesis, it becomes evident the immediate necessity to thoroughly understand the key functions of this cell population to elucidate the mechanisms underpinning neurogenesis.

Several studies have identified the relevance of characterizing the astrocyte cultures (Healy et al., 2018; Huang et al., 2022; Kulkarni et al., 2015). The main focuses rely on the detection, segmentation, morphological analysis, spatial profiling, cell-cell contacts, and classification of astrocyte and astrocyte-neuron cultures. However, the analysis of increasingly large and three-dimensional (3D) image datasets poses a challenge for manual techniques, pushing the incorporation of high-throughput automated analysis into modern laboratories. Nowadays, deep learning is the primary tool of choice for automating biological image analysis (H. Wang et al., 2018).

The first barrier to studying astrocytes is their morphological complexity and high heterogeneity (Allen & Eroglu, 2017; Huang et al., 2022). These characteristics hinder the direct application of many general-purpose algorithms for biological image analysis and the characterization of astrocyte cultures. Secondly, annotated astrocyte data's scarcity is well acknowledged in the literature (Huang et al., 2022). In addition, laboratory-specific experimental conditions largely determine the composition of differentiated astrocyte cultures, further challenging the cross-experimental generalization capacity of the algorithms (Kayasandik et al., 2020). Nevertheless, specific strategies exist, such as transfer-learning and fine-tuning, to circumvent this issue that we shall later elaborate.

Furthermore, most annotated datasets in existing literature primarily feature bidimensional (2D) im-

ages. Given the growing enthusiasm for the *in vitro* replication of 3D models of the human brain to more accurately resemble the *in vivo* cellular interactions and development (Shou et al., 2020), there arises a compelling need for the empirical exploration of 3D analysis in these cultures or even in *ex vivo* scenarios and animal models.

1.2 Objectives and Deliverables

The primary computational objective of the Master's thesis project is to develop a high-throughput analysis pipeline tailored for the characterization of 3D astrocyte-neuron cultures. Concurrently, wet lab experiments will involve the cultivation of murine-derived astrocytes in diverse 3D hydrogels. Furthermore, the aim is to establish a high-throughput confocal microscopy platform to streamline the acquisition of intricate images crucial for computational analysis.

The overarching goal of the computational project is to develop a comprehensive pipeline for cell nuclei quantification, astrocyte identification, and the classification of astrocytes into distinct subtypes based on morphological features such as protoplasmic-like or fibrous-like types, as well as mononuclear or multinuclear states. Alongside pipeline development, our objective is to curate a meticulously annotated dataset of 3D astrocyte cultures to facilitate their thorough characterization.

Initially, we plan to leverage well-established deep convolutional neural networks (CNNs) specifically designed for segmenting and classifying 2D cell images. We aim to fine-tune these networks using 2D projections of 3D astrocyte cultures, employing transfer learning techniques to accommodate the unique characteristics of glial and nerve cells. Subsequently, our goal is to extrapolate and adapt these models to process 3D image data, particularly z-stack images. Recognizing the limited dataset availability, our strategy involves implementing data augmentation methodologies to maximize model efficiency despite the constraints of a relatively small dataset.

Finally, we will also attempt to extend our pipeline beyond individual cell analysis and delve deeper into the realm of cell interactions and *in silico* staining. Effectively characterizing overlapping regions with connectomics-based techniques serves as an invaluable tool for comparing cultures under various conditions, while the *in silico* staining technique will allow to characterize cultures in a more efficient and reproducible manner, and permitting the shift from fluorescent to transmitted-light microscopy.

In this project, our aim was to thoroughly explore fundamental concepts and conduct an extensive review of the existing tools pertinent to the proposed objectives. Additionally, a provisory set of illustrative z-stack cell images collected by the Götz Lab group at the Helmholtz Zentrum München was profiled. There was also undertaken a preliminary assessment of the selected state-of-the-art tools for cell imaging segmentation to understand their intrinsic behavioral limitations and establish performance baselines. Our primary focus remained on astrocyte cultures.

1.3 Project Outline

This report is structured to provide a comprehensive understanding of vital biological imaging techniques, image processing tools pertinent to 3D astrocyte cell cultures, and the core machine learning and deep learning concepts pivotal for constructing a robust characterization pipeline for astrocyte-neuron cell cultures. Note that this work does not delve into the methodologies used for the astrocyte cultures nor the specific techniques used for image acquisition, since these details are still under adjustments by our partners at the Helmholtz Zentrum München.

Chapter 2 initiates with an exploration of foundational theoretical aspects in biological imaging techniques (section 2.1), machine learning (section 2.2), and deep learning principles (section 2.3). Furthermore, section 2.4 explores convolutional neural networks, which are pivotal in tasks such as classification, object detection, and segmentation within computer vision.

The contemporary state-of-the-art work is explored in chapter 3. Particular emphasis was placed in the most used architectures for image classification (section 3.1), object detection (section 3.2), image segmentation (section 3.3), and *in silico* staining (section 3.4) in biological domains, detailing both strengths and weaknesses.

In chapter 4 we delve into the characterization of the astrocyte cultures dataset of the Helmholtz Zentrum München using Fiji's python packages (section 4.1). In addition, we detail the dataset annotation strategies employed and some of the most preponderant dataset characteristics (section 4.2).

In chapter 5, we detail the proposed approach for the development of an integrated pipeline capable of nuclei quantification, astrocyte identification, and the classification of astrocytes into distinct subtypes based on morphological features and multinucleated state. This proposal is solely based on previously developed pipelines and network architectures, and its primary purpose is to evaluate the performance of existing models.

The preliminary results of the performance of existing architectures in segmentation tasks using the astrocyte-culture dataset are detailed under chapter 6.

The concluding chapter, chapter 7, explores the main achievements of the present work and draws future directions.

Chapter 2

Background

This chapter serves as an encompassing cornerstone, introducing fundamental concepts related to biological imaging techniques, machine learning, and deep learning tools pivotal to the succeeding thesis chapters. Its primary goal is to acquaint the reader with the foundational knowledge necessary to grasp the methods and techniques elaborated upon in subsequent endeavors.

2.1 Biological Imaging Techniques

Microscopy relies on two fundamental methodologies: optical and electron microscopy (EM), each diverging in their core working principles. While optical microscopy utilizes light to capture a magnified image of the sample, EM operates on electron-based techniques. Electron microscopy encompasses two primary methods: transmission electron microscopy (TEM), which involves transmitting electrons through thin sample sections, and scanning electron microscopy (SEM), which scans the three-dimensional surface of the sample (Inkson, 2016). EM is renowned for its unparalleled resolution and magnification and has proven considerably beneficial in several biological imaging settings, such as for imaging subcellular components (Lam et al., 2021). However, we will abstain from delving deeper into these intricacies in this project as they are not the main focus.

Within biomedical imaging, optical microscopy stands as the conventional approach. Depending on the specific aims of a study, various techniques are already available for imaging analysis. It is crucial to acknowledge that each technique can capture distinct dimensional aspects of a sample. Therefore, selecting the appropriate microscopy technique should align with the study objectives. In this project, we narrow our focus to three critical optical microscopy techniques due to their prevalent applications in cellular imaging: transmitted-light (TL), fluorescent, and confocal microscopy (Christiansen et al., 2018).

Transmitted-light microscopy, as opposed to reflection microscopy, is employed to glean insights from samples by transmitting light through them and is notably prevalent in examining biological specimens. Bright-field microscopy stands out as one of the most widely utilized configurations employed in TL microscopes. This method involves illuminating the sample with light, leveraging variations in light absorption within denser regions of the sample to create contrast. Staining the samples prior to imaging is

customary to enhance visibility, aiding in accentuating contrasts. Conversely, phase-contrast microscopy captures images contingent upon variations in cell component thickness and refractive indices. Despite being notably effective in thin samples, this technique may encounter limitations with thicker specimens. In addition, differential interference contrast (DIC) microscopy is often employed for enhanced imaging of thicker samples (Stylianou et al., 2021).

Modern cell research employs different methods to target, label, and study cells using fluorescence microscopy. These strategies can be grouped into two main categories: genetic and small molecule. In genetic approaches, cells can be genetically modified using, for instance, fluorescent reporters expressed by the promoters of upregulated proteins in the target cell or fluorescent antibodies used to label fixed samples by immunohistochemistry (IHC). Conversely, small molecule strategies rely on cell-resident transporters to travel into the target cells and selectively label them (Preston et al., 2019).

Genetic strategies are the most common for the study of astrocytes due to the scarcity of astrocyte-specific chemical markers for small molecule labeling. Glial fibrillary acidic protein (GFAP), S100 calcium-binding protein B (S100β), and aldehyde dehydrogenase 1 family member L1 (Aldh1L1) are the most commonly used astrocyte IHC labeling proteins. GFAP participates in the astrocyte response to injury by contributing to scar formation. It is the most abundant intermediate filament protein in astrocytes, which endows it as the most widely used astrocyte labeling protein. Nonetheless, GFAP has some limitations, including a higher prevalence in white matter than gray matter, an inability to label all astrocytic processes, and variable expression levels across diverse astrocyte populations. On the other hand, S100β, responsible for mediating the interactions between glial cells and neurons, labels the cell nucleus and the cytoplasm of astrocytes. However, S100β is not exclusive to astrocytes as it can also be found in subsets of neurons, oligodendrocytes, and neural precursor cells. In some studies, Aldh1L1, involved in astrocyte folate metabolism, has shown broader expression than GFAP. Moreover, it exhibits greater accuracy in labeling astrocytic morphology than GFAP, revealing finer processes. Therefore Aldh1L1 is sometimes used simultaneously with GFAP (Preston et al., 2019).

In most fluorescent microscopy cellular images, it is usual to mark the cell nucleus with a fluorescent marker. The most extensively used nuclear marker is 4',6'-diamino-2-fenil-indol (DAPI). DAPI marks both live and fixed cells effectively through its strong binding to adenine-thymine-rich regions in the DNA. Additionally, the transcription factor SOX9 has proven to be an effective astrocyte-specific nuclear marker in the central nervous system outside the neurogenic regions (Sun et al., 2017).

In addition, confocal microscopy is renowned for its capacity to obtain high-resolution images in thick samples. Its fundamental operation involves creating a focal point of light while effectively eliminating out-of-focus light. This mechanism empowers the microscope to penetrate deep into tissues, facilitating precise optical sectioning of the samples. Comprehensive high-resolution 3D image stacks can be generated by reconstructing these sections. Specifically, fluorescence confocal microscopy finds extensive application in the 3D reconstruction of cellular images. Employing fluorescent labels to mark cells enables the accurate determination of protein and cellular structure localization within 3D cell cultures (Elliott, 2020).

2.2 Machine Learning

In modern biology and biomedical sciences, artificial intelligence (AI) has emerged as a widely adopted approach for managing biological data's increasing volume and intricate nature. Within AI, machine learning (ML) is defined as a set of approaches for automatically detecting patterns within data (Murphy, 2012). Its applications can be used to learn predictors for decision-making or descriptors for knowledge acquisition.

Despite the broad applicability and efficiency of classic machine learning techniques to process simple multivariate data, in more complex data domains the scenario is not usually the same. Until the 2010s, image data analysis often required careful feature engineering or considerable domain expertise to transform the raw data into a more suitable internal representation, which ML could not achieve on its own. Contrarily, deep learning (DL) techniques are representation-learning methods that automatically uncover data representations necessary for the learning task. Therefore, with careful architecture design and hyperparameter tuning, they are generally more effective approaches when working with complex data domains (LeCun et al., 2015).

2.2.1 Supervised Learning

Supervised learning stands as the predominant paradigm in the realm of machine learning. This approach is primarily employed for predictive tasks, in which the overarching objective is to acquire a mapping function that can relate inputs, $\mathbf{x} \in \mathcal{X}$, to corresponding outputs or targets, $\mathbf{z} \in \mathcal{Z}$. The process of training a supervised learning model relies on the use of annotated or labeled data. Therefore, the model is trained on a dataset consisting of input-output pairs, formally denoted as $\mathcal{D} = (\mathbf{x}_i, \mathbf{z}_i)_{i=1}^{N}$, where N denotes the number of observations (Goodfellow et al., 2016).

The nature of the observations can vary substantially depending on the problem. In simple multivariate structures, observations are generally described by an ordered set of features. Alternative data structures are also prominent in different domains, such as time series, text, and two-dimensional or three-dimensional images. Output variables can be either numerical or categorical, associated with regression or classification problems.

In the context of biological image analysis, a regression task could entail the estimation of properties like cell surface area or volume, while classification tasks might involve identifying the spatial localization of cells within the image or classifying them into different cell types. In both of these endeavors, the framework of supervised learning prevails, necessitating the prior manual annotation of training data.

2.2.2 Classification and Regression

In single-output classification problems, the objective is to learn a mapping function, denoted by $f : \mathbb{R}^d \rightarrow \{c_1, ..., c_K\}$, where $c_k, k = 1, ..., K$ are the possible classes and K is the cardinality. Binary and multiclass forms of classification are observed when $K = 2$ and $K > 2$, respectively (Murphy, 2012).

There are several traditional ML approaches to deal with classification problems. Probabilistic ap-

proaches, such as the Bayesian classification, use the Bayes theorem to predict the class of a given unobserved sample, $\mathbf{x}_{new} \in \mathbb{R}^d$, where d is the dimensionality of the data. This is accomplished by estimating the class that maximizes the posterior probability, $\hat{z} = \arg\max_c\{p(c_k|\mathbf{x}_{new})\}$. Another popular approach, based on information theory, is the decision tree (DT) classifier. DTs recursively partition the data space into subspaces that contain mainly observations of only one class, with relatively few exceptions. Additionally, a support vector machine (SVM) aims to learn the optimal linear hyperplane that maximizes the distance between classes in transformed data spaces, an approach with well-established relevance in diverse domains (Zaki & Meira Jr, 2020).

Conversely, in a single-output regression problem, the objective is to learn a mapping function $f : \mathbb{R}^d \to \mathbb{R}$, where the output is real-valued (Murphy, 2012). Linear regression and variations such as Ridge and Lasso regressions are some of the most common regression ML approaches.

In linear regression, the output is estimated through a linear combination of the input variables with a learned weight vector $\mathbf{w} = [w_0\, w_1\, \cdots\, w_d]$, where w_0 is the estimated bias and w_j the weight associated to each observation's feature, x_j, $j = 1, \ldots, d$. Thus, given a dataset X containing $\{\mathbf{x}_i\}_{i\in\{1,\ldots,N\}}$ observations whose design matrix is $\tilde{X} = [\mathbf{1}\, \mathbf{x}_1\, \cdots\, \mathbf{x}_N]^T$, the estimation output is $\hat{z}_i = \mathbf{w}^T\mathbf{x}_i$. In a regression problem, the objective is to minimize a loss function that quantifies the difference between the predicted and the real values. In linear regression, the method of the least squares, which aims to minimize the sum of squared residual errors (SSE),

$$E_{SSE}(\mathbf{w}) = \sum_{i=1}^{n}(z_i - \hat{z}_i)^2, \tag{2.1}$$

is the most used. The weight vector that minimizes the loss function can be derived from setting the gradient of the SSE to zero, yielding $\mathbf{w} = (\tilde{X}^T\tilde{X})^{-1}\tilde{X}^T\mathbf{z}$ (for detailed derivation see Zaki and Meira Jr). Another alternative is to use the stochastic gradient descent (SGD) update rule. With SGD, the weight vector is randomly initialized and updated recursively with one random observation each time at a predefined learning rate, η,

$$\begin{aligned}\mathbf{w}^{t+1} &= \mathbf{w}^t - \eta \cdot \nabla_{\mathbf{w}}(\tilde{\mathbf{x}}_k) \\ &= \mathbf{w}^t + \eta \cdot (z_k - \tilde{\mathbf{x}}_k^T\mathbf{w}^t) \cdot \tilde{\mathbf{x}}_k,\end{aligned} \tag{2.2}$$

where $\tilde{\mathbf{x}}_k = [1\, x_1\, \cdots\, x_k]$.

Nevertheless, linear regression estimators are often prone to overfitting the training observations, particularly when considering data representations in high-dimensional spaces. To combat this problem, regularization parameters can be added to the SSE, such as in the Ridge and Lasso regressions:

$$E_{Ridge}(\mathbf{w}) = \sum_{i=1}^{n}(z_i - \hat{z}_i)^2 + \lambda\|\mathbf{w}\|_2^2, \tag{2.3}$$

$$E_{Lasso}(\mathbf{w}) = \sum_{i=1}^{n}(z_i - \hat{z}_i)^2 + \lambda\|\mathbf{w}\|_1. \tag{2.4}$$

In Ridge regression, the penalty term shrinks the weight coefficients towards zero, improving their stability (Goodfellow et al., 2016). Contrastingly, the Lasso regression typically produces more sparse solutions since the minimal achievable loss often occurs on an axis, which sets the variable's weight to zero (Russell & Norvig, 2021). Furthermore, the strength of the regularization term is adjusted by λ hyperparameter, which needs to be optimized by cross-validation (Wichert & Sa-Couto, 2021).

Additionally, perceptrons are computational units that aim to mimic the human neuron and can solve simple regression and classification problems. These units effect a weighted summation of their inputs, resulting in a net value, and subsequently apply a nonlinear activation function, ϕ, to produce the output,

$$\hat{z} = \phi\left(\sum_{j=1}^{d}(w_j x_j) + b\right) = \phi(\mathbf{w}^T\mathbf{x} + b) = \phi(net). \tag{2.5}$$

Activation functions are crucial as they allow the capture of nonlinear features without requiring the input transformations, setting perceptrons apart from conventional linear regressors, and are chosen according to the problem. The most simple among these is the identity function, commonly used in regression problems and returns its argument. For binary classification problems, the step function,

$$\textit{step}(net) = \begin{cases} 0, & net \leq 0 \\ 1, & net > 0 \end{cases}, \tag{2.6}$$

was the originally proposed activation function. Nevertheless, its discrete mapping outputs do not allow to discern the observations' distance to the modeled hyperplane, potentially hampering the learning process. Consequently, the step function has largely been deprecated in most applications and replaced by alternative methods. With the sigmoid function,

$$\sigma(net) = \frac{1}{1 + e^{-net}}, \tag{2.7}$$

the output of the perceptron can be thought of as a probability between 0 and 1, which enables the conceptualization of the accuracy of the classification. On the other hand, the hyperbolic tangent,

$$\textit{tanh}(net) = \frac{e^{net} - e^{-net}}{e^{net} + e^{-net}}, \tag{2.8}$$

has a shape similar to the sigmoid, but the output ranges between -1 and +1. Finally, the rectifier linear unit (ReLU) function,

$$\textit{ReLU}(net) = \begin{cases} 0, & net \leq 0 \\ net, & net > 0 \end{cases}, \tag{2.9}$$

is a popular choice when learning neural networks described by multi-layer compositions of perceptrons.

Machine learning tasks often involve predicting multiple unrelated target variables simultaneously, leading to what is known as a multi-output problem. These target variables can span both categorical and numerical domains, resulting in a multi-label classification or a multi-output regression problem, respectively. These different types of targets can also coexist within the same multi-output context.

2.2.3 Evaluation Metrics

In supervised learning problems, the primary objective is to learn the optimal model that estimates an output $\hat{\mathbf{z}}$, as accurately as possible. To assess the performance of a given predictor without incurring biases, a common strategy is to split the dataset into a training and a testing set with the hold-out method. This method allows to train and test the model with independent sets of observations, generally with a 70-30% or 80-20% ratio. Moreover, it is important to bear in mind that a substantial amount of training data is necessary to produce an accurate model. Hence, in scenarios where the data is limited, the k-fold cross-validation method can be more appropriate. This method divides the original dataset X into k mutually exclusive subsets: $X = \{X_1, ..., X_k\}$. Afterwards, it trains and validates the architecture k times, using $X \setminus X_k$ as the training set and X_k as the testing set. Furthermore, in stratified k-fold cross-validation, the data is split so the targets are identically distributed in every partition.

The most general classification performance metrics are the *error rate* and *accuracy*. The error rate,

$$\textit{error rate} = \frac{1}{n}\sum_{i=1}^{n} I(z_i \neq \hat{z}_i), \tag{2.10}$$

measures the percentage of the model's incorrect predictions over the training set, where I is an indicator function that assumes the value 1 when the argument is true and 0 otherwise. Whereas the accuracy,

$$\textit{accuracy} = \frac{1}{n}\sum_{i=1}^{n} I(z_i = \hat{z}_i) = 1 - \textit{error rate}, \tag{2.11}$$

is the percentage of the model's correct predictions.

On the other hand, the model's performance can be assessed by evaluating class-conditional metrics. In a multi-class problem, these metrics include the *precision*, *recall*, and *F1-score*. The precision,

$$\textit{precision}_k = \frac{n_{kk}}{m_k}, \tag{2.12}$$

is equivalent to the class-conditional accuracy, measuring the percentage of correct predictions of class c_k, n_{kk}, over all observations predicted to be in that class, m_k. Conversely, the recall of the class c_k,

$$\textit{recall}_k = \frac{n_{kk}}{n_k}, \tag{2.13}$$

corresponds to the fraction of correct predictions of that class over all observations from the same class, n_k. However, there is often a tradeoff between the precision and recall that does not allow a complete overview of the model's performance. Thus, the class-conditional F1-score, also regarded as Dice's coefficient,

$$\textit{F1-score}_k = 2 \cdot \frac{\textit{precision}_k \cdot \textit{recall}_k}{\textit{precision}_k + \textit{recall}_k} = \frac{2 \cdot n_{kk}}{n_k + m_k}, \tag{2.14}$$

computes their harmonic mean. Additionally, the overall F1-score for the classifier,

$$\textit{F1-score} = \frac{1}{K}\sum_{k=1}^{K} \textit{F1-score}_k, \tag{2.15}$$

can be obtained by averaging the class-conditional F1-scores.

In the multi-label classification scenario, the most common performance metric is the Jaccard index, also known as the intersection over union (IoU) coefficient. Considering a multi-label problem with cardinality L, in which the objective is to estimate L categorical targets, $\hat{z}_i^{(j)}$, for each observation $\mathbf{x}_i$, with $i = 1, ..., N$ and $j = 1, ..., L$, the IoU,

$$\textit{IoU}_i = \frac{1}{L} \cdot \sum_{j=1}^{L} I(z_i^{(j)} = \hat{z}_i^{(j)}), \tag{2.16}$$

is the fraction of accurate target-variable classifications over the total number of target-variables. Furthermore, the mean IoU provides a global metric for the model's performance.

In regression problems, SSE has been previously discussed as a crucial metric for training a regression model. However, its utility extends beyond training; it can also serve as a valuable tool for assessing the quality of the resulting regressor. Additionally, the residuals, which are the absolute differences between the actual and the predicted values, are highly informative in evaluating a regressor. With these residuals, we can compute various performance metrics, including the mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE),

$$\textit{MAE} = \frac{1}{n}\sum_{i=1}^{n} |z_i - \hat{z}_i|, \qquad \textit{MSE} = \frac{1}{n}\sum_{i=1}^{n} (z_i - \hat{z}_i)^2, \qquad \textit{RMSE} = \sqrt{\textit{MSE}}. \tag{2.17}$$

2.3 Deep Learning

Deep learning techniques are motivated by the inefficacy of traditional ML algorithms to answer more complex supervised learning problems (Goodfellow et al., 2016). At the core of DL models are artificial neural networks (ANNs), which are computational models inspired by the structure and functioning of the human brain and use perceptrons as their basic units.

An artificial neural network emerges from the interconnection of multiple perceptrons, featuring an input layer responsible for aggregating the input data $\mathbf{x}$, hidden layers that perform intermediate computations, and an output layer where the final prediction $\hat{\mathbf{z}}$ is generated. In more intricate problem domains, the network's complexity must be enhanced, often necessitating the inclusion of a greater number of hidden layers. Such networks are commonly referred to as deep neural networks.

2.3.1 Artificial Neural Networks

Artificial neural networks possess the ability to effectively model both regression and classification problems by learning network weights that minimize a loss, $\mathcal{L}(\mathbf{w})$, associated with the error between the

estimation and the targets. Generally, the weights update is performed iteratively via gradient descent rules (Zaki & Meira Jr, 2020).

In regression tasks, a common loss function is the squared error loss function,

$$\mathcal{L}_{SSE}(\mathbf{w}) = \frac{1}{2}\|\mathbf{z} - \hat{\mathbf{z}}\|^2 = \frac{1}{2}\sum_{j=1}^{L}(z_j - \hat{z}_j)^2, \tag{2.18}$$

where L is the dimensionality of the response vector, which employs the principle adjacent to the classical regression learning problems debated above (equation 2.1).

In classification tasks, the output layer generally has as many neurons as the number of classes in the problem, and the target response is commonly encoded as a one-hot vector. Thus, in a multi-class classification problem, the softmax function,

$$\textit{softmax}(net_i) = \frac{e^{net_i}}{\sum_{j=1}^{L} e^{net_j}}, \tag{2.19}$$

is applied on the output layer to associate each output neuron to a rough probability estimate. In contrast with regression, the loss function for multi-class classification tasks is often modeled by the cross-entropy loss function,

$$\mathcal{L}_{CE}(\mathbf{w}) = -\sum_{j=1}^{L} z_j \cdot \ln(\hat{z_j}). \tag{2.20}$$

The gradient descent method can minimize the loss function when the non-linear activation functions in the network are differentiable. The gradient descent updates the model weights in the opposite direction of the gradient of the loss function, $\nabla\mathcal{L}(\mathbf{w}^t) = \frac{\partial\mathcal{L}(\mathbf{w}^t)}{\partial\mathbf{w}^t}$, at a predefined learning rate, $\eta > 0$,

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta\nabla\mathcal{L}(\mathbf{w}^t), \tag{2.21}$$

where t is the iteration index. The learning rate plays an essential role in the learning convergence of the model, thus it should be properly tuned (Wichert & Sa-Couto, 2021).

Additionally, the weights can be updated considering all observations by summing the gradients produced by each sample individually, with a batch gradient descent approach, or consecutively with one observation at a time, with stochastic gradient descent. While the batch gradient descent is more stable, it is slower and easily tied in local minima. Therefore, if the dataset is too large, adopting SGD or establishing a trade-off between the two poles might be preferable via mini-batch gradient descent.

To avoid overfitting the training dataset, *dropout* is a common strategy to decrease the test dataset error. Dropout is a stochastic regularization method in which a subset of hidden units is chosen and deactivated at each training step. This way, the deactivated units are not allowed to learn, and the model is less likely to overfit. Additionally, the training is faster since the network has fewer parameters at each training step. The dropout ratio hyperparameter is manually defined and sets the percentage of hidden units deactivated at each step.

2.3.2 Dataset Augmentation, Transfer Learning, and Fine-Tuning

Developing a robust deep learning model requires a substantial amount of annotated training data. This is often challenging, particularly in the field of biology, where data collection is typically labor-intensive and costly. To overcome this hurdle, data augmentation techniques can be a valuable resource.

Data augmentation involves creating synthetic data and incorporating it into the dataset. This approach has proven particularly effective in addressing image classification problems, such as object recognition and object classification (Goodfellow et al., 2016).

Common image augmentation operations encompass translation, rotation, scaling, cropping, and noise injection. These operations are relatively straightforward to implement and have been demonstrated to significantly enhance the model's performance. Furthermore, some researchers have delved into introducing noise to the hidden units of ANNs, extending data augmentation to multiple levels of abstraction. Their findings demonstrate high efficacy in this approach (Goodfellow et al., 2016).

Transfer learning is an additional strategy to manage the scarcity of observations in a dataset. It allows using a partially learned neural network in an arbitrary learning task and extrapolate it to another learning task. Importantly, many of the factors that explain variations in the dataset of the original task are also relevant for the variations that need to be captured in the dataset of the target task, under the assumption they are similar to some extent (Goodfellow et al., 2016).

In the computer vision setting, many datasets can share low-level notions of edges and visual shapes. The effects of shared geometric transformations can aid the learning process of a new task while accelerating its training. These features are preserved in the first layers of the pre-trained model, while the problem-specific features can be learned by the model's deeper layers (predictor layers).

Additionally, fine-tuning is a valuable approach within the realm of transfer learning. While the simplest forms of transfer learning maintain the pre-trained model's initial parameters, fine-tuning takes it a step further by allowing adjustments to better align with the characteristics of a new dataset.

Fine-tuning typically unfolds in two distinct phases. In the feature extraction phase, the parameters of the top layers are held constant from the pre-trained model, while the parameters of the predictor layers are learned using the new dataset as input. This allows the model to capture domain-specific features from the new data and learn the prediction task. In the fine-tuning phase, some of the pre-trained model layers are unfrozen, making them adaptable to the new dataset during the following training. This two-step process enables the model to leverage the knowledge gained from its initial pre-training while tailoring its parameters to the specific requirements of the new task or dataset (Goodfellow et al., 2016).

2.4 Convolutional Neural Networks

Tailoring the choice of the neural network to the intricacies of a given problem is of pivotal importance to obtain optimal results. In the contemporary landscape of neural networks, diverse architectures are designed to respond to specific problem domains. Convolutional Neural Networks (CNNs) applications are notable in computer vision's domain, including object detection, classification, and segmentation.

Due to the intrinsic properties of an image, its meaning is generally not driven by individual input pixel values but by their adjacency patterning. Due to this property, if we were to construct a fully connected network (FCN) with an image as input, the network would yield comparable results regardless of the spatial organization of the input pixels during the training process. Additionally, it would be intractable to compute the weights of a FCN for large images (Russell & Norvig, 2021).

CNNs can be thought of as a specialized type of multilayer perceptron that operates in a localized and sparse manner and is designed to exploit the spatial structure of the input data. The definition of a CNN is any neural network that contains a convolutional layer. Accordingly, the traditional CNN architecture comprises an input layer, followed by several convolutional-pooling layers, one or multiple fully connected layers, and an output layer. Note that fully connected layers are essential to map the output to the target vector.

The design of a CNN respects three fundamental principles: sparse interactions, parameter sharing, and equivariance to translation. Firstly, CNNs establish sparse interactions between its neurons by employing a filter, also referred to as a *kernel*. This strategic application to multiple image regions facilitates the detection of meaningful features at lower levels, such as edges. Secondly, the parameter-sharing feature distinguishes CNNs by applying the same kernel across multiple image regions. In contrast to FCNs, which compute distinct weights for every non-linear relation between input and output, CNNs preserve the weight values in each convolutional layer. Finally, the equivariance to translation property ensures that the output adapts correspondingly to changes in the input, meaning that if an object undergoes translation in the input, its representation will undergo a parallel translation in the output of a convolutional layer. These features refine the network's ability to discern intricate details by connecting a subset of neurons in layer p to a single neuron in layer $p+1$ and substantially reduce the number of parameters stored, also contributing to a more generalized model (Goodfellow et al., 2016).

The standard CNN layer unfolds into three distinct stages. In the initial convolution stage, multiple convolutions are executed concurrently, generating an array of linear activations. Following this, the activation stage introduces a crucial non-linear element by applying an activation function to the net values. The ReLU activation function stands out as a prevailing choice. Finally, a pooling function is applied further modifying the output for subsequent layers. In certain instances, pooling is essential to handle the classification of inputs of varying sizes by extracting the same number of summary features from the convolutional layer, regardless of the input size.

2.4.1 Convolution, Pooling and Upsampling

The convolution can be thought of as the integration (or summation in the discrete scenario) of the product obtained by sliding one function, $w(t)$, over the other, $x(t)$,

$$s(t) = (x * w)(t) = \int x(\tau) \cdot w(t-\tau) d\tau. \tag{2.22}$$

Translating to the convolutional network terminology, one can think of $x(t)$ as the input of the network, and $w(t)$ as the kernel. Under this light, the output of the convolution operation, $s(t)$, denotes the feature

map (Zaki & Meira Jr, 2020).

Considering a 2D input, let $\mathbf{X} \in \mathbb{R}^{n \times n}$ be the input matrix, and let $\mathbf{W} \in \mathbb{R}^{k \times k}$ be the matrix of weights, with $k \leq n$. Let also $\mathbf{X}_k(i,j) \in \mathbb{R}^{k \times k}$,

$$\mathbf{X}_k(i,j) = \begin{pmatrix} x_{i,j} & x_{i,j+1} & \cdots & x_{i,j+k-1} \\ x_{i+1,j} & x_{i+1,j+1} & \cdots & x_{i+1,j+k-1} \\ \vdots & \vdots & \cdots & \vdots \\ x_{i+k-1,j} & x_{i+k-1,j+1} & \cdots & x_{i+k-1,j+k-1} \end{pmatrix}, \tag{2.23}$$

denote the submatrix of $\mathbf{X}$ that starts at row i and column j.

The feature map obtained through the convolution of $\mathbf{X}$ and $\mathbf{W}$,

$$\mathbf{X} * \mathbf{W} = \begin{pmatrix} sum(\mathbf{X}_k(1,1) \odot \mathbf{W}) & \cdots & sum(\mathbf{X}_k(1,n-k+1) \odot \mathbf{W}) \\ sum(\mathbf{X}_k(2,1) \odot \mathbf{W}) & \cdots & sum(\mathbf{X}_k(2,n-k+1) \odot \mathbf{W}) \\ \vdots & \cdots & \vdots \\ sum(\mathbf{X}_k(n-k+1,1) \odot \mathbf{W}) & \cdots & sum(\mathbf{X}_k(n-k+1,n-k+1) \odot \mathbf{W}) \end{pmatrix}, \tag{2.24}$$

corresponds to the $(n-k+1) \times (n-k+1)$ matrix that is obtained by summing entries of the matrix resulting from the Hadamard product of the kernel, $\mathbf{W}$, with each submatrix of $\mathbf{X}$ in the sliding window. The summation function, $sum(\mathbf{A})$, simply sums every entry of the matrix $\mathbf{A}$.

The employment of the convolution function leads to the successive decrease of the data size. The progressive shrinkage of the matrices, leads to the loss of valuable information, especially at the borders, and limits the number of convolutional layers.

To prevent these issues, padding is commonly applied in CNNs. It consists of adding a default number of zeros in each dimension of the input at both sides, before performing the convolution. This way, the output dimesions can be adjusted while the impact of the convolutional operation on the edge pixels is also reduced. Padding enables the preservation of the input size, hence it allows to have arbitrarly deep convolutional layers in a CNN (Zaki & Meira Jr, 2020).

On the other hand, striding is often used to reduce the spatial dimension of the output feature maps, while capturing more high-level features of the input and discarding more localized features. The stride, $s \geq 1$, defines the jumps of the sliding window during convolution (Zaki & Meira Jr, 2020).

The pooling function aims to replace the output of the net at a certain location by the most meaningful summary statistic of its neighbouring outputs. By doing this, the pooling helps to turn the representation invariant to small translations of the input, thus acquiring a wider feature context. There are a number statistics that can be employed, among which the average, the L_2-norm, the weighted-average, or the maximum value are typical options. The latter approach is the most commonly employed and is referred to as *max pooling* (Goodfellow et al., 2016).

The pooling function uses a fixed zero bias, and kernel weights fixed to 1. Neither of these are ever updated in backpropagation, thus the pooling function is maintained throughout the training. Generally, in polling the stride equal to the kernel size. This way, the pooling function is always applied over disjoint

$k \times k$ windows and each convolution output value is only used once by the summary statistics.

In some architectures, there is also useful to upsample the feature map in opposition to the downsampling produced by the pooling layers. These layers are referred to as upsampling layers, or deconvolution layers, and their purpose is to increase the dimensionality of the feature map. The mechanism underlying the upsampling is essentially a convolution with a fractional input stride $1/f$, in which factor f is an integer value. The resulting feature map is calculated by applying a deconvolutional filter to the input. This filter can be either fixed, using a bilinear interpolation, for example, or learned by backpropagation like the kernel in convolution layers (Long, Shelhamer, & Darrell, 2015).

2.4.2 3D Convolutional Neural Networks

3D CNNs are an extension of the traditional (2D) CNNs. They are designed to operate with volumetric data such as 3D medical images, video sequences, and spatio-temporal data. Since in 3D convolutions the input is a 3-dimensional tensor, defined as $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the kernel must also be a 3-dimensional tensor, $\mathbf{W} \in \mathbb{R}^{k_1 \times k_2 \times k_3}$, that slides in the three dimensions of the input (Zaki & Meira Jr, 2020).

In CNNs were the input is a 2D coloured image, the channels correspond to the red-green-blue (RGB) filters. In this particular case, the kernel typically has the same depth as the number of channels: $k_3 = n_3$. Hence, $\mathbf{W} \in \mathbb{R}^{k_1 \times k_2 \times n_3}$, and the feature map obtained is a $(n_1 - k_1 + 1) \times (n_2 - k_2 + 1)$ matrix, similar to equation 2.24.

Most commonly, the third dimension of the input tensor also contains valuable spatial information, such as in 3D images. In these instances, it is essential to use a kernel smaller than the input data in the third dimension. Therefore, if $k_3 \neq n_3$, then the convolution of $\mathbf{X}$ with $\mathbf{W}$ results in a $(n_1 - k_1 + 1) \times (n_2 - k_2 + 1) \times (n_3 - k_3 + 1)$ tensor, whose third dimension simply results from the expansion of the convolution in depth.

Chapter 3

Related Work

In biological image analysis setting, the literature frequently addresses four fundamental tasks: detection, segmentation, classification, and object tracking. The present work explores cutting-edge methods tailored explicitly for cell detection, segmentation, and classification from fluorescent microscopy imaging. This chapter will explore some of the most commonly used state-of-the-art algorithms for each one of these tasks. In addition, complementary algorithms have been recently proposed to predict fluorescent labels from unlabeled cellular images, a process commonly termed as *in silico* staining. Section 3.4 will delve into the comprehensive overview of state-of-the-art techniques in this domain.

At the end of this chapter, Table 3.1 lists the principle networks described in the following sections.

3.1 Image Classification

Image classification stands as a crucial computer vision task where a model undergoes training to categorize images into predefined classes. Recent advancements in this field focus predominantly on enhancing the classifier's performance using standardized databases. These databases not only facilitate direct model performance comparison but also offer a diverse array of images crucial for ensuring model generalization and preventing overfitting. Furthermore, once a network is trained on one of these datasets, the model's adaptability shines as it can be fine-tuned to suit specific applications using smaller annotated datasets, which is particularly invaluable in the biological domain, where the scarcity of annotated data frequently presents a challenge (Moen et al., 2019). Notably, the ImageNet dataset (Russakovsky et al., 2015), also known as the ILSVRC dataset, stands as one of the most renowned collections, comprising over 1.28 million images for training, 50k for validation, and 100k for testing, spanning across a thousand object classes.

The most commonly used architectures for classification tasks are based on very deep convolutional networks (VGG) and residual networks (ResNets) (H. Wang et al., 2018). Simonyan and Zisserman (2014) introduced the VGG networks trained in the ImageNet dataset. VGG networks are characterized by their increased depth compared to previous generalization models, reaching up to 19 layers (VGG16 and VGG19 are extensively documented in the literature), showing remarkable classification accuracies

and broad generalization attributable to the increased representation depth. Their architecture embodies five variable-sized stacks of 3×3 padded convolutional layers followed by a 2×2 max-pooling layer. The CNN is followed by three fully connected layers: the first two have 4096 channels each and 0.5 dropout rate, while the ultimate layer serves as the classification layer, with 1000 output channels.

Meanwhile, the ResNet, pioneered by He et al. (2016), maintained a complexity inferior to the VGG networks while performing in a much larger depth – reaching up to 152 layers. These networks tackled common problems of deep networks, such as the vanishing/exploding gradient problem, in which the weight updates are either too small or too unstable, and the degradation problem, in which the model accuracy saturates and then starts to degrade rapidly. ResNets' underlying principle is using residual blocks with shortcut connections between them. This way, the learning of residual maps is significantly easier to optimize and have greater performance. The general architecture of a ResNet consists of several padded convolutional layers organized into blocks. Between each block, the skip connection allows the input of that block to be added element-wise to its output, creating a shortcut for the gradient to flow during training. Following the sequence of the residual blocks, a fully connected layer maps the result to 1000 output channels corresponding to the classes of the ImageNet dataset.

A concrete application of classification to biological data analysis is cell phenotyping. A network designed for cell phenotyping can be trained to either directly predict the phenotype, in which the network's last layer corresponds to a classification layer, or extract a feature vector, in which the autoencoder can extract the most meaningful features from the input images. These features can then be used to either supervisedly assign a phenotype from a set of classes known *a priori* or unsurpervisedly cluster the observations. The description of phenotypes with unsupervised clustering allows the discovery of novel classes and surpasses the need for manually annotated datasets (Pratapa et al., 2021).

Many authors have attemped to develop DL tools for this classification problem using different approaches. For instance, Godinez et al. (2017) developed a multi-scale CNN (M-CNN) that considers the input image at different spatial scales in parallel and assigns a probability to each possible phenotype class. However, a model like this is limited to the classes defined *a priori* and cannot detect novel phenotypes. To address this limitation, Sommer et al. (2017) developed the *CellCognition Explorer* tool to extract feature vectors with an autoencoder and infer the detection of novel phenotypes from nuclear morphology screening data with unsupervised clustering.

Alternatively, Pawlowski et al. (2016) used state-of-the-art algorithms, such as Inception-v3, ResNet-152, ResNet-101, and VGG16, pre-trained on the ImageNet dataset to extract features and perform morphological profiling of fluorescence microscopy images of cultures human cells without additional fine-tuning. Instead of using unsupervised learning, the target mechanisms of action were benchmarked by averaging the features of each class, and the observations were then classified using the 1-nearest neighbour classifier.

Despite VGG and ResNet networks being specifically designed for image classification, adaptations of these networks are also being used for other applications. Namely, autoencoder features of their CNN layers are notably beneficial to perform object detection and segmentation tasks through the extraction of features, as we will explore deeper in sections 3.2 and 3.3.

3.2 Object Detection

The task of object detection involves pinpointing objects within an image using bounding boxes and assigning corresponding labels. In images with multiple objects, detection is a crucial step preceding classification. As each instance can only be associated with one label, separate classification of the multiple objects within the image is necessary for accurate analysis (Gupta et al., 2019).

Girshick et al. (2014) introduced the concept of combining a region proposal algorithm with CNNs to devise R-CNNs for object detection. This approach involves three distinct modules: the initial module generates region proposals utilizing a selective search algorithm, followed by a large CNN to extract feature vectors, and finally, a module for object classification using class-specific linear SVMs. A notable limitation of this network is its relatively slow process, requiring the generation of approximately 2000 category-independent region proposals for each input image. Additionally, due to its multi-staged nature, it cannot be optimized end-to-end. Building upon R-CNNs, Ren et al. (2015) proposed a more efficient variant known as Faster R-CNN, consisting of two networks. The first network, a fully convolutional region proposal network (RPN), produces object proposals with corresponding objectness scores, measuring the likelihood of a region belonging to specific object classes rather than the background. This RPN utilizes a VGG16 as a feature extractor alongside two sibling fully connected layers for generating objectness scores and bounding boxes, respectively. The second network, a Fast R-CNN (Girshick, 2015), utilizes a fixed-length feature vector obtained through a region of interest (ROI) pooling layer for object detection within the proposed regions. Faster R-CNN significantly improves efficiency compared to the traditional R-CNN and allows for end-to-end training.

A different approach to this task is to frame object detection as a regression problem. Redmon et al. (2016) introduced You Only Look Once (YOLO) networks as a unified solution for real-time object detection by predicting bounding boxes and class probabilities simultaneously and directly from full images. YOLO divides the input images into a grid of size S and for each grid cell predicts B bounding boxes and a respective confidence score – $p(\textit{object}) \times \text{IoU}^{\textit{truth}}_{\textit{pred}}$ –, in which $p(\textit{object})$ formally defines the probability of the bounding box containing a given object, and the class-conditional probabilities – $p(c_k|\textit{object})$ – by looking at the entire image and not just the regions of interest. The class-specific confidence scores for each box are obtained by multiplying these identities. The value per grid cell is then thresholded to predict the final bounding box results. The YOLO architectures have undergone multiple transformations over the years, resulting in significant improvements in performance by balancing its speed and accuracy (Terven & Cordova-Esparza, 2023).

Object detection architectures are not limited to R-CNN and YOLO based-models (Zaidi et al., 2022). However, these are considered the most popular approaches, attributable to their stable performance and coherent results across multiple validated datasets (Meijering, 2020; H. Wang et al., 2018).

In the biological setting, the cell detection task is essential for characterizing samples through cell counting, phenotyping, and process automation through automated positioning of cells (Sadak et al., 2020). However, the automatic detection of astrocytes, in particular, is often challenged by their complex morphology and versatile branching structure (Suleymanova et al., 2018). Huang et al. (2022)

used YOLO, specifically YOLOv5, to perform the automated detection of GFAP-immunolabeled astrocytes in brightfield and fluorescent microscope images. Besides providing an accurate framework for the quantitative analysis of astrocytes, this architecture could also be applied to characterize their spatial distribution and extract morphological features that can be used to cluster astrocytes based on their phenotypic characteriztics. The architecture of YOLOv5 comprises three consecutive sections: backbone, neck, and head. The backbone is designed to extract variable-sized feature maps from multiple convolution and pooling layers, fused in the neck section to enhance information flow and capture patterns at different spatial scales in the image. Lastly, the head section generates the output results: the bounding box for the detected objects, detection scores, and class probabilities. In this paper, the astrocytes were all classified to the same class. However, it is also possible to distinguish different astrocyte morphologies using this tool and a well-annotated dataset.

YOLOv5 has proved to be a competitive alternative to previous astrocyte detection networks, such as FindMyCell (Suleymanova et al., 2018) and GESU-Net (Kayasandik et al., 2020), especially in dense cell population images. More recently, YOLOv8 (Jocher et al., 2023) has been released with an architecture similar to YOLOv5 and supporting multiple vision tasks, such as object detection, segmentation, pose estimation, tracking, and classification (Terven & Cordova-Esparza, 2023). YOLOv8 has already been successfully validated for the classification of white blood cells (Nugraha & Erfianto, 2023), which substantiates its applicability for astrocyte detection. However, the need for annotated astrocyte images is still the main factor hindering the development of more accurate algorithms. This problem leads to nonoptimal recall and precision values in several datasets due to the missed and incorrectly identified astrocytes, respectively.

3.3 Image Segmentation

Within computer vision, image segmentation entails pixel-wise classification to discern individual objects from the background (H. Wang et al., 2018). Moreover, image segmentation branches into semantic and instance segmentation tasks. Semantic segmentation generally differentiates between background and foreground, assigning a binary class to each pixel. On the other hand, instance segmentation goes one step further and delineates each distinct instance appearing in the image (Leyendecker et al., 2022). Both semantic and instance segmentation tasks can be cast as a classification problem in which fully convolutional neural networks show remarkable performance (Yin et al., 2022).

The DeepLab and SegNet networks are based on the feature representation provided by the classification architectures ResNet-101 and VGG16, respectively. These networks possess a fully convolutional neural architecture and are designed for the *semantic segmentation* task. The DeepLab, pioneered by Chen et al. (2017), proposed the incorporation of atrous convolutions, atrous spatial pyramid pooling (ASPP), and conditional random fields (CRFs) to improve the quality of the segmentation output. To increase feature resolution, the authors employed atrous convolutions to the last convolutional layers by replacing the pooling layers with upsampling filters and subsequently applying a bilinear interpolation to recover feature maps with the same resolution as the input image. On the other hand, ASPP was

employed by resampling the feature layers at different scales, thus allowing the capture of objects at multiple scales. In order to improve the localization accuracy, the CRFs were employed to capture fine edge details by considering the surrounding pixel labels (Y. Wang et al., 2022). In this paper, one ResNet and one VGG-based variant were proposed. By benchmarking with the PASCAL VOC 2012 dataset, the residual network variant achieved better semantic segmentation and state-of-the-art results.

Conversely, the development of SegNet (Badrinarayanan et al., 2017) was predominantly motivated by road scene applications. This network demonstrates a unique capability to segment objects at variable sizes while exhibiting more efficient memory usage during inference compared to other architectures like DeepLab and DeconvNet. In stark contrast to DeepLab, SegNet functions by independently producing the probability of each pixel class, regardless of its neighbouring pixels. Although the validation of this network was confined to road scene datasets, it showcased competitive performance against other state-of-the-art algorithms. Notably, beyond its intended road scene application, some researchers have successfully applied the SegNet architecture in the biological domain for tasks, such as cell segmentation, achieving satisfactory performance (Daniel et al., 2022; Tran et al., 2018).

On the other hand, the U-Net was initially conceived for biomedical image *instance segmentation*. Pioneered by Ronneberger et al. (2015), it diverges from typical convolutional networks by embracing a two-dimensional fully convolutional architecture. Its key advantages include efficient training due to fewer parameters and the ability to achieve accurate segmentation with a smaller set of training images. The traditional U-Net consists of an encoder and a decoder. The encoder is a typical convolutional network comprising multiple sequences of two 3×3 unpadded convolutional layers, ReLU activation, and 2×2 polling layers that double the feature channels. In the decoder, pooling layers are substituted with upsampling layers, creating a roughly symmetric architecture to the encoder, forming the U-shaped structure. Furthermore, the decoder also receives information from lower layers, which is essential for the reconstruction due to the loss of information derived from the unpadded convolutions. The application of successive convolutional layers contributes to the model's accuracy by detecting fine features in the input (Yin et al., 2022).

Since the U-Net was first introduced, numerous authors have proposed additional networks based on its architecture for biomedical image segmentation, such as Stardist and Cellpose. Stardist, proposed by Schmidt et al. (2018), was designed to increase the accuracy of the previous models by approximating the bounding box surrounding each cell instance to a star-convex polygon. Star-convex polygons adjust much better to the cell or nucleus shape than bounding boxes and, therefore, are less prone to segmentation errors, such as falsely merging bordering cells or suppressing valid cell instances, especially in densely-packed cell images. Stardist separately predicts the probability of each pixel belonging to an object and its distance to the object boundary, thus predicting a star-convex polygon for each pixel. In addition, the Stardist algorithm was extended for nuclei segmentation in 3D fluorescence microscopy datasets by predicting star-convex polyhedra with Stardist 3D (Weigert et al., 2020). This algorithm uses a modified 3D variant of ResNet to predict the radial distances and the object probabilities instead of a U-Net, outperforming state-of-the-art algorithms, such as watershed and U-Net 3D (Çiçek et al., 2016).

Alternatively, Cellpose, developed by Stringer et al. (2021), successfully generates topological maps

to segment cell nuclei or cytoplasm instances. In this model, the U-Net standard convolutional building blocks are replaced by residual blocks, which significantly improved the network performance and substantially outperformed Stardist on the tested datasets. The original model was trained primarily on DAPI-annotated fluorescent microscopy images. However, given the diversity of dataset annotation styles, a second version of Cellpose was already developed to allow fine-tuning the pre-trained model (Pachitariu & Stringer, 2022). Similarly to Stardist, Cellpose also has an extension for the robust cell segmentation of 3D microscopy image data with Cellpose 3D (Eschweiler et al., 2022), which also uses a U-Net as the backbone structure of the architecture.

Mask R-CNN (He et al., 2017) takes a different approach from Stardist and Cellpose. Instead of directly predicting a shape that fits the detected object, it starts by determining the objects' bounding boxes by employing the Faster R-CNN algorithm. The network reports a parallel prediction of the binary mask and the class label in the output layer. This algorithm was not explicitly designed for biomedical applications and has a broader set of applications. Therefore, despite still being used for cell segmentation tasks (Waibel et al., 2021), some authors suggest it is not as competitive as other alternative algorithms, such as Stardist, in specific data domains (Schmidt et al., 2018).

More recently, self-attention transformers are also being used to aggregate associative features among patch images enhancing the networks performance (Sugimoto et al., 2022). UTNet (Gao et al., 2021) and U-Net Transformer (Petit et al., 2021) are two U-net-based networks that employ self-attention to enhance medical image segmentation. In addition to image segmentation, self-attention transformers have also been used for multi-class cell detection (Sugimoto et al., 2022).

3.4 *In Silico* Staining

In silico staining, also referred to as *in silico* labeling, represents a pixel-wise regression task that has proven successful in predicting fluorescent markers from TL microscopy images (Waibel et al., 2019). While fluorescent microscopy, as discussed in section 2.1, offers notable advantages, its reliance on sophisticated instrumentation, which is not as easily acessible as TL microscopes, limitations in simultaneous labeling due to spectral overlap, and potential harm to biological samples through fixation or phototoxicity, remain significant concerns (Christiansen et al., 2018; Ounkomol et al., 2018). However, *in silico* staining effectively mitigates these drawbacks, transcending the limitations associated with traditional fluorescent methods, enabling cost-effective and consistent labeling without compromising the integrity of biological samples.

Christiansen et al. (2018) implemented a U-Net-based architecture to train pairs of TL z-stack images, collected with brightfield, phase-contrast and DIC methods, and fluorescently labeled images to predict the *in silico* staining. In their experiments, the network showed a high accuracy in fluorescence prediction, with a Pearson correlation of 0.87 or higher between the true and predicted pixel intensities in cell nuclei. Furthermore, the trained network could also be used in transfer learning settings by learning a new set of labels from a small amount of additional training data. Besides the cell nuclei prediction, the network was also efficiently used to predict the cell type and subcellular structure.

Alternatively, Atwell et al. (2023), also from the Helmholtz Zentrum München, developed an AI imaging tool, Bright2Nuc, to predict cell nuclei staining of 3D stem cell cultures from z-stack brightfield images acquired with confocal microscopy. Bright2Nuc is a U-Net-based network and works similarly to the network proposed by Christiansen et al. (2018). In addition, the authors also developed a pipeline capable of segmenting the nuclei, predicting the differentiation state, and tracking the single cells from the *in silico* labeled images.

Moreover, recent efforts have been devoted to creating versatile algorithms that can perform multiple computer vision tasks to enhance the analysis of biological images. In particular, Bright2Nuc DL framework was based on the InstantDL pipeline (Waibel et al., 2019) developed for biomedical image segmentation and classification. InstantDL is a highly versatile tool designed for multiple image processing tasks and it is benchmarked on state-of-the-art DL algorithms. More precisely, it enables semantic segmentation and pixel-wise regression tasks on 2D and 3D input images using a U-Net architecture, instance segmentation tasks on 2D input images using the Mask R-CNN architecture, and classification tasks in 2D input images using a ResNet-50.

Table 3.1: Summary of the most common deep neural networks used for computer vision tasks in biological image analysis.

Neural Network	Tasks	Dimensionality	Evaluation metrics	Breakthroughs/ Innovations	Citation
VGG	• classification • localization	2D	• top-1 error • top-5 error	Significant improvement on the prior state-of-the-art configurations. Increased depth, with 16 to 19 layers.	(Simonyan & Zisserman, 2014)
ResNet (based on VGG)	• classification	2D	• top-1 error • top-5 error	Easier to optimize. Higher accuracy. Increased depth (up to 8× deeper than VGG nets).	(He et al., 2016)
R-CNN	• object detection	2D	• mean average precision (AP)	Combination of RPNs with CNNs. Small amount of annotated data.	(Girshick et al., 2014)
Faster R-CNN (based on VGG16 and R-CNN)	• object detection	2D	• top-5 error	Higher efficiency compared to R-CNN. End-to-end optimization.	(Ren et al., 2015)
YOLO	• object detection	2D (also 3D in later versions)	• mean AP • IoU	Real-time prediction. End-to-end optimization.	(Redmon et al., 2016)
DeepLab (based on the ResNet)	• semantic segmentation	2D	• mean IoU (Jaccard index)	Convolution with upsampled filters, atrous spatial pyramid pooling, and combination of deep CNNs with probabilistic graphical models.	(Chen et al., 2017)
SegNet (based on the VGG16)	• semantic segmentation	2D	• accuracy • mean IoU • F1-score	More efficient memory usage during inference.	(Badrinarayanan et al., 2017)
U-Net (Fully-convolutional network)	• semantic segmentation • instance segmentation	2D (also 3D in later versions)	• IoU • wrapping error • rand error • pixel error	Increased learning capacity from few training images. Yields more precise segmentations.	(Ronneberger et al., 2015)
StarDist (based on the U-Net)	• instance segmentation	2D (also 3D in later versions)	• IoU • MAE	Cell nuclei localization via star-convex polygons (in opposition to bounding boxes)	(Schmidt et al., 2018; Weigert et al., 2020)
Cellpose (based on the U-Net)	• instance segmentation	2D and 3D	• IOU • mean AP • mean IoU	Learning from a wide-variety of microscopy images. Bypasses retraining or parameter adjustments.	(Eschweiler et al., 2022; Stringer et al., 2021)
Mask R-CNN (based on the Faster R-CNN)	• instance segmentation	2D	• mean AP	Bounding box prediction followed by object segmentation.	(He et al., 2017)
Bright2Nuc (based on the U-Net)	• pixe-wise regression	3D	• average Pearson correlation	Integrated pipeline to infer *in silico* labeling from brightfield microscopy images, and conduct nuclei segmentation and phenotype classification.	(Atwell et al., 2023)
InstantDL (based on U-Net, Mask-RCNN, ResNet50)	• classification • instance segmentation • semantic segmentation • pixel-wise regression	2D and 3D	• mean IoU • AUC • median pixel-wise Pearson correlation	Integrated pipeline to perform multiple computer vision tasks.	(Waibel et al., 2019)

Chapter 4

Data Profiling

The astrocyte cultures under study were obtained from murine cortical gray matter from P5-6 posnatal mice. The astrocytes were cultured for 7 days *in vitro* (DIV) on tissue culture T-flasks. Afterwards, the astrocytes were embedded and cultured in sodium alginate hydrogels carrying an arginine-glycine-aspartate (RGD) sequence for integrins attachment (Bellis, 2011).

The dataset consists of 21 three dimensional 8 bit images obtained with multiphoton microscopy. The images were obtained in z-stacks of 1024×1024 pixels, corresponding to 591.05×591.05 µm ($n = 16$), 587.22×587.22 µm ($n = 3$), 582.47×582.47 µm ($n = 1$) or 584.29×584.29 µm ($n = 1$), with variable depths, ranging between 39 ($n = 2$), 47 ($n = 1$), 64 ($n = 17$), and 85 ($n = 1$) slices, equivalent to 78, 94, 128, and 170 µm, respectively. Moreover, each z-stack has two channels: one corresponding to the GFAP marker, and the other to the DAPI marker wavelengths. Thus, the images have four dimensions corresponding to the width ($x = 1024$ pixels), height ($y = 1024$ pixels), depth (z), and channels ($c = \{1, 2\}$).

The images' quality is affeted by the intrinsic acquisition methodologies. Therefore, in several captures, the images quality can be deprecated by the background noise (Figure 4.1a), microspope artifacts (Figure 4.1b) and gel artifacts (Figure 4.1c) which difficults their visual assessment. Furthermore, this deteoration is not comparable between images, which might compromise the annotation and subsequently the learning processes.

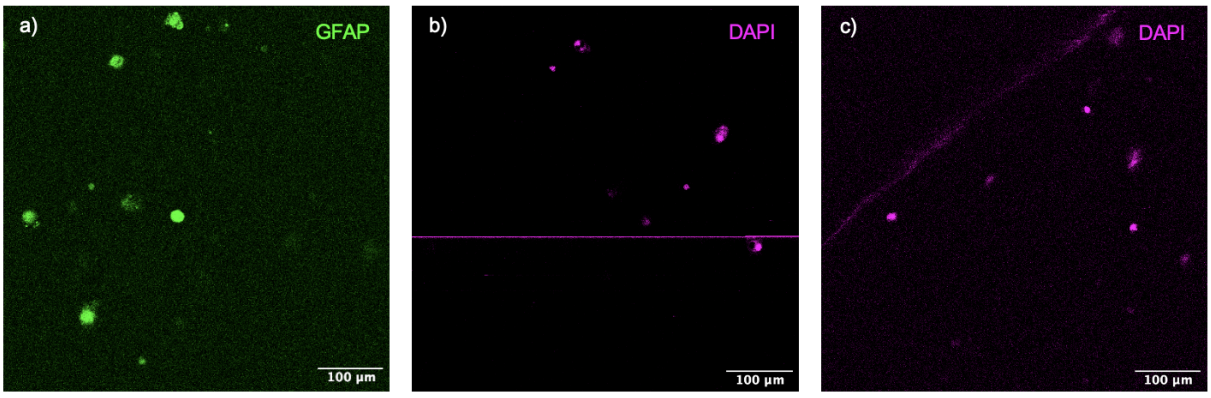

Figure 4.1: Artifacts detected in the dataset images. (a) Visualization of background noise (image 17, $z = 1$); (b) Visualization of microscope artifacts (image 21, $z = 5$); (c) Visualization of gel artifacts (image 10, $z = 1$).

4.1 Image Preprocessing using Fiji

The images underwent preprocessing in Fiji (Schindelin et al., 2012) via an automated pipeline designed to diminish background noise and fine-tune contrast levels (Figure 4.2a). This preprocessing pipeline was scripted in Python, using version 2.14.0 of the Fiji python package, pyimagej (Rueden et al., 2022). Preprocessing serves as a crucial step in preparing the images for manual annotation, which is essential for training our models effectively. However, our end goal is to enable advanced deep learning models to independently extract valuable features straight from the raw images. Ultimately, we aim to empower these models to discern and utilize critical patterns and information without relying on preprocessing steps.

The initial steps of the preprocessing pipeline involve spliting the color channels and normalizing the histograms. This process effectively doubles the original dataset, creating two distinct sets of images, each corresponding to a specific wavelength. Meanwhile, considering the inherent limitations of depth imaging in 3D cultures, an analysis to address the diminishing fluorescent signal in function of the depth was conducted. Atwell et al. (2023) undertook a similar assessment of wavelength-dependent intensity decay concerning imaging depth, by computing the relative average fluorescence intensity (RAFI) across varying depths. By assuming that the average image intensity should remain consistent with imaging depth, they derived the imaging-depth decay rate using linear regression and subsequently applied the necessary correction.

In this project, the RAFI was determined by comparing the average signal intensity (measured in the 8 bit scale) at a particular depth with the minimum average signal intensity of the cell culture. The findings indicated an absence of a significant decay trend ilustrated by the overall low variance of slices mean intensity per stack (Figure 4.2b and Figure 4.2c). Overall, the mean RAFI of the DAPI channel was 1.218 ± 0.049 and the mean RAFI of the GFAP channel was 1.353 ± 0.268. These results lead to the conclusion that no adjustments were required for the signal intensity.

Optimal contrast is achieved through histogram analysis of each z-stack, wherein minimum and maximum pixel values for the images are set. Through the global analysis of the dataset histogram, it is noticible that the data is shifted towards lower intensities and high intensity pixels are less frequent (Figure 4.2d and Figure 4.2e). The assessment of individual image histograms revealed heterogeneity across the dataset. Therefore, for accurate analysis, the contrast adjustment has to be performed manually for each image. It is important to note that contrast adjustments do not necessarily alter pixel values but optimize the visual spectrum (Figure 4.2f and Figure 4.2g). Consequently, this step is not mandatory for the preprocessing phase and was reserved solely for image annotations.

To suppress high frequencies in the image and minimizing spatial spread, there were also applied filters to the image. In a trial preprocessing approach there were only tested the mean and the bidimensional Gaussian filters, with radius, r, or standard deviation, σ, equal to 1 pixel, respectively (Figures 4.2h and 4.2i). Changing the filters' parameters can also serve as a potential data augmentation strategy to expand the training dataset for further analysis. However, this procedure must be approached cautiously to prevent blurring of low-intensity nuclei.

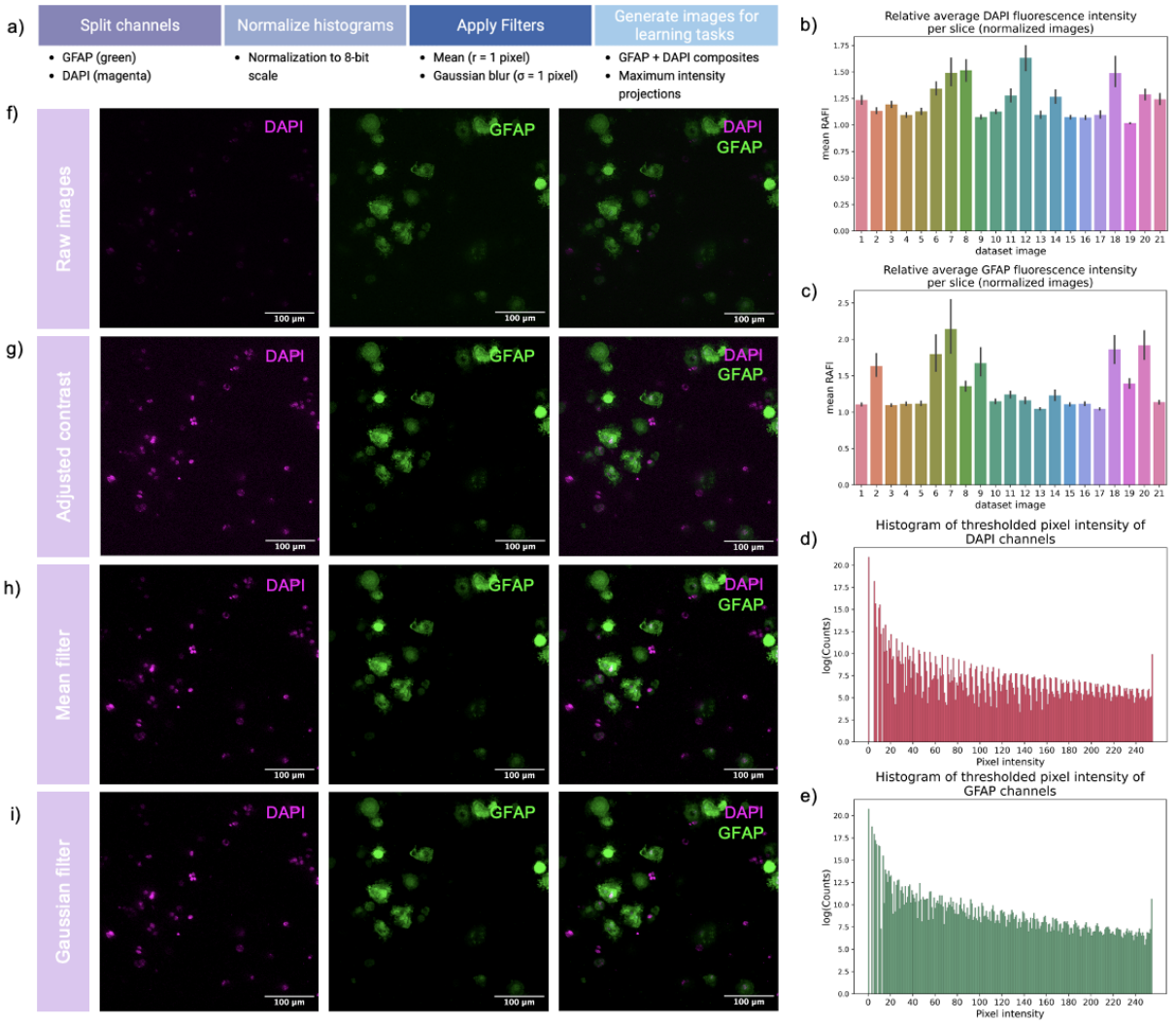

Figure 4.2: Fiji preprocessing. (a) Fiji preprocessing pipeline to adjust contrast, apply filters, and obtain final input images for each learning task; (b) Mean of the relative average DAPI fluorescence intensity calculated per slice in normalized images; (c) Mean of the relative average GFAP fluorescence intensity calculated per slice in normalized images; (d) Histogram of the logarithm of the sum of all images pixels in DAPI channel; (e) Histogram of the logarithm of the sum of all images pixels in GFAP channel; (f) Raw maximum intensity projections of image 2; (g) Raw maximum intensity projections of image 2 with adjusted contrast; (h) Maximum intensity projection after the application of the mean filter ($r = 1$ pixel) to image 2; (i) Maximum intensity projection after the application of the gaussian filter ($\sigma = 1$ pixel) to image 2.

Depending on the learning task that is being addressed, some configurations of input training images might be more suitable than others. To prepare the training datasets for the proposed tasks, four different datasets were bluit: two for 2D and two for 3D analysis (Table 4.1). The tridimensional images encompass one z-stack of only the DAPI channel (*3d nuclei* dataset), and one composite of the GFAP and DAPI channels (*3d composite* dataset). The *3d nuclei* dataset will be used to assess nuclei specific characteristics, such as cell nuclei count and cell nuclei volume. On the other hand, the *3d composite* dataset will be useful for quantifying astrocyte related characteristics such as astrocyte count, astrocyte type classification, multinucleated astrocyte state classification, and attemptively the astrocyte territory. For the 2D analysis, there were performed maximum intensity projections of the 3D datasets. Every resulting preprocessed image was manually validated to guarantee the correct preprocessing.

Table 4.1: Annotation and description of the secondary datasets obtained from the original dataset through channel selection and maximum intensity projection manipulations.

Dataset	Annotation	Description
2d nuclei	Nuclei instance and semantic segmentation using Fiji free hand selection tool.	Manually annotated maximum intensity projection of the DAPI channel for segmentation tasks.
3d nuclei	Nuclei instance and semantic segmentation using Fiji free hand selection tool.	Manually annotated z-stacks of the DAPI channel for 3D segmentation tasks.
2d composite	Classification labels regarding the type of cell (astrocyte or not), astrocyte type, and the multinucleated state.	Manually annotated maximum intensity projection of composite images of DAPI and GFAP channels.
3d composite	Classification labels regarding the type of cell (astrocyte or not), astrocyte type, and the multinucleated state.	Manually annotated z-stack composites images of DAPI and GFAP channels.

4.2 Image Annotation and Characterization

To produce the annotations required for the learning tasks, the dataset needs to be manually annotated with the nuclei positions and respective classifications. This task was divided into four steps: (1) instance segmentation of nuclei in the *2d nuclei* dataset; (2) instance segmentation of nuclei in the *3d nuclei* dataset; (3) identification of astrocytes and classification of the astrocyte type; and (4) identification of multinucleated astrocytes.

4.2.1 Segmentation

The nuclei segmentation was performed manually in the maximum intensity projection images of the DAPI channel, the *2d nuclei* dataset (Figure 4.3a). The segmentation was performed using the Fiji's free hand selection tool (Figure 4.3b), after contrast adjustement in the maximum intensity projections. The number of masks obtained are equivalent to the number of nuclei present in each image, and therefore encompass a straightforward method to count the number of cells in each image.

The masks obtained are fed to the learning networks to compare its preditions to the groundtruth and fine-tune the learning model as well as to assess the models' performance on a testing set. After manually segmenting the nuclei, a semantic segmentation mask (Figure 4.3c) can be accessed, as well as individual instance segmentation masks (Figure 4.3d).

Given the resulting masks, consider the following statistics pertaining to the nuclei charateristics across images. Over the 21 images, a total of 2034 nuclei were identified, giving an average of approximately 96.86 nuclei per image. However, the standard deviation is quite high, with a value of 42.12, which ilustrates the high heterogeneity of cell density across the images. Furthermore, median number of nuclei per image is 84 supporting that most images have lower cell counts, however there are a few that deviate substantially from the mean. A candidate outlier, image 13, was also identified, presenting a cell count of 194 (Figure 4.3e).

On the other hand, the segmented nuclei area as an approximately stable median value arround 69.05 µm^2. However, there could be identified several outliers across the images (Figure 4.3f). This can

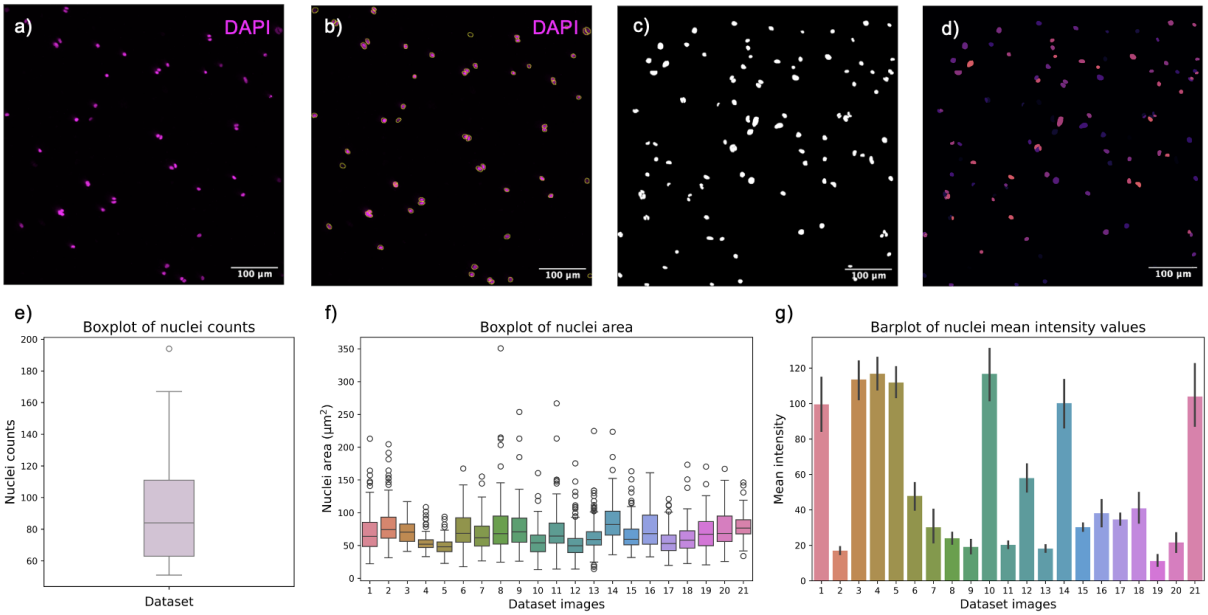

Figure 4.3: 2D nuclei profiling and statistics. (a) Maximum intensity projection of image 11 from the *2d nuclei* dataset; (b) Manual segmentation of nuclei in Fiji using the hand selection tool; (c) Semantic segmentation resulting from the manual segmentation in Fiji; (d) Instance segmentation resulting from the manual segmentation in Fiji; (e) Boxplot of nuclei counts over the entire dataset; (f) Boxplot of nuclei area, in µm^2, for each image in the dataset; (g) Barplot of nuclei mean intensity values, using the mean as estimator.

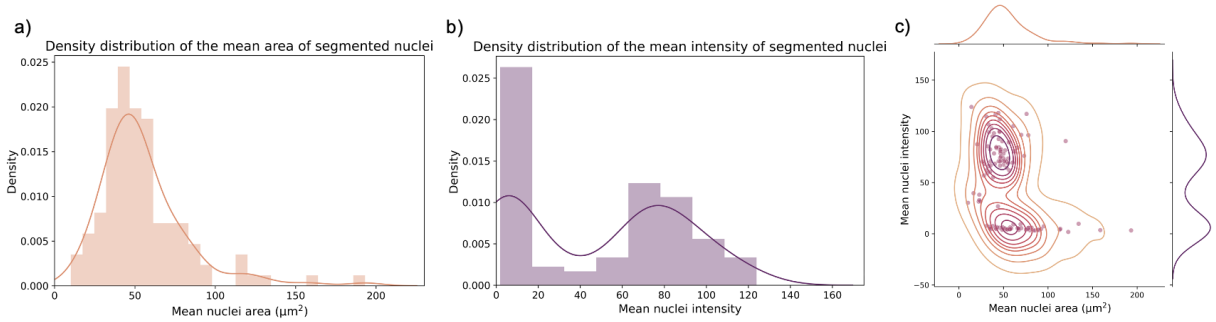

Figure 4.4: 3D nuclei profiling and statistics. (a) Density distribution of the mean area of the segmented nuclei in the z-stack of image 1; (b) Density distribution of the mean intensity of the segmented nuclei in the z-stack of image 1; (c) Bivariate distribution of the mean area and the mean intensity of the segmented nuclei in the z-stack of image 1.

be caused by the misclassification of the segmentation masks resulting from the difficulty of identifying the nuclei borders in some instances. Analysing the mean intensity values of the segmented nuclei of each image, its perceptible the high contrast heterogeinity across images (Figure 4.3g).

There was also performed a segmentation of the nuclei in z-stack of image 1. While in the maximum intensity projection there had been identified 111 nuclei, in the z-stack there were segmented 117 nuclei. The additional nuclei detected in the z-stack were hidden by other cells in the maximum intensity projection or were coupled in a single segmentation due to the difficulty of distinguishing the multiple instances on 2D projections. On the other hand, the mean intensity of the nuclei decreased from 99.53 in the maximum intensity projection to 48.52 in the z-stack. This is mainly due to the fact that the segmentations in the z-stack are performed over multiple slices, crossing the varying diameters of the nuclei sections.

Accordingly, the mean nuclei area also decreased from 70.89 µm^2 to 55.13 µm^2.

Furthermore, the density distribution of the mean area of the segmented nuclei resembles a Normal distribution (Figure 4.4a). Whereas, the analysis of the density distribution of the mean intensity of segmented nuclei reveals two peaks (Figure 4.4b). The left peak corresponds to the very low intensity nuclei. However, upon analysing the relation between both variables, we concluded they were weakly correlated (Figure 4.4c), with a Pearson correlation of -0.38.

4.2.2 Classification

The manual classification tasks are performed on the composite datasets (Figure 4.5a). Even tough there were created a 2D and a 3D version of this datasets, for this classification the *3d composite* dataset is more suitable since it allows depth discrimination and therefore is less susceptible to misclassification errors.

For the identification of astrocytes and classification of the astrocyte type, the nuclei are classified into one of three classes: *1* – protoplasmic-like astrocyte, *2* – fibrous-like astrocyte, or *3* – not an astrocyte (Figure 4.5b). For the ilustration purposes of this report, the annotation was made purely with points using the Fiji's multi-point tool. However, for the training of the deep learning networks, the annotation as to be adapted to the requirements of the chosen models and typically is inputted into the network in a .csv like format.

On the other hand, the multinucleated state classification is specific to astrocytes. Therefore, only nuclei labeled as astrocytes will be classified as multinucleated or not. This way, this can be understood as a binary classification task, in which label *0* corresponds to nucleus of mononucleated astrocytes and label *1* to nucleus of multinucleated astrocytes (Figure 4.5c).

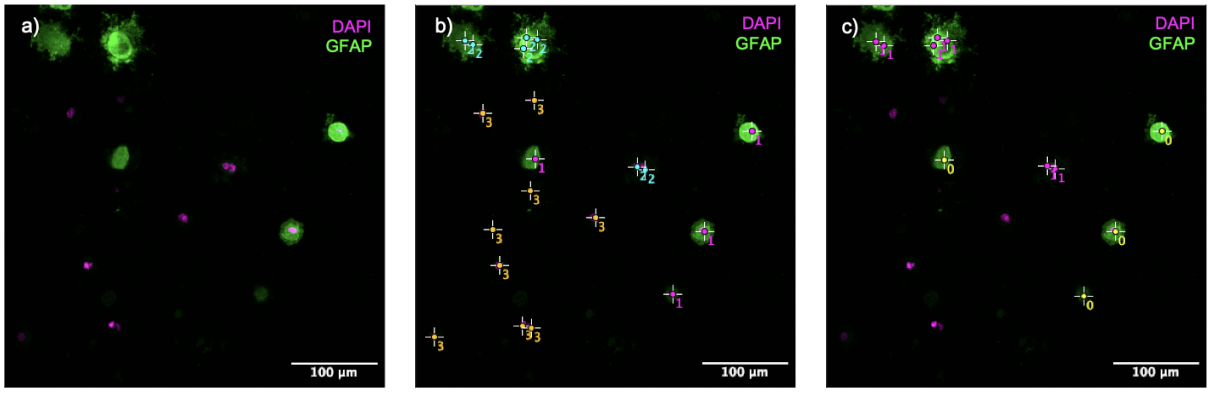

Figure 4.5: Maximum intensity projection of the labeling of image 6 ($z = [1, 24]$). (a) Unlabeled image; (b) Classification of the astrocyte type into *1* – protoplasmic-like, *2* – fibrous-like, and *3* – not an astrocyte; (c) Classification of the astrocyte's multinucleated state into *0* – nucleus of mononucleated astrocyte and *1* – nucleus of multinucleated astrocyte.

Chapter 5

Solution Proposal

The objective of the Master's Thesis project is to develop a comprehensive deep learning pipeline for the characterization of 3D astrocyte cultures, with potential to characterize astrocyte-neuron co-cultures as well. The project aims to achieve several computational objectives, including the quantification of cell nuclei, the identification and quantification of astrocytes, the classification of astrocytes into protoplasmic-like or fibrous-like types, the classification of astrocytes into mononuclear or multinuclear cells, and attempting the characterization of astrocyte territories within the cell cultures.

The computational approach involves leveraging established state-of-the-art networks to characterize the cell cultures and assess the feasibility of fine-tuning the models with a limited dataset. Specifically, InstantDL will be used for nuclei segmentation and quantification, while YOLOv8 will be employed to identify and classify nuclei corresponding to astrocytes based on type and multinucleation state. Furthermore, an attempt to fine-tune SynEM (Staffler, 2019) to delineate astrocyte territories within the cultures will be made, and *in silico* staining will also pottentially be addressed resorting to Bright2Nuc.

Concurrently, wet lab experiments will involve the cultivation of murine-derived astrocytes in diverse 3D hydrogels. The overarching goal is to employ the developed pipeline to characterize each gel culture condition. Through this analysis, the project aims to evaluate the impact of varying culture conditions on cell and culture phenotype. Additionally, optimization of cell density and imaging parameters, such as z-stack resolution using confocal microscopy, will be explored based on insights gained from computational analysis.

5.1 Nuclei Segmentation and Quantification

To segment and count nuclei instances in each image, the InstantDL pipeline employs the Mask R-CNN network. This network takes 2D images as input and outputs binary masks for each nuclei segmentation, allowing a straightforward nuclei quantification.

The architecture's performance will be assessed by employing different learnt weights to the network. Firstly, we will test its performance with pre-trained weights provided by Waibel et al. (2021). This model was trained using data from the 2D nuclear instance segmentation of the nuclear detection

challenge dataset (DSB2018) (Caicedo et al., 2019). Secondly, we will attempt to improve the network's performance by fine-tuning the model to our dataset. Given the limited dataset, it is essential to calibrate the ratio between training and validation sets, as well as the number of folds for cross-validation to balance statistical and efficiency requirements. As such, there are going to be performed two k-fold cross-validations with $k = 3$ and $k = 7$.

Furthermore, the impact of data augmentation on the model's performance will also be assessed. InstantDL is already equipped with several data augmentation strategies such as: feature scaling, standard normalization, horizontal and vertical flips, Poisson noise, rotations, amplification, contrast and brightness adjustment, gamma correction, thresholding and Gaussian blurring. The data augmentation strategies are incorporated in the training by randomly modifying the training and validation images at predefined rates. Thus, the training subset is solely augmented indirectly: the model is fed with the same subsets as without data augmentation. The performance of the Mask R-CNN architecture will be evaluated using five distinct models in addition to the generalized ImageNet model, as outlined in Table 5.1.

Table 5.1: Segmentation Mask R-CNN models.

Model	**Cross Validation**	**Data Augmentation**	**Description**
dsb2018	–	–	Model trained with the DSB2018 dataset.
ftk3	3-fold	No	Models obtained by fine-tuning the model trained with DSB2018 dataset to our dataset.
ftk7	7-fold	No	
aftk3	3-fold	Yes	Models obtained by fine-tuning the model trained with DSB2018 dataset to our dataset after data augmentation.
aftk7	7-fold	Yes	

InstantDL does not provide an architecture tailored to perform 3D instance nuclei segmentation. For this scenario, InstantDL pipeline will be extended with StarDist 3D segmentation algorithm. Similarly to the training process of Mask R-CNN models, we will attempt to fine-tune the 3D weights model provided by Weigert et al. (2020) to our dataset and compare its performance to the generalized model.

For both segmentation tasks (2D and 3D instance segmentation), the data has to be manually annotated as described in section 4.2.1. Although the segmentation errors have been minimized using a standardized data preprocessing pipeline, data annotation is highly susceptible to variations and subjective to the annotator. For this matter, cross-expert validation and concordance analysis can be conducted to ensure accurate annotations.

To evaluate the models' performance the segmentation predictions will be compared to the ground truth masks by measuring IoU coefficient and the ratio of segmented nuclei in the prediction and the ground truth. This quantities must be thoughtfully analyzed and interpreted to assess the models' performance accurately.

The true positive (TP), false negative (FN), and false positive (FP) nuclei segmentations are good indicators of how the predictions related to the expected results. To compute these quantities, we need

to be aware of null expectations when interpreting the results of metrics. Illustrating, there were only considered segmentations which the IoU surpassed 20%. This way, the TP nuclei segmentations correspond to the number of masks that have at least one prediction whose IoU is above 20%. Contrastingly, the FN segmentations correspond to the number of ground truth masks that are not associated to any prediction, and the FP segmentations to the number of predictions that are not associated to any ground truth mask. Specifically, the relation between the expected and the predicted results can be computed with the true positive ratio (TPR),

$$TPR = \frac{TP}{TP + FN}, \tag{5.1}$$

the positive predictive value (PPV),

$$PPV = \frac{TP}{TP + FP}. \tag{5.2}$$

and the mean average precision (AP),

$$AP = \frac{TP}{TP + FP + FN}. \tag{5.3}$$

In addition, we will also attempt to compare the performance of different architectures to perform the nuclei segmentation task in the 2D and 3D scenarios. Particularlly, we will attempt to compare Mask-RNN to Cellpose 2D and StarDist 2D, and Stardist 3D to U-Net 3D and Cellpose 3D.

5.2 Astrocyte Classification Tasks

Three major astrocyte classification tasks are outlined: (1) identify if a segmented nuclei belongs to an astrocyte; (2) classify the astrocyte type; (3) classify the multinucleated state. Note that the first and second tasks can be done simultaneously under a multiclass task formulation correponding each segmented nucleus to one of three labels: *1* – protoplasmic-like astrocyte, *2* – fibrous-like astrocyte, or *3* – not an astrocyte, as detailed in section 4.2.2. However, to better assess our model's classification capacities, we will divide this task into two. Starting by evaluating the capacity of the model to distinguish astrocytes in general from other cells, protoplasmic-like and fibrous-like astrocytes classes will be combined into a single class: astrocytes. Ultimately we will attempt to aggregate both astrocyte type identification and characterization of the multinucleated state tasks into a unique classification pipeline.

To address the classification tasks, we will resort to the annotated *3d composite* dataset and attempt to employ the Stardist 3D and the YOLOv8 networks, being the former used for validation. On the one hand, Stardist 3D allows both to segment and classify each nuclei instance on the flight, which can be a good approach for classifying densely packed cells. On the other hand, YOLOv5 has been proved by multiple laboratories as a valuable asset for cell classification (Beghin et al., 2022; Huang et al., 2022) and one of the most accurate classification tools well-generalized for multiple classification domains. In this project we want to assess if its successor, YOLOv8, can equalize the exceptional performance.

For the classification tasks, we must employ transfer learning to map the network result to the desired outputs. In addition, data augmentation strategies will also be employed to increase the dataset training

size and assess its impact on the models' performance.

The models' performance will be evaluated and compared through the measurement of accuracy, the mean AP and mean IoU of the predicted bounding boxes, as accomplished in previous works (Huang et al., 2022; Weigert et al., 2020).

Chapter 6

Preliminary Results

This chapter presents the preliminary results obtained for the instance segmentation of 2D cell nuclei using the InstantDL pipeline for instance segmentation. The best model obtained with the Mask R-CNN architecture (*ftk3* model) is further compared to Stardist 2D and Cellpose 2D.

6.1 Baseline Solution

Mask R-CNN architecture was used to conduct 2D instance segmentation of nuclei in the *2d nuclei* dataset, encompassing 21 bidimensional images (1024×1024 pixels) obtained through fluorescence microscopy. This architecture was validated using six models: two obtained from the literature and four fine-tuned to our dataset. The models from the literature were trained using the ImageNet dataset (*imagenet* model) and the DSB2018 dataset (*dsb2018* model) from the nuclear detection challenge. The latter was fine-tuned to our dataset using training subsets of k-fold cross validations ($k = 3$ and $k = 7$), with and without data augmentation, as previously outlined in Table 5.1.

The *imagenet* model was the least successful among the multiple tests. Its TPR, correspondent to the probability of a segmentation accurately segmenting a nuclei, was of only 20.69%, the PPV, correspondent to the probability of a nuclei being detected by the model, was of 13.32%, and the mean AP was of only 8.82%. Furthermore, there were only identified 485 true positive segmentations out of the 2034 nuclei segmented in the annotated dataset, which indicates a poor sensitivity. In addition, 93.12% of the predictions had IoU coefficients lower than 20% with their respective ground truth.

Using the *dsb2018* model, the nuclei predictions were significantly more accurate. The PPV was 79.13%, however the FP rate was still considerably high, leading to a TPR of only 47.24% and mean AP of 42.01%. On the other hand, the mean IoU of the predictions was 41.65% and 16.28% of the predicted segmentations correspondend to more than one ground truth mask, indicating that this model was overpredicting the nuclei size.

Fine-tuning the model to our dataset significantly improved the segmentation task results. Comparing the results from the 3-fold and 7-fold cross-validations (*ftk3* and *ftk7* models), the TPR and the mean AP of *ftk3* was higher, however its PPV was lower than for *ftk7* model (Table 6.1). Even though *ftk3* model

made more nuclei predictions they were not as specific as the predictions of *ftk7*, and multiple nuclei were aggregated in the same prediction, leading to a higher percentage of the collated predictions value. On the other hand, the mean IoU was comparable between these models.

To assess the impact of data augmentation strategies, standard normalization, feature scaling, horizontal and vertical flips, and Poisson noise were employed. These transformations were applied randomly to the training set images with $1/3$ chance, each. The values of the computed metrics were comparable between the models obtained with data augmentation strategies in the training (*aftk3* and *aftk7*) and its non-augmented homologous (Table 6.1).

Considering all the metrics, *ftk3* was the proposed model that performed better in the given dataset. Despite higher percentage of collated predictions than *ftk7* and *aftk7*, *ftk3* was able to predict more closely the total number of nuclei with outstanding TPR and mean average precision.

Table 6.1: Summary of the models performance regarding the TPR, PPV, mean AP, mean IoU coefficient, mean ratio between predicted and ground truth the number of nuclei, and percentage of collated predictions (Preds).

Model	TPR (%)	PPV (%)	Mean AP (%)	Mean IoU (%)	Mean Ratio (%)	Collated Preds (%)
imagenet	20.69	13.32	8.82	26.64	214.71	0.82
dsb2018	47.24	79.13	42.01	41.65	52.37	16.28
ftk3	89.91	94.02	85.04	62.72	88.12	4.65
ftk7	83.53	95.62	80.45	63.62	80.19	2.70
aftk3	85.10	94.13	80.81	63.50	85.42	5.14
aftk7	84.60	95.69	81.49	63.80	82.15	2.58

6.2 Validation

To corroborate the results from the Mask R-CNN architecture, *ftk3* model was compared to Stardist 2D and Cellpose 2D, without fine-tuning (Table 6.2).

The performance of Stardist 2D, using a model trained with the nuclei segmentation data from the DSB2018, was very competive with the results obtained with Mask R-CNN. The mean ratio between predicted and ground truth number of nuclei indicates Stardist 2D tends to over estimate the effective number of nuclei, in contrast to Mask R-CNN. On the other hand, it does not discriminate individual instances as well, exhibiting a percentage of collated predictions of 4.85% and predicting more false positives, which lowers its PPV.

Contrastingly, Cellpose 2D was not as effective as the two previous architectures. It presented several outliers to extreme under-estimation or over-estimation of the number of nuclei, and a standard

Table 6.2: Validation of the architectures performance regarding the TPR, PPV, mean AP, mean IoU coefficient, mean ratio between predicted and ground truth the number of nuclei, and percentage of collated predictions.

Architecture	TPR (%)	PPV (%)	Mean AP (%)	Mean IoU (%)	Mean Ratio (%)	Collated Preds (%)
Mask R-CNN	89.91	94.02	85.04	62.72	88.12	4.65
Stardist 2D	88.08	85.78	76.86	64.83	108.53	4.85
Cellpose 2D	79.12	26.90	25.12	58.73	468.43	7.46

deviation of 707.47. Also due to this poor estimation, the mean ratio between predicted and ground truth number of nuclei is considerably elevated and its PPV is markedly low.

In Figures 6.1a-b the similarily between the performances of the fine-tuned models using the Mask R-CNN architecture and the Stardist 2D is clearly ilustrated. In addition, Cellpose 2D notably underperforms the remaining models. Furthermore, in Figures 6.1c-f are ilustrated the ground truth masks and the predictions from *ftk3* model, Stardist 2D and Cellpose 2D, respectively. This exemple, ilustrates the inadequacy of Cellpose 2D to generalize to all images, in contrast to the other models.

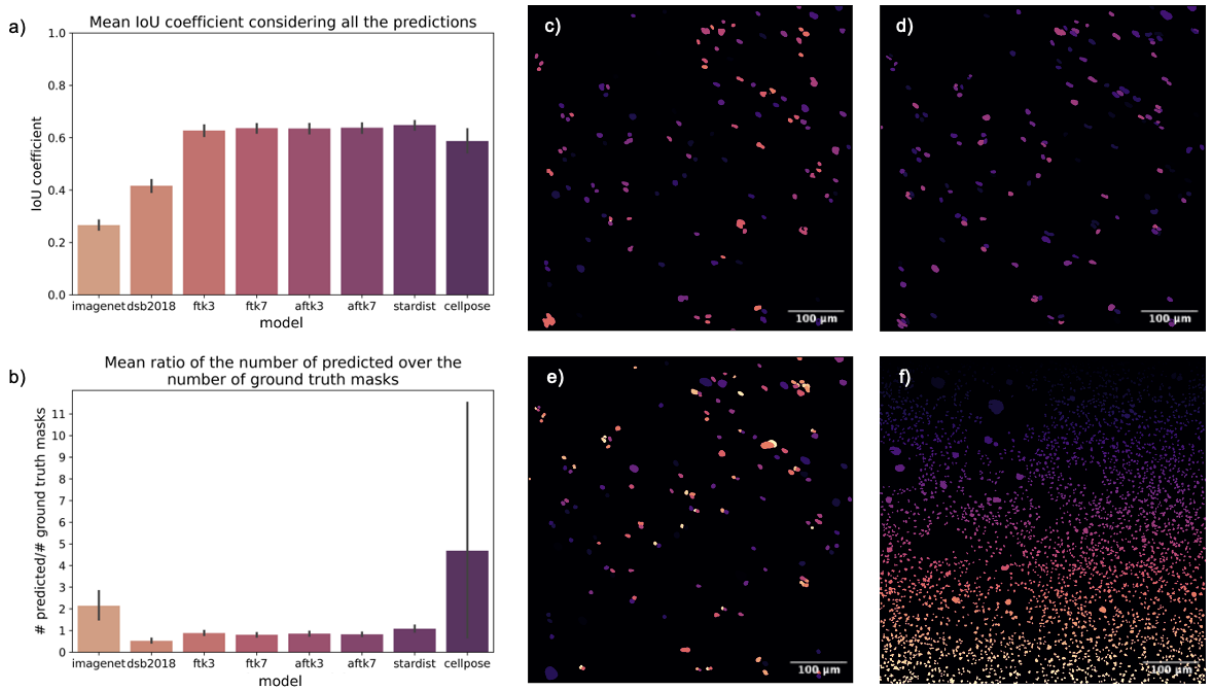

Figure 6.1: Summary and visualization of the performance of the tested models and architectures. (a) Bar plot of the mean IoU coefficient across the multiple models tested; (b) Box plot of the ratio of the number of predicted over the number of ground truth masks across the multiple models tested; (c) Ground truth masks of image 11; (d) Predicted masks in image 11 using *ftk3* model; (e) Predicted masks in image 11 using Stardist 2D; (f) Predicted masks in image 11 using Cellpose 2D.

6.3 Discussion

The presented solution simplifies the network optimization problem by fixing multiple network hyperparameters, such as the batch size, the learning rate, and the number of epochs. The possibility of optimizing these parameters is well-acknowledged in the literature. However, using a batch of size 2, a learning rate of 0.002, and the number of epochs set to 20 allowed us to prove the acceptable efficiency of the network in segmenting nuclei instances in the given dataset. Optimizing these values could potentially enhance the network's performance, although this is not guaranteed. As such, it needs to be validated in future work.

The data augmentation strategies employed did not yield the expected performance improvements in the models as anticipated. To enhance performance, a more tailored approach to augmentation tech-

niques could be explored, optimizing their application to maximize efficacy. For instance, increasing the rate of data transformations on the training set or diversifying the transformation techniques employed can lead to different conclusions and possibly improve the network's performance.

Regarding the validation of Cellpose 2D, there were noticed several concerning aspects of its functioning. Aside from the suboptimal outcomes achieved by Cellpose 2D, several predefined conditions within the algorithm impact its functionality. Notably, the model type selection significantly influences its performance. Despite the utilization of the *cytoplasm* model instead of the *nucleus* model, due to our dataset's intrinsic characteristics, the results were less competitive compared to other architectures. Moreover, this model exhibited significantly longer estimation times, taking up to 60 times longer than alternative models to produce predictions, independently from the number of resulting predictions. Its primary advantage lied in computing the nuclei area and predict its position simultaneously. However this statistic can also be retrieved with relative ease from the foremost models. Nevertheless, we acknowledge that fine-tuning Cellpose 2D model can hold significant improvements in its performance as previously described in Mask R-CNN model.

Conversely, Stardist 2D, without fine-tuning, exhibited highly competitive results, comparable to the best-performing model using the Mask R-CNN architecture. We hypothesize that fine-tuning Stardist 2D to our dataset can easily increase the results' accuracy.

Overall, the mean IoU and mean AP results align well with those reported in previous studies (Schmidt et al., 2018; Stringer et al., 2021). Despite the use of a more conservative threshold IoU in this analysis compared to other studies in the literature, the achieved results affirm the credibility and effectiveness of the selected model. We anticipate that further refinement through additional training iterations will enhance these segmentation outcomes. This suggests that exceptional segmentation results can be attained even when working within the constraints of a limited dataset.

Chapter 7

Conclusions

7.1 Achievements

In the present work, we laid a solid groundwork by exploring essential biological imaging techniques, delving into machine learning and deep learning, with a specific focus on CNNs. A comprehensive literature review has extensively covered primary computer vision tasks in the biomedical domain, particularly in cell culture analysis. This review enabled us to identify the state-of-the-art architectures used for tasks like image classification, object detection, image segmentation, and *in silico* staining, providing insights into their mechanisms and limitations.

Drawing from this comprehensive understanding, we have devised an approach for designing a pipeline for the characterization of astrocyte-neuron 3D cell cultures. Starting with image preprocessing, *pyimagej* package was utilized to automate the essential preprocessing steps. This automation not only accelerated the preprocessing stage but also serves as a robust tool for future analysis. Employing this pipeline, we meticulously annotated and characterized our dataset, which is vital for understanding the dataset's distribution and comparing it to future datasets where our models will be employed.

Furthermore, we have outlined the critical steps necessary for thoroughly characterizing astrocyte-neuron cultures. Our primary focus has been on bidimensional nuclei segmentation and quantification. This task is a cornerstone for future analysis, allowing us to understand the main limitations of the dataset, and motivating the tridimensional analysis. Additionally, we have successfully applied selected networks to this task. Mask R-CNN architecture within the InstantDL pipeline from the Carsten Marr Lab (Helmholtz Zentrum München, Institute of AI for Health), was fine-tuned with six distinct models, and validated with Stardist 2D and Cellpose 2D.

The main challenges encountered in this work were related to the manual annotation of the groud truth nuclei segmentation. This process was not only exceedingly time-consuming – taking approximately 30 minutes for 2D segmentation and 3 hours for 3D segmentation of a single image – but also inherently subjective. Despite these challenges, this arduous task emphasizes the urgency for automation, underscoring its pivotal role in advancing this field of work.

7.2 Future Work

Regarding the computational work underlying this project, in future prospects we aim to extend the 2D nuclei segmentation and quantification to the 3D scenario. Based on previous works, we foresee that the 3D segmentation can be more challenging to achieve. However, we hope that this approach allow us to thoroughly characterize the astrocyte-neuron cultures more precisely than 2D segmentation. In addition, we also ambition to identify the astrocytes in our images and classify them regarding their type and multinucleated state. We aim to achieve accurate classifications that allow us to discern prevalent statistics across different culture conditions.

The laboratory work, planned to start in March and last until December 2024, will initiate with the preparation of the cell-cultures of murine astrocytes in 3D hydrogels under different culture conditions and establish a high-throughput imaging platform for image acquisition using a confocal microscopy setup. After its deployment, the culture images will be analyzed using the aforementioned pipeline. Since the laboratory work at the Helmholtz Zentrum München is already under development, and its advances until I join the team are not clearly established, we cannot deliver an accurate plan of works at the time. Still, Figure 7.1 outlines a provisory plan of the prospective scientific works until the completion of the Master's Thesis project.

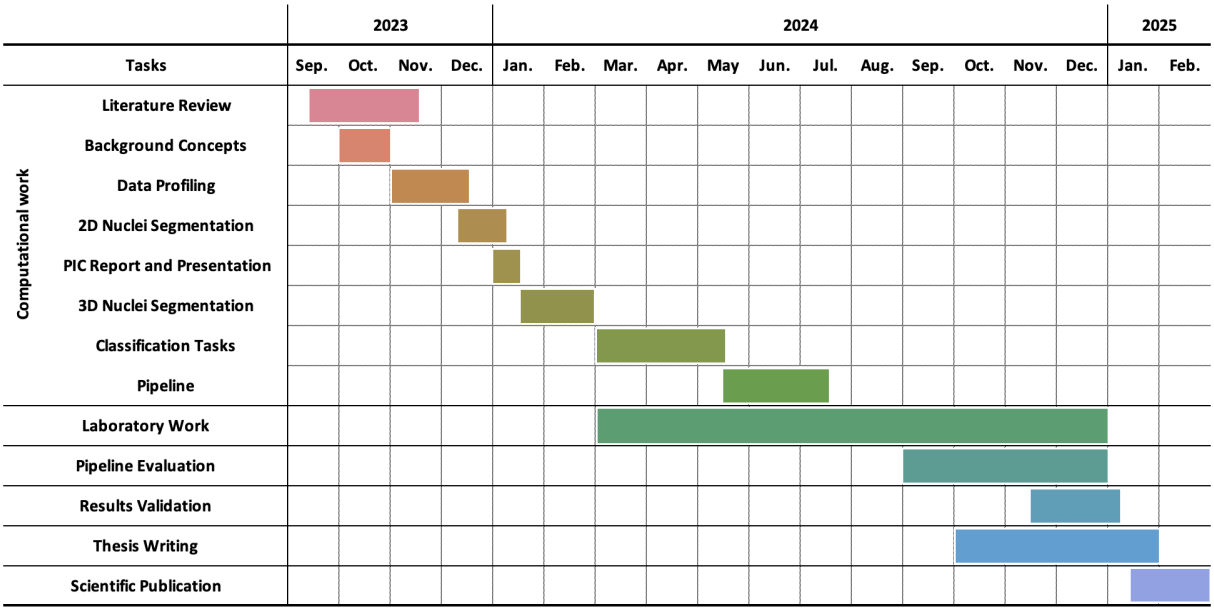

Figure 7.1: Gantt chart of the Master Thesis project planning.

Bibliography

Allen, N. J., & Eroglu, C. (2017). Cell biology of astrocyte-synapse interactions. *Neuron*, *96*(3), 697–708. doi: 10.1016/j.conb.2023.102704

Araki, T., Ikegaya, Y., & Koyama, R. (2021). The effects of microglia-and astrocyte-derived factors on neurogenesis in health and disease. *European Journal of Neuroscience*, *54*(5), 5880–5901. doi: 10.1111/ejn.14969

Atwell, S., Waibel, D. J. E., Boushehri, S. S., Wiedenmann, S., Marr, C., & Meier, M. (2023). Label-free imaging of 3d pluripotent stem cell differentiation dynamics on chip. *Cell Reports Methods*, *3*(7). doi: 10.1016/j.crmeth.2023.100523

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(12), 2481–2495. doi: 10.48550/arXiv.1511.00561

Beghin, A., Grenci, G., Sahni, G., Guo, S., Rajendiran, H., Delaire, T., ... Ong, H. T. (2022). Automated high-speed 3d imaging of organoid cultures with multi-scale phenotypic quantification. *Nature Methods*, *19*(7), 881–892. doi: 10.1038/s41592-022-01508-0

Bellis, S. L. (2011). Advantages of rgd peptides for directing cell association with biomaterials. *Biomaterials*, *32*(18), 4205–4210. doi: 10.1016/j.biomaterials.2011.02.029

Caicedo, J. C., Goodman, A., Karhohs, K. W., Cimini, B. A., Ackerman, J., Haghighi, M., ... others (2019). Nucleus segmentation across imaging experiments: the 2018 data science bowl. *Nature methods*, *16*(12), 1247–1253. doi: 10.1038/s41592-019-0612-7

Cassé, F., Richetin, K., & Toni, N. (2018). Astrocytes' contribution to adult neurogenesis in physiology and alzheimer's disease. *Frontiers in cellular neuroscience*, *12*, 432. doi: 10.3389/fncel.2018.00432

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, *40*(4), 834–848. doi: 10.48550/arXiv.1606.00915

Christiansen, E. M., Yang, S. J., Ando, D. M., Javaherian, A., Skibinski, G., Lipnick, S., ... others (2018). In silico labeling: predicting fluorescent labels in unlabeled images. *Cell*, *173*(3), 792–803. doi: 10.1016/j.cell.2018.03.040

Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T., & Ronneberger, O. (2016). 3d u-net: learning dense

volumetric segmentation from sparse annotation. In *Medical image computing and computer-assisted intervention–miccai 2016: 19th international conference, athens, greece, october 17-21, 2016, proceedings, part ii 19* (pp. 424–432). doi: 10.48550/arXiv.1606.06650

Daniel, J., Rose, J., Vinnarasi, F., & Rajinikanth, V. (2022). Vgg-unet/vgg-segnet supported automatic segmentation of endoplasmic reticulum network in fluorescence microscopy images. *Scanning*, *2022*. doi: 10.1155/2022/7733860

Elliott, A. D. (2020). Confocal microscopy: principles and modern practices. *Current protocols in cytometry*, *92*(1), e68. doi: 10.1002/cpcy.68

Eschweiler, D., Smith, R. S., & Stegmaier, J. (2022). Robust 3d cell segmentation: extending the view of cellpose. In *2022 ieee international conference on image processing (icip)* (pp. 191–195). doi: 10.48550/arXiv.2105.00794

Gao, Y., Zhou, M., & Metaxas, D. N. (2021). Utnet: a hybrid transformer architecture for medical image segmentation. In *Medical image computing and computer assisted intervention–miccai 2021: 24th international conference, strasbourg, france, september 27–october 1, 2021, proceedings, part iii 24* (pp. 61–71). doi: 10.48550/arXiv.2107.00781

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the ieee international conference on computer vision* (pp. 1440–1448). doi: 10.48550/arXiv.1504.08083

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 580–587). doi: https://doi.org/10.48550/arXiv.1311.2524

Godinez, W. J., Hossain, I., Lazic, S. E., Davies, J. W., & Zhang, X. (2017). A multi-scale convolutional neural network for phenotyping high-content cellular images. *Bioinformatics*, *33*(13), 2010–2019. doi: 10.1093/bioinformatics/btx069

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Grade, S., & Götz, M. (2017). Neuronal replacement therapy: previous achievements and challenges ahead. *NPJ Regenerative medicine*, *2*(1), 29. doi: 10.1038/s41536-017-0033-0

Griffiths, B. B., Bhutani, A., & Stary, C. M. (2020). Adult neurogenesis from reprogrammed astrocytes. *Neural Regeneration Research*, *15*(6), 973. doi: 10.4103/1673-5374.270292

Gupta, A., Harrison, P. J., Wieslander, H., Pielawski, N., Kartasalo, K., Partel, G., ... others (2019). Deep learning in image cytometry: a review. *Cytometry Part A*, *95*(4), 366–380. doi: 10.1002/cyto.a.23701

Götz, M., & Huttner, W. B. (2005). The cell biology of neurogenesis. *Nature reviews Molecular cell biology*, *6*(10), 777–788. doi: 10.1038/nrm1739

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the ieee international conference on computer vision* (pp. 2961–2969). doi: 10.48550/arXiv.1703.06870

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778). doi: 10.48550/arXiv.1512.03385

Healy, S., McMahon, J., Owens, P., Dockery, P., & FitzGerald, U. (2018). Threshold-based segmenta-

tion of fluorescent and chromogenic images of microglia, astrocytes and oligodendrocytes in fiji. *Journal of neuroscience methods*, *295*, 87–103. doi: 10.1016/j.jneumeth.2017.12.002

Huang, Y., Kruyer, A., Syed, S., Kayasandik, C. B., Papadakis, M., & Labate, D. (2022). Automated detection of gfap-labeled astrocytes in micrographs using yolov5. *Scientific Reports*, *12*(1), 22263. doi: 10.1038/s41598-022-26698-7

Inkson, B. J. (2016). Scanning electron microscopy (sem) and transmission electron microscopy (tem) for materials characterization. In *Materials characterization using nondestructive evaluation (nde) methods* (pp. 17–43). Elsevier. doi: 10.1016/B978-0-08-100040-3.00002-X

Jocher, G., Chaurasia, A., & Qiu, J. (2023, January). *Yolo by ultralytics.* Retrieved from `https://github.com/ultralytics/ultralytics`

Jäkel, S., & Dimou, L. (2017). Glial cells and their function in the adult brain: a journey through the history of their ablation. *Frontiers in cellular neuroscience*, *11*, 24. doi: 10.3389/fncel.2017.00024

Kandel, E. R., Schwartz, J. H., Jessell, T. M., Siegelbaum, S., Hudspeth, A. J., Mack, S., et al. (2000). *Principles of neural science* (Vol. 4). McGraw-hill New York.

Kayasandik, C. B., Ru, W., & Labate, D. (2020). A multistep deep learning framework for the automated detection and segmentation of astrocytes in fluorescent images of brain tissue. *Scientific reports*, *10*(1), 5137. doi: 10.1038/s41598-020-61953-9

Kulkarni, P. M., Barton, E., Savelonas, M., Padmanabhan, R., Lu, Y., Trett, K., ... Roysam, B. (2015). Quantitative 3-d analysis of gfap labeled astrocytes from fluorescence confocal images. *Journal of neuroscience methods*, *246*, 38–51. doi: 10.1016/j.jneumeth.2015.02.014

Lam, J., Katti, P., Biete, M., Mungai, M., AshShareef, S., Neikirk, K., ... others (2021). A universal approach to analyzing transmission electron microscopy with imagej. *Cells*, *10*(9), 2177. doi: 10.3390/cells10092177

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444. doi: 10.1038/nature14539

Leyendecker, L., Haas, J., Piotrowski, T., Frye, M., Becker, C., Fleischmann, B. K., ... Schmitt, R. H. (2022). A modular deep learning pipeline for cell culture analysis: Investigating the proliferation of cardiomyocytes. In *International conference on medical imaging with deep learning* (pp. 760–773).

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3431–3440). doi: 10.48550/arXiv.1411.4038

Meijering, E. (2020). A bird's-eye view of deep learning in bioimage analysis. *Computational and structural biotechnology journal*, *18*, 2312–2325. doi: 10.1016/j.csbj.2020.08.003

Moen, E., Bannon, D., Kudo, T., Graf, W., Covert, M., & Van Valen, D. (2019). Deep learning for cellular image analysis. *Nature methods*, *16*(12), 1233–1246. doi: 10.1038/s41592-019-0403-1

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Nugraha, S. J. A., & Erfianto, B. (2023). White blood cell detection using yolov8 integration with detr to improve accuracy. *Sinkron: jurnal dan penelitian teknik informatika*, *8*(3), 1908–1916. doi: 10.33395/sinkron.v8i3.12811

Oberheim, N. A., Takano, T., Han, X., He, W., Lin, J. H., Wang, F., ... others (2009). Uniquely hominid features of adult human astrocytes. *Journal of Neuroscience*, *29*(10), 3276–3287. doi: 10.4103/1673-5374.340405

Ounkomol, C., Seshamani, S., Maleckar, M. M., Collman, F., & Johnson, G. R. (2018). Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy. *Nature methods*, *15*(11), 917–920. doi: 10.1038/s41592-018-0111-2

Pachitariu, M., & Stringer, C. (2022). Cellpose 2.0: how to train your own model. *Nature methods*, *19*(12), 1634–1641. doi: 10.1038/s41592-022-01663-4

Pawlowski, N., Caicedo, J. C., Singh, S., Carpenter, A. E., & Storkey, A. (2016). Automating morphological profiling with generic deep convolutional networks. *BioRxiv*, 085118. doi: 10.1101/085118

Petit, O., Thome, N., Rambour, C., Themyr, L., Collins, T., & Soler, L. (2021). U-net transformer: Self and cross attention for medical image segmentation. In *Machine learning in medical imaging: 12th international workshop, mlmi 2021, held in conjunction with miccai 2021, strasbourg, france, september 27, 2021, proceedings 12* (pp. 267–276). doi: https://doi.org/10.48550/arXiv.2103.06104

Pratapa, A., Doron, M., & Caicedo, J. C. (2021). Image-based cell phenotyping with deep learning. *Current opinion in chemical biology*, *65*, 9–17. doi: 10.1016/j.cbpa.2021.04.001

Preston, A. N., Cervasio, D. A., & Laughlin, S. T. (2019). Visualizing the brain's astrocytes. In *Methods in enzymology* (Vol. 622, pp. 129–151). Elsevier. doi: 10.1016/bs.mie.2019.02.006

Pérez-Rodríguez, D. R., Blanco-Luquin, I., & Mendioroz, M. (2021). The participation of microglia in neurogenesis: a review. *Brain sciences*, *11*(5), 658. doi: 10.3390/brainsci11050658

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 779–788). doi: 10.48550/arXiv.1506.02640

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*. doi: 10.48550/arXiv.1506.01497

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–miccai 2015: 18th international conference, munich, germany, october 5-9, 2015, proceedings, part iii 18* (pp. 234–241). doi: 10.48550/arXiv.1505.04597

Rueden, C. T., Hiner, M. C., Evans III, E. L., Pinkert, M. A., Lucas, A. M., Carpenter, A. E., ... Eliceiri, K. W. (2022). Pyimagej: A library for integrating imagej and python. *Nature methods*, *19*(11), 1326–1327. doi: 10.1038/s41592-022-01655-4

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, *115*(3), 211-252. doi: 10.1007/s11263-015-0816-y

Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson.

Sadak, F., Saadat, M., & Hajiyavand, A. M. (2020). Real-time deep learning-based image recognition for applications in automated positioning and injection of biological cells. *Computers in Biology and*

Medicine, *125*, 103976. doi: 10.1016/j.compbiomed.2020.103976

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., ... others (2012). Fiji: an open-source platform for biological-image analysis. *Nature methods*, *9*(7), 676–682. doi: 10.1038/nmeth.2019

Schmidt, U., Weigert, M., Broaddus, C., & Myers, G. (2018). Cell detection with star-convex polygons. In *Medical image computing and computer assisted intervention–miccai 2018: 21st international conference, granada, spain, september 16-20, 2018, proceedings, part ii 11* (pp. 265–273). doi: 10.1007/978-3-030-00934-2_30

Shou, Y., Liang, F., Xu, S., & Li, X. (2020). The application of brain organoids: from neuronal development to neurological diseases. *Frontiers in cell and developmental biology*, *8*, 1092. doi: 10.3389/fcell.2020.579659

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. doi: 10.48550/arXiv.1409.1556

Sommer, C., Hoefler, R., Samwer, M., & Gerlich, D. W. (2017). A deep learning and novelty detection framework for rapid phenotyping in high-content screening. *Molecular biology of the cell*, *28*(23), 3428–3436. doi: 10.1091/mbc.E17-05-0333

Staffler, B. S. (2019). *Machine learning for connectomics* (Unpublished doctoral dissertation). Technische Universitat Munchen.

Stringer, C., Wang, T., Michaelos, M., & Pachitariu, M. (2021). Cellpose: a generalist algorithm for cellular segmentation. *Nature methods*, *18*(1), 100–106. doi: 10.1038/s41592-020-01018-x

Stylianou, A., Gkretsi, V., Sampaio, P., Glösmann, M., & Walter, A. (2021). Transmission light microscopy [Book Chapter]. In *Imaging modalities for biological and preclinical research: A compendium, volume 1* (p. I.1.a-1 to I.1.a-15). IOP Publishing. Retrieved from 10.1088/978-0-7503-3059-6ch1 doi: 10.1088/978-0-7503-3059-6ch1

Sugimoto, T., Ito, H., Teramoto, Y., Yoshizawa, A., & Bise, R. (2022). Multi-class cell detection using modified self-attention. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 1855–1863). doi: 10.1109/CVPRW56347.2022.00202

Suleymanova, I., Balassa, T., Tripathi, S., Molnar, C., Saarma, M., Sidorova, Y., & Horvath, P. (2018). A deep convolutional neural network approach for astrocyte detection. *Scientific reports*, *8*(1), 12878. doi: 10.1038/s41598-018-31284-x

Sun, W., Cornwell, A., Li, J., Peng, S., Osorio, M. J., Aalling, N., ... others (2017). Sox9 is an astrocyte-specific nuclear marker in the adult brain outside the neurogenic regions. *Journal of Neuroscience*, *37*(17), 4493–4507. doi: 10.1523/JNEUROSCI.3199-16.2017

Terven, J., & Cordova-Esparza, D. (2023). A comprehensive review of yolo: From yolov1 to yolov8 and beyond. *arXiv preprint arXiv:2304.00501*. doi: 10.48550/arXiv.2304.00501

Tran, T., Kwon, O.-H., Kwon, K.-R., Lee, S.-H., & Kang, K.-W. (2018). Blood cell images segmentation using deep learning semantic segmentation. In *2018 ieee international conference on electronics and communication engineering (icece)* (pp. 13–16). doi: 10.1109/ICECOME.2018.8644754

Waibel, D. J. E., Shetab Boushehri, S., & Marr, C. (2021). Instantdl: an easy-to-use deep learn-

ing pipeline for image segmentation and classification. *BMC bioinformatics*, *22*, 1–15. doi: 10.1186/s12859-021-04037-3

Waibel, D. J. E., Tiemann, U., Lupperger, V., Semb, H., & Marr, C. (2019). In-silico staining from bright-field and fluorescent images using deep learning. In *Artificial neural networks and machine learning–icann 2019: Image processing: 28th international conference on artificial neural networks, munich, germany, september 17–19, 2019, proceedings, part iii 28* (pp. 184–186). doi: 10.1007/978-3-030-30508-6_15

Wang, H., Shang, S., Long, L., Hu, R., Wu, Y., Chen, N., ... others (2018). Biological image analysis using deep learning-based methods: literature review. *Digital Medicine*, *4*(4), 157. doi: 10.4103/digm.digm_16_18

Wang, Y., Ahsan, U., Li, H., Hagen, M., et al. (2022). A comprehensive review of modern object segmentation approaches. *Foundations and Trends® in Computer Graphics and Vision*, *13*(2-3), 111–283. doi: https://doi.org/10.48550/arXiv.2301.07499

Weigert, M., Schmidt, U., Haase, R., Sugawara, K., & Myers, G. (2020). Star-convex polyhedra for 3d object detection and segmentation in microscopy. In *Proceedings of the ieee/cvf winter conference on applications of computer vision* (pp. 3666–3673). doi: 10.1109/WACV45572.2020.9093435

Wichert, A. M., & Sa-Couto, L. (2021). *Machine learning-a journey to deep learning: With exercises and answers*. World Scientific.

Yin, X.-X., Sun, L., Fu, Y., Lu, R., Zhang, Y., et al. (2022). U-net-based medical image segmentation. *Journal of Healthcare Engineering*, *2022*. doi: 10.1155/2022/4189781

Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., & Lee, B. (2022). A survey of modern deep learning based object detection models. *Digital Signal Processing*, *126*, 103514. doi: 10.1016/j.dsp.2022.103514

Zaki, M. J., & Meira Jr, W. (2020). *Data mining and machine learning: fundamental concepts and algorithms*. Cambridge University Press.