MELBA Journal Machine Learning for Biomedical Imaging

AIxCell: A Domain-Specific and Meta-Learning based AutoML-System for Cellular Image Segmentation

Jan-Henner Roberg [1], Lars Leyendecker [2], Sebastian Schönleben [2], Robert H. Schmitt [2,3]

1 Fraunhofer Center for Assistive Information and Communication Solutions AICOS, Rua Alfredo Allen 455/461, 4200-124 Porto, Portugal

2 Fraunhofer Institute for Production Technology IPT, Steinbachstraße 17, 52074 Aachen, Germany

3 Laboratory for Machine Tools and Production Engineering (WZL) of RWTH Aachen University, Campus Boulevard 30, 52074 Aachen, Germany

Abstract

Biomedical image analysis, especially in cell and tissue microscopy, is a necessary and tedious task in bio laboratories that requires biomedical expertise and is therefore highly cost-intensive. Additionally, the subjectivity of the analysis and the susceptibility to human errors affect the comparability of scientific results. While deep learning holds promise for automating these analysis tasks, biomedical specialists rarely possess the necessary skill set and capacities to develop, deploy and maintain robust DL applications for their individual analyses. We propose a four-stage domain-specific image automated machine learning (AutoML) system architecture that aims to balance the generality-specificity trade-off that AutoML-systems typically suffer from and apply it to the domain of cellular image analysis. We introduce *AIxCell* to automate the design, construction and training of deep learning-based pipelines for cellular image analysis. By leveraging a portfolio-based meta-learning approach and multi-fidelity training (i.e., successive halving), *AIxCell* identifies, trains and provides the optimal image analysis pipeline to the biomedical expert. The results show the effectiveness of meta-learning in a domain-specific setting and that *AIxCell* reliably outperforms baseline solutions. Our findings highlight the potential of domain-specific image AutoML-systems to enhance efficiency and limit cost in biomedical image analysis.

Keywords

Cellular Image Analysis, Domain-Specific AutoML, Meta-Learning, Multi-Fidelity Optimization, Deep Learning

Article informations

`https://doi.org/https://doi.org/10.59275/j.melba.2026-gdgb`©2026 J. Roberg, L. Leyendecker, S. Schönleben, R. H. Schmitt. License: CC-BY 4.0

Volume 2026, Received: 2024-09, Published 2026-02

Corresponding author: jan.henner.roberg@gmail.com

1. Introduction

Analyzing biomedical image data, e.g. in cell and tissue microscopy, is labor-intensive, time-consuming and cumbersome. These tasks are performed by specialized laboratory personnel and are therefore cost-intensive. Moreover, the subjectivity of the analysis and the susceptibility to application- and instrument-specific errors affect the comparability of scientific results. The automation of these analyses would allow biomedical experts, such as biologists, physicians, and virologists, to use their time more effectively for creative, value-adding activities and would objectify study results. Deep learning (DL) algorithms, which use deep artificial neural networks for semantic knowledge extraction from image data, have achieved impressive results especially in the life science domain (Anagnostidis et al., 2020; Xu et al., 2022; Moen et al., 2019; van Valen et al., 2016; Leyendecker et al., 2022). However, developing DL applications requires experience and expertise in data science, machine learning, information technology, and software development. Biomedical specialists rarely have the necessary skillset and capacities to develop, deploy and maintain robust DL applications for their individual analyses. Moreover, commissioning specific stand-alone solutions for a variety of use-cases is oftentimes not economically feasible for clinical institutions and laboratories. Therefore, to enable biomedical specialists to automate their individual cell and tissue analyses with the help of DL-based image analysis pipelines, technology is required to fully automate the development task.

Given this objective, we propose *AIxCell*, a domain-specific automated machine learning (AutoML) system, that aims to balance the generality-specificity trade-off between different image analysis tasks. *AIxCell* aims to automatically configure and train the best possible DL-based image analysis pipeline for the microscopy image dataset and analyses task description provided by the user. The first and most important task for cellular image analyses is segmentation of the regions of interest. Therefore, *AIxCell* configures pre-processing, neural network architecture and hyperparameters for optimal segmentation under the conditions of user-defined computational resources. To finalize the analysis task, based on the segmented image, *AIxCell* allows the user to configure post-processing steps tailored to their use-case. The identified best data pipeline is provided to the expert for integration into their analysis workflow. Biomedical image analysis, especially in cell and tissue microscopy, is crucial but often labor-intensive and prohibitively expensive. While deep learning holds promise for automating these tasks, its effectiveness typically requires task-specific adjustments made by human experts. We introduce *AIxCell*, a domain-specific automated machine learning (AutoML) system tailored for cellular image analysis to automate these task-specific adaptations. By leveraging meta-learning and successive halving, *AIxCell* identifies the optimal image analysis pipeline, aiming to support biomedical experts by alleviating the complexities of solution development. The meta-data used for training the meta-learning models consist of 1,423 evaluations of pipeline performances on eight different microscopy analyses use-cases. We investigate how different meta-features (i.e., information on the analyses task and dataset similarities) impact the performance of the meta-models and the final accuracy of the segmentation. Our results show the effectiveness of meta-learning in a domain-specific setting and that *AIxCell* reliably outperforms baseline solutions, while being outperformed by SOTA solutions developed by human experts. Nonetheless, we illustrate the potential of domain-specific AutoML-systems for image analyses tasks.

2. Related Works

In this section, we review the state-of-the-art for cellular image analysis tools, AutoML for image data and meta-learning-based approaches for tabular data. Finally, we highlight how the presented experiments and overall system contribute to research in these fields.

2.1 Systems for Cellular Image Analyses

A variety of systems that allow biomedical experts to analyze cellular images have been developed, here we will briefly review a few. Ilastik (Berg et al., 2019) contains pre-defined workflows for image segmentation, object classification, counting, and tracking. With its impressive user interface and good performance of the default workflows, it finds widespread adoption. To perform the segmentation task, Ilastik relies on a default workflow, that consists of pixel-wise feature extraction and then pixel-wise classification using a random forest. CellProfiler (Jones et al., 2008) is another cellular image analysis software tool. The main focus of CellProfiler is on automatically extracting features and offering a wide variety of visualization tools. To perform the segmentation task, researchers have to add their own workflow since no default option is provided. Cellpose (Stringer et al., 2021) is a recent model for instance segmentation of cellular images, that offers state-of-the-art performance and does not require re-training for new datasets. The model is a residual U-Net trained on a large dataset composed of cellular images from a variety of microscopy modalities and fluorescent markers. Cellpose achieves great performance due to this training on a carefully curated, diverse dataset and by learning to predict gradients that point to the center of the individual objects. Instance segmentation is different to semantic segmentation and it is unclear if the approach of Cellpose would work for semantic segmentation. None of the presented systems perform dataset-specific optimization of their segmentation workflows. This tuning of hyperparameters is important for optimal performance of machine learning systems (Olson et al., 2017).

AutoML focuses on finding optimal ML solutions for new problems (Hutter et al., 2019). Most research within AutoML has focused on tabular data and a variety of well-performing systems are available (Kotthoff et al., 2019; Zimmer et al., 2020; Erickson et al., 2020; Le et al., 2020; Feurer et al., 2019). These systems mostly rely on traditional machine learning approaches, while for image data deep learning approaches are superior. There are significant differences between optimizing traditional ML-pipelines and DL-pipelines (Hutter et al., 2019). In general, predicting an optimal DL-pipeline for biomedical images is challenging due to diverse images, multiple image modalities, and approaching new tasks (Zhang and Cao, 2019).

2.2 AutoML for Image Data

Research on AutoML for image data focuses on neural architecture search (NAS) that aims to find the optimal DL architecture for a given task (Elsken et al., 2018). A successful AutoML tool that performs NAS is AutoKeras (Jin et al., 2023). AutoKeras combines Bayesian optimization with a novel method called neural morphism. This is the process of modifying a given neural network architecture by applying discrete operations such as inserting a layer, adding a skip connection between two layers. What morphism operation to apply and evaluate next is determined by utilizing Bayesian optimization. The benefit of this network morphism is that previously trained weights can be reused, reducing the evaluation time for different configurations. Moreover, the functionality of the network is preserved throughout the optimization process. Optimizing the hyperparameters of a DL-pipeline is different from NAS. Due to changing hyperparameters influencing the entire training process, techniques such as neural morphism are not applicable. While finding an optimal DL architecture is important for optimal performance, tuning hyperparameters is also crucial.

One AutoML-system for computer vision that optimizes both neural architecture and hyperparameters is T-AutoML (Yang et al., 2021). Specifically developed for 3D lesion segmentation in medical images, its search space is similar to the presented system. In the T-AutoML workflow, 100 uniformly sampled configurations are trained on the target dataset and a transformer model learns to predict the better validation accuracy given two configurations. Using the 100 evaluated and an additional 100 randomly sampled configurations the transformer predicts the best out of these 200 configurations. In this way, T-AutoML has achieved state-of-the-art performance on two benchmark lession detection datasets.

Another popular system that configures the entire segmentation pipeline based on the UNet architecture is NN-UNet (Isensee et al., 2021). NN-UNet relies on a set of carefully designed heuristics to automatically configure its pipeline for each dataset, rather than using traditional hyperparameter search methods. These heuristics govern key aspects of the segmentation process: preprocessing, network architecture, training, and model selection. Preprocessing and network architecture selection are based on image size, image modality and available compute resources. During training, NN-UNet employs a fixed set of data augmentation techniques with dynamic learning rate adjustment. This heuristic-driven approach enables NN-UNet to adapt to diverse medical imaging tasks without manual intervention and makes it a highly competitive comparison for image segmentation tasks.

In addition, a variety of proprietary AutoML-systems exist. Google AutoML[1], H20 hydrogen torch[2], Microsoft Azure AutoML[3], Amazon Sagemaker[4] and BigML's OptiML[5] all offer automatic configuration of DL-models for image analyses. The user, therefore, does not need to configure a model themselves, but can just upload their dataset, pay a fee, and get a trained model in return. A comparison of some of the most prominent AutoML-systems for image classification was provided by Yang et al. (2023) who developed MedMNIST an AutoML benchmark for biomedical image classification.

2.3 Meta-Learning in AutoML

In AutoML, meta-learning or learning to learn, refers to using previous experience to learn how to perform a new task (Vanschoren, 2019). Previous experience is available in the form of meta-data consisting of tuples $P(i, j)$ where i is some dataset and j is the pipeline configuration. For each tuple $P(i, j)$ there are some evaluation scores such as accuracy and run-time, showing how well configuration j did on dataset i.

There are different ways in which AutoML-systems use this meta-data to find optimal ML solutions for new datasets. For instance, the meta-data can be used to warm-start an optimization process by first evaluating configurations that perform well on similar datasets and are therefore likely to perform well. This has been used in the (early) version of Auto-Sklearn which has won many AutoML competitions (Feurer et al., 2019). Another way of leveraging the meta-data is to have a meta-model learn to predict performance metrics based on a concatenation of the vectors describing the configuration j and the dataset i. In this way, the meta-model can then be used to rank all possible configurations for a new dataset. Then the top-N predicted configurations can be evaluated on the new dataset and the configuration is determined.

Two examples of tabular AutoML-systems that use a ranking approach are AutoBagging and RankML. AutoBagging (Pinto et al., 2017) finds the best out of 63 possible bagging workflow configurations by using an XGBoost model trained on a meta-base that includes 140 classification datasets with a total of 8,820 datapoints. To quantify datasets, they extracted 158 meta-features using an automatic meta-feature extraction tool (Pinto et al., 2016). RankML (Laadan et al., 2019) searches various available components with scikit-learn, for an optimal ML workflow for a given classification or regression task. The overall search space is limited by representing configurations as acyclic graphs, in the order that machine learning pipelines are usually designed. To construct the meta-base

1. `https://cloud.google.com/automl`
2. `https://h2o.ai/platform/ai-cloud/make/hydrogen-torch/`
3. `https://azure.microsoft.com/en-us/services/machine-learning/automatedml`
4. `https://cloud.google.com/automl`
5. `https://bigml.com/whatsnew/optiml`

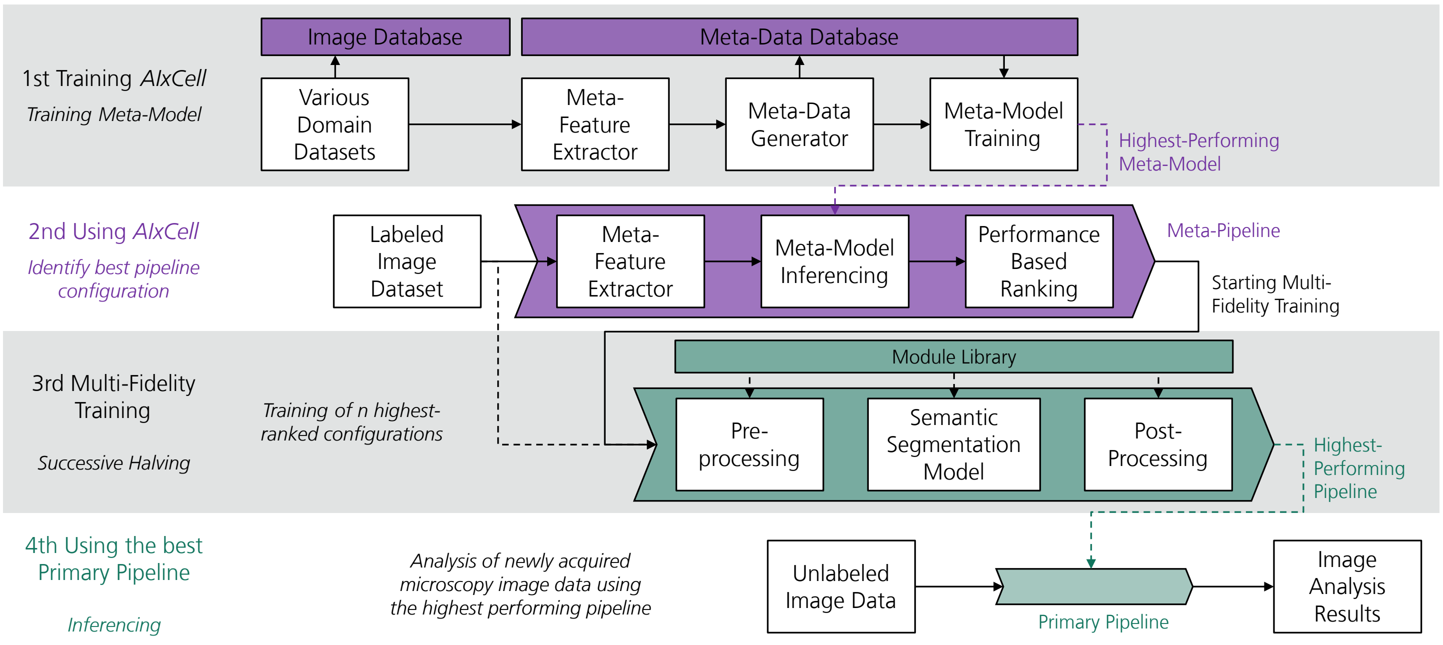


Figure 1: Schematic structure and sequence diagram of the domain-specific AutoML-System

ML-pipelines were randomly constructed for each task, regression or classification, and dataset. In this way, 142,006 pipelines for classification and 171,482 pipelines for regression were evaluated. Overall 149 classification and 79 regression datasets were included in the meta-base. An XGBoost model with a pairwise loss function was trained on this meta-base and then used to predict the top configurations for novel datasets. The resulting system achieved performances similar to state-of-the-art AutoML-systems on various AutoML benchmarks, at a fraction of time. So RankML showed that it is possible to predict optimal ML-pipelines for a new task, given a sufficiently large meta-base. This is especially beneficial when the goal is to find an optimal configuration in a short amount of time.

2.4 Our Contribution

As shown above, for cellular image analyses, a variety of public and proprietary software tools exist that offer many functionalities and are of great value to biomedical experts. However, none of these systems perform dataset-specific adaptations of the segmentation workflow. Optimizing hyperparameters by using AutoML techniques could improve the performance of these systems. Existing works on image-based AutoML focus on NAS revealing that research into hyperparameter tuning of vision systems is scarce. The main challenge in optimizing DL-pipelines for image data is that it takes significant time to evaluate a configuration. Therefore, many optimization-based approaches that were developed for tabular data are not viable in the image regime. One technique from tabular AutoML-systems that has shown promise for finding a configuration in a short time frame is meta-learning. The caveat of meta-learning is that it requires the construction of a meta-base. The large computational cost of evaluating a configuration on image data limits the possible size of meta-bases for image data. For finding an optimal balance in the generality-specificity trade-off, we propose a domain-specific AutoML-system by purposefully limiting the application domain of our AutoML-system to microscopy images of cells and tissue. By doing so, we aim to utilize meta-learning with a relatively small meta-base contributing to the real-world challenges in biomedical research. To our best knowledge, such a meta-learning-based domain-specific AutoML-system for cellular image segmentation has not been developed before.

3. Methods

In this section, we start by explaining the structure and functions of *AIxCell*. Following that, we provide a detailed description of its individual components. These include the module library, search space, meta-features, meta-learning, and the multi-fidelity approach.

3.1 Domain-Specific AutoML-System Architecture and Functionality

According to the schematic flow chart displayed in Figure 1, *AIxCell* comprises four stages and an interplay of multiple core components, which we will describe briefly in the following and in more detail in the later sections. It should be noted that the first stage involves the training and updating of *AIxCell*. This is only relevant for the developers and maintainers not the users of the system. Using *AIxCell* involves

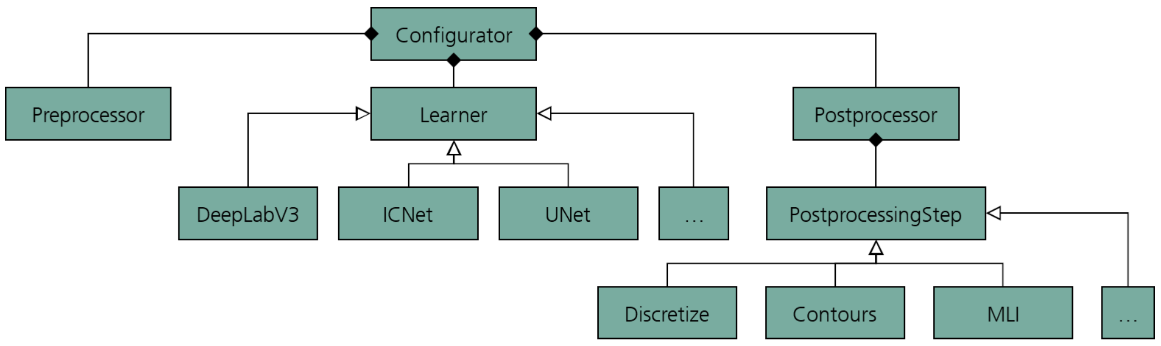


Figure 2: A visualization of the module libraries architecture. Abstractions facilitate the exchange of models as well as preprocessing and postprocessing components.

stages 2, 3, and 4 (see Figure 1). To train and update the AutoML-System, in stage 1, a new meta-model is trained, which ranks pipeline configurations depending on their predicted performance for a given dataset. Stage 1 can only be started by the maintainer of the *AIxCell* instance. Two databases are used: First, the *Image Database* that stores the labelled microscopy image datasets (unstructured data). Second, the tabular *Meta-Data Database*, which contains the performances of different prime pipeline configurations on different datasets and the characteristics (meta-features) of these datasets (structured data). The meta-data is collected by the *Meta-Data Generator* that constructs, trains and evaluates various DL-pipelines on all datasets. The *Meta-Feature Extractor* (see Section 3.4) extracts various features from the image dataset and combines them with user-defined meta-features. These meta-features are used for characterizing both the analysis task and the image dataset. A precise characterization of the analysis task is crucial for our portfolio-based AutoML-System allowing the meta-model to draw relationships between problem characteristics and solution configurations and performances. Finally, multiple *Meta-Models* are trained and evaluated on the *Meta-Data Database*. The best-performing *Meta-Model* is used in the *Meta-Pipeline* in stage 2.

In stage 2, the biomedical experts use *AIxCell* to identify a best-performing DL-pipeline for their analyses task and dataset. The *Meta-Feature Extractor* extracts the meta-features of the dataset and calculates the similarity measure with respect to the other datasets. The *Meta-Model* predicts the IoU-performance of all N possible DL-pipeline configurations and ranks them accordingly. Afterwards, in stage 3, the automated multi-fidelity training of the n out of N best-performing DL-pipeline configurations is started. Therefore, the pipelines are constructed according to their configurations based on the *Module Library*, that stores all preprocessing, modeling, and postprocessing modules. We use successive halving (see Section 3.7) as multi-fidelity approach. At set training intervals, half of the configurations that are currently performing worst are successively dropped. This continues until there is only one configuration left, which is then trained to completion and used for inferencing in stage 4. Stage 4 resembles the actual application of the DL-pipeline for analyzing newly acquired data points or datasets (inference mode).

3.2 Module Library

The module library provides the building blocks of the primary DL-pipeline. According to figure 1 the pipeline consists of preprocessing, model and postprocessing. The Meta-Model or a human user can define the pipeline by choosing from the provided options and configure the hyperparameters. Then the module library can be used to construct, train and infer the pipeline.

Figure 2 shows the structure of the module library. The three components, the preprocessor, learner and postprocessor, are managed by the configurator. The preprocessor is responsible for loading the image data and preparing them so that the images can be processed by the DL-model. Preprocessing can include for example patching, slicing, augmentation, channel removal, and stacking. The learner component represents the model and offers a variety of model architectures, e.g. UNet, ICnet and DeepLabV3, that can be configured via hyperparameters. The last part of the DL-pipeline is the postprocessing of the segmented image, consisting of various operations including generic and use case specific processing as well as visualization of results. Examples are stitching and discretization of the model output, the computation of the confluence of a cell culture or the mean linear intercept of lung tissue. The configurator receives the definition of the pipeline in the form of a YAML-file, builds it by selecting the specified implementation of the abstract components and can then either train or retrain the model or infer the pipeline.

During the design of the module library special attention has been paid to the aspects of reusability and exchangeability of components as well as extending the libraries by additional component implementation. First of all, the abstract learner component allows the usage of models with different architectures and hyperparameter configuration with minimal necessary adaption of the rest of the pipeline. As shown in figure 2 the postprocessor has an internal pipeline structure consisting of multiple PostprocessingStep instances. Second, while different use cases require different postprocessing, they have certain operations, like discretization or confluence computation, in common. Due to this structure, they can be easily reused in different pipelines. Finally, the declarative definition of the pipelines in form of the YAML-files is readable for both the human user and the components of *AIxCell*, allowing reuse of existing pipeline definitions as template for new pipelines.

3.3 Search Space and Default Settings

For the design of an AutoML-system, a core decision to be made is to choose what hyperparameters to automatically optimize. The challenge for designing this search space is to bias the system towards good solutions while including enough variability to find an optimal configuration while reducing computational efforts to the necessary minimum.

For meta-learning-based domain-specific AutoML-systems, the search space needs to be customized to the problem and be restricted enough to allow for successful meta-learning on a small meta-base. In order to not require extensive computational resources for training, we limit both patch-size and batch-size.

As a foundation for the development of *AIxCell*, we developed multiple DL-pipeline configurations on different cell-biological datasets. In this process, we selected the following hyperparameters to be tuneable for dataset-specific adaptation: batch size, patch size, image augmentations, network backbone, and learning rate. The patch size refers to the size of the patches extracted from the original images. Patching is reasonable for semantic segmentation in microscopic images, because of the high dimensionality of the raw images making processing them as a whole through a DNN infeasible. Because we identified different forms of image augmentations to show large impact on model performance, we included flipping, grid distortion, and a combination of brightness and blurring into the search space. Image augmentations were randomly applied during the training process. A full list of all hyperparameters and their possible values can be found in Table 1.

Automatic class-based sampling Furthermore, we encountered that class-based sampling to counter unbalanced datasets is important. Learning class weights in the AutoML-system will greatly increase the computational complexity. Therefore, we define a simple rule to automatically determine when and how to apply class-based sampling. First a threshold based on the pixel-wise class distribution ***x*** and number of classes ***n*** in a dataset is calculated:

Table 1: The Search space considered in *AIxCell*. For the augmentations, the value refers to the probability of applying an augmentation during the training process. Overall, there are 576 possible configurations.

Hyperparameter	**Possible Values**
Patch size	64, 128, 256
Batch size	4, 8, 16, 32
Learning rate	0.0005, 0.001, 0.01
Backbone	ResNet-18, ResNet-50
Augmentation: Flip	0, 0.5
Augmentation: Brightness	0, 0.5
Augmentation: Grid distortion	0, 0.5

$\mathbf{t} = \frac{\sum x}{n} + \frac{\sum x}{n} * 0.75$. If any of the classes occur on a larger percentage of pixels then ***t*** the dataset is deemed imbalanced.

In that case, an acceptable low and high threshold for each class distribution value is calculated: $a_{low} = \mu(x) - \mu(x) * 0.5$ and $a_{high} = \mu(x) + \mu(x) * 0.2$. If a class occurrence percentage $x[i]$ is below or above this threshold, a re-sampling factor is calculated: $factor = a_{low}$ or $a_{high}/x[i]$. The factor is capped at value of 2 for up-sampling, but not capped for down-sampling. We do this to limit the number of duplicated images in the dataset. Class-based (re-)sampling is done based on the patches extracted from the images. Please note that, depending on the dataset, parts of the image might not belong to any class. Therefore we use $\sum x$ for the imbalance threshold and $\mu(x)$ for the acceptable threshold.

Default Settings While we allow certain hyperparameters to be adaptive, others are set to remain at their default values. Given preliminary experiments, we decided to use the U-Net model architecture (Ronneberger et al., 2015) on default in *AIxCell*. The encoder part of the network has five down-sampling layers that each contain four convolutional blocks. The convolutional blocks are based on ResNet and contain 18 or 50 layers. The decoder part of the network also has five convolutional blocks that follow the same structure as the encoder blocks. The activation function of the final layer is sigmoid for binary segmentation and softmax for multi-class problems. The U-Net model utilizes Adam optimizer and categorical focal loss (Lin et al., 2018). For multi-class problems, the loss is adopted by including a weighting factor γ, set to a value of 2 in all experiments, that increases the importance of difficult-to-analyze samples while decreasing the weight of easy-to-analyze samples (Nguyen et al., 2018). Finally, we define the patch stride as half the patch size on default.

Table 2: Engineered Meta-Features, created by combining handcrafted Meta-Features with hyperparameter settings. Sum(image augmentations) refers to how many of the three augmentations are applied and backbone is set to 0.5 for ResNet-18 or 2 for ResNet-50.

Engineered Features	**Calculation**
number of patches per image	(mean image shape/patch size)^2
batches per epoch	n patches * n instances / batch size
augmentation batches	sum(image augmentations) * 2 * batches per epoch
learning rate batches	learning rate * batches per epoch
backbone classes	backbone * number of classes

3.4 Meta-Features

In our portfolio-based AutoML-system, to enable meta-learning, a precise characterization of the dataset and the analysis task is key for attributing model performances to pipeline configurations and task characteristics. We do so by extracting both task and dataset specific meta-features. Due to the limited amount of meta-training data, the dimensionality of data-specific meta-features must be kept to a minimum of informative features. In the presented study, three different types of meta-features are evaluated. Handcrafted meta-features are features of datasets that DL-experts typically use when optimizing DL-configurations. User-defined features refer to features given by the biomedical expert, and finally optical meta-features that aim to quantify the visual characteristics of the images.

Handcrafted Meta-Features We extract the following set of handcrafted meta-features: the number of instances, the number of classes, the class distribution, the mean image shape, and the image format (RGB or grayscale). The mean image shape refers to the mean of the average width and height of all images in a dataset. Class distribution is calculated pixel-wise. For each class, we calculate the number of pixels over the total amount of pixels. Therefore, the class distribution is represented by a vector with the length of the number of classes. Since the number of meta-features is required to be identical for each dataset, we compress this list to a scalar value. This value is calculated by taking the average distance between class distribution values. To illustrate this consider a dataset with three classes. Averaged across all images in the dataset, class one occurs on 20% of all pixels, class two on 10% and class three on 70%. This leads to a class distribution vector of [0.2, 0.1. 0.7]. The average distance between class distribution values is 0.4, this is the scalar meta-feature used to represent class distribution.

In addition, we extract information about the regions of interest (ROI) in the images. In semantic segmentation there are multiple ROI per class, to compress these into usable meta-features we calculate the mean, standard deviation, minimum, and maximum ROI size for each class. These are then averaged for each image, and finally averaged across all images in the dataset yielding four additional dataset-specific meta-features. Overall, *AIxCell* utilizes nine handcrafted meta-features for each dataset.

User-defined Meta-Features To also take the characteristics of the analysis task and the data acquisition modality into consolidation, we include three different meta-features provided by the biomedical expert: the microscopy technique, object of interest, and the magnification level. The microscopy techniques and objects of interest are represented as binary vectors and the magnification level as a scalar value. Moreover, an additional feature representing if the magnification level is known for a dataset is included. After binary encoding, *AIxCell* utilizes ten user-defined meta-features for each dataset.

Optical Meta-Features For image-based AutoML-systems, we consider information on the optical appearance of images as important to find precisely matching pipeline configurations. Radiomics are data-characterization algorithms aiming to extract a standardized set of features from images (Lambin et al., 2017). These features consist of first-order statistics, shape descriptors, and textural descriptors. This way of quantitative image characterization has shown great promise in clinical decision support systems to improve diagnostic, prognostic, and predictive accuracy (Lambin et al., 2017; Gitto et al., 2021). Specifically, we utilize the Python implementation *pyradiomics* (van Griethuysen et al., 2017) to extract 95 radiomics features per class. Classes are defined by ROI which are annotated as masks for all images in the dataset. For each ROI, a set of 95 features is extracted. We average these feature vectors obtaining one 95-dimensional radiomics vector for each image. Averaging all image-level vectors furthermore leads to one radiomics vector for each dataset. Since these radiomics features have different scales, we perform feature-wise normalization. This is done in two ways: once over all radiomics image vectors and once over all radiomics dataset vectors.

To illustrate this radiomics feature extraction, assume a dataset with 20 images, each with three ROI represented by image masks. This leads to three radiomics vectors x per image; these are then averaged to obtain one radiomics vector per image $\mu[x]$. Feature-wise normalization is first

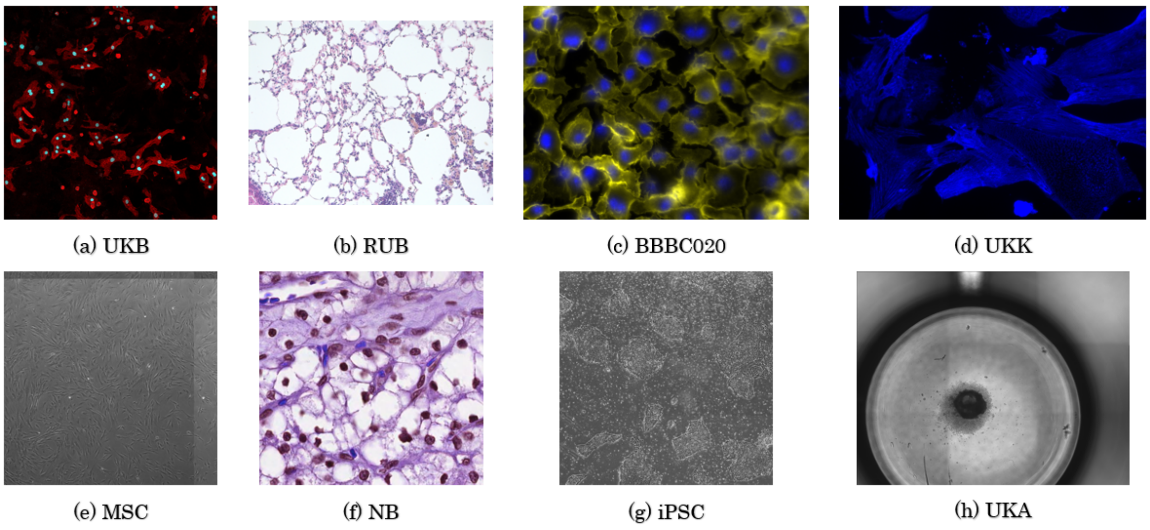


Figure 3: Example images from each of the datasets used to create the Meta-Base.

applied over the 20 radiomics vectors $\mu[x]$ in the dataset, and the resulting vectors are averaged to obtain a normalized dataset descriptor $\hat{y}$. Due to differences between datasets, feature-wise normalization is then applied again over all dataset descriptors $\hat{y}_i$ in the *Meta-Base*. Therefore, the final radiomics representation of a dataset is $y' = \frac{\hat{y}-\mu_j}{\sigma_j}$, where μ_j is the average of the feature in $\hat{y}$ over all datasets and σ_j is the standard deviation of the feature in $\hat{y}$ over all datasets.

There are multiple alternatives to represent radiomics dataset vectors. In our experiments, we evaluate the following: PCA to compress the vector to four dimensions that still capture 94.45% of the total variance between datasets; cosine similarity between the dataset radiomics vectors and using the entire 95-dimensional vector.

Feature Engineering All meta-features described above, are constant for each dataset. However, all the within-dataset variance comes from the varying hyperparameters. To create more meaningful meta-features, we combine dataset meta-features and hyperparameter settings. Table 2 includes the calculations for the engineered handcrafted meta-features.

We also combine the optical meta-features with the hyperparameters, to capture potential interaction effects between the two. Either the first four principal components or the eight similarity scores are multiplied by the learning rate, backbone, and the number of classes. In this way, an additional four respectively eight meta-features are created.

3.5 Meta-Base

As stated in Section 1, in meta-learning-based AutoML-systems, the goal is to leverage previous experience to identify the best-performing pipeline configuration for a new dataset. In general, this experience comprises (dataset-configuration-performance)-triplets from configurations evaluated on various datasets. We call this portfolio of experiences the *Meta-Base*. To construct the *Meta-Base*, a diverse set of datasets needs to be included to ensure transferability and applicability to unseen datasets. In *AIxCell*, the application domain is limited to enable successful meta-learning with a small number of datasets that still cover the application domain.

The *Meta-Base* consists of pipeline configurations, sampled without replacement from the search space, trained on eight different datasets. Entries of the *Meta-Base* are (dataset-configuration-performance)-triplets comprising meta-features, pipeline configuration, and the validation set IoU-score.

The datasets used to construct the *Meta-Base* include images obtained using three different microscopy techniques (Brightfield, Fluorescence, and Phase Contrast) and contain different segmentation objects. An example image for each dataset is given in Figure 3 and an overview of the characteristics of each dataset is given in Table 3. The *RUB* (Leyendecker et al., 2025), *UKK*, *UKA*, and *UKB* datasets were provided by university clinics. The iPSC and MSC datasets stem from our own research at Fraunhofer IPT. Finally, *NB* (Kumar et al., 2017) and *BBBC020* (Ljosa et al., 2012) are publicly available datasets.

Further publicly available dataset obtained using Electron Microscopy are described in Aswath et al. (2023). Additionally, Shi et al. (2025) provide a large scale dataset for cellular segmentation in images acquired using different microscopy techniques. These are not used in the presented study but can be used to validate our approach.

Since there are eight datasets and 576 possible configurations, there are a total of 4,608 possible data points. To evaluate how well pipeline configurations perform after training, we evaluate them on a hold-out validation

Table 3: Main characteristics of the datasets used to create the Meta-Base. The dimensions are averaged over all images in a dataset and the ROI refers to what the region that biomedical experts aimed at analyzing.

Dataset	Dimensions	Instances	Classes	ROI	Microscopy Technique
RUB	968x1292	72	2	Tissue	Brightfield
UKA	2657x3395	71	4	Embroid Bodies	Phase Contrast
UKB	1024x1024	32	4	Nuclei	Fluorescence
UKK	957x1414	229	1	Sarcomeres	Fluorescence
iPSC	2666x2666	40	6	Stem Cell	Phase Contrast
MSC	2052x2052	202	3	Stem Cell	Phase Contrast
BBBC020	1040x1388	20	2	Nuclei	Fluorescence
NB	1000x1000	120	1	Nuclei	Fluorescence

set. During each training run, the datasets are randomly divided into 80% training and 20% validation images. In this way, the train validation split is different each time. To limit training times and still ensure convergence, we implement early stopping. We train the configurations until the validation set IoU-score changes no more than 0.05 for eight consecutive epochs. We define a comparatively large performance delta of 0.05, because on some datasets, we encountered high variability in validation IoU-score between different epochs. To construct the *Meta-Base*, a server with an A100 SXM4 40 GB GPU is used to train and evaluate pipeline configurations.

3.6 Meta-Learning

The core component of *AIxCell* is its meta-learning pipeline. A (meta) ML-model is trained on the observations from the *Meta-Base* and then used to predict a ranking of configurations for a new dataset according to their predicted IoU-score. The input to the ML-model is the hyperparameter vector concatenated with a meta-feature vector and the output is a ranking of configurations.

In order to find the best meta-learning pipeline, a suitable evaluation metric is required. This metric needs to capture the main objective of the meta-model which is to limit the search space toward good configurations. Therefore, the difference between a random selection and using the meta-model to limit the search space is assessed using the z-score. Accordingly, we define the z-score to be:

$$z = \frac{\mu(\hat{y}) - \mu(y_{all})}{\sigma(y_{all})}$$

where $\hat{y}$ are the IoU scores of the predicted top 10% configurations and y_{all} are the IoU scores of all configurations. The larger the z-score, the better the selection from the meta-model is in comparison to a purely random selection.

To allow for further analyses, we report the Pearson correlation between the predicted and the true ranking, the RMSE for pointwise ranking approaches, and the difference in the validation set IoU-score between $\mu(\hat{y})$ and $\mu(y_{all})$.

To output a ranking of configurations, we compare two different approaches. First, by using regression models to directly predict IoU scores and second, the learning-to-rank algorithm LamdaMART with a pairwise loss function. We evaluate the following regression models: Linear Regression, Decision Tree, SVM, Random Forest, and XGBoost.

3.7 Multi-Fidelity Approach

Evaluating configurations on image data is computationally expensive and the number of configurations that can be fully evaluated is therefore limited. This makes AutoML for images considerably more challenging than for tabular data. A common approach that human experts use to combat this problem is to evaluate configurations on a subset of the data or some otherwise reduced version of it. Multi-fidelity methods refer to algorithms that formalize these manual heuristics, using so-called low fidelities to approximate the actual performance of configurations (Hutter et al., 2019).

For *AIxCell*, we utilize successive halving, a multi-fidelity method that allows for training and comparing the predicted best-performing configurations in a given time frame.

In successive halving, all configurations are first evaluated on some initial evaluation budget, then the worst performing half of all active configurations is dropped and the evaluation budget is doubled. This procedure is then repeated until only one active configuration is left. Since the user defines time resources for training a DL-pipeline, we set this training time as the available budget for successive halving. Given the user-defined training time, the budget for each step is derived. After each training step, the configurations are evaluated on a validation set to determine their performance. The successive halving process is started with the top n out of N predicted configurations from the meta-model.

Table 4: Overview of all Meta-Feature representations investigated for the optimal meta-learning pipeline. On the left side the possible sets of meta-features, described in section 3.4, are presented. To the right the different evaluated combinations of these are shown, in total 12 different meta-feature representations are evaluated.

Meta-Feature Representation	1	2	3	4	5	6	7	8	9	10	11	12
Handcrafted Meta-Features	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	
Engineered Handcrafted		✓	✓	✓	✓			✓	✓		✓	
User-given										✓	✓	
4 PCA's Radiomics Vectors				✓							✓	
Radiomics Cosine Similarity			✓				✓				✓	
Engineered Radiomics PCA						✓		✓			✓	
Engineered Radiomics Similarity									✓		✓	
95-dim Radiomics Vectors												✓

4. Experiments

This section details the experiments performed to find the optimal meta-learning pipeline and the evaluation of *AIxCell*.

4.1 Meta-Learning Experiments

The performance of the AutoML-system and thus the user benefit depend primarily on the ability of the meta-learning pipeline to identify high-performing DL-pipeline configurations. Therefore, we evaluate different meta-features and meta-model combinations.

For each meta-model, we evaluate twelve different meta-feature representations (see Table 4). These twelve representations do not cover all possible meta-feature combinations. Instead, they represent a subset of selected combinations that allow to investigate the effects of each set of meta-features. In total, we evaluate $8 * 12 = 76$ meta-learning pipelines.

To find the best one out of these meta-learning pipelines, we utilize leave-one-dataset-out cross-validation. In this way, the meta-learning pipeline is trained on seven datasets from the meta-base and evaluated on the eighth dataset. Each dataset is the validation set once. The performance of a meta-model and meta-feature combination is then averaged across all validation datasets. For achieving higher statistical significance, we evaluate each meta-pipeline across 30 different random seeds.

4.2 AutoML-System Evaluations

After identifying the best-performing meta-learning pipeline, we utilize this pipeline to evaluate the performance of *AIxCell*. To reiterate, the goal of *AIxCell* is to allow biomedical experts to automate their analysis tasks with DL-pipelines optimized for their individual use cases. To this end, we compare *AIxCell* to a baseline DL-pipeline from the *AIxCell* search space and NN-UNet (Isensee et al., 2021) a state-of-the-art image segmentation baseline. For NN-UNet we use the default plans but limit the VRAM to 7GB. Furthermore, we also compare *AIxCell* to the individually and by DL-experts developed DL-pipelines for each use-case. All experiments were carried out on a PC with 16GB of RAM, an Intel i7 Processor, and an RTX 3060 GPU with 8GB of VRAM.

To evaluate *AIxCell*, we define the number of best-ranked configurations taking part in successive halving as $n = 50$ out of N. Since the number of configurations is halved each time, six steps are needed to identify the best pipeline configuration. Based on the initial number of configurations and the total number of steps, we know that 27.38% of the total budget is used to train the best configuration. We then choose the total budget for each dataset based on their mean image dimensions and total number of instances.

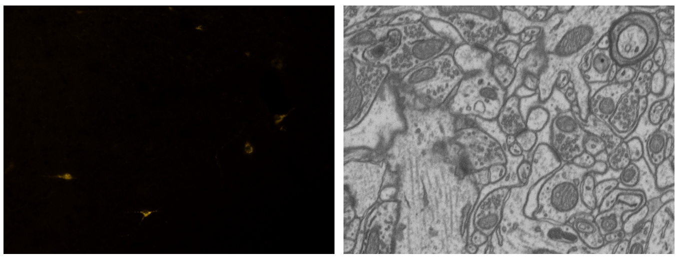

(a) Fluocells (b) Electron Microscopy EPFL

Figure 4: Example Images of the two publicly available datasets used to evaluate *AIxCell*

The first open-source dataset is *Fluocells* (Clissa et al., 2021) with 283 images of neuronal cells. The images obtained using fluorescent microscopy have varying dimensions with a mean dimension of 1600x1200 pixels. The task is binary segmentation, separating neuronal cells and background. See Figure 4a for an example image. We split the dataset into 80 % training, 10 % validation, and 10 % test data. Due to the large number of instances and relatively large image dimensions of *Fluocells*, we assign a compute budget of 6 hours. As a result, within successive halving, the final configuration is trained for 1.64 hours.

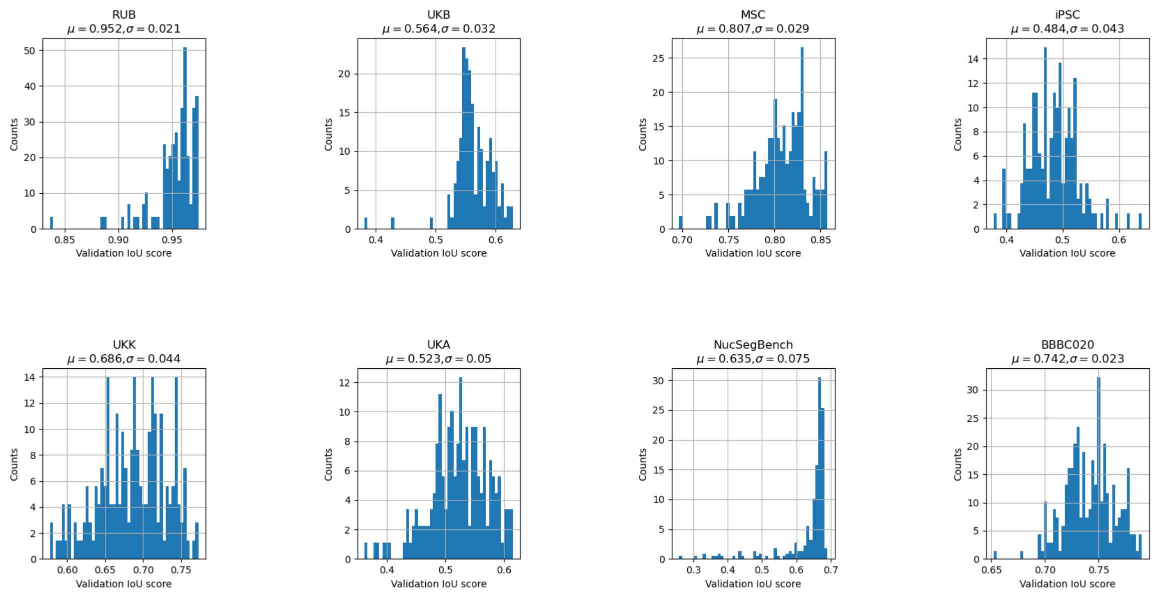


Figure 5: Distribution of validation set IoU scores on each dataset. Scores were divided into 50 bins for visualization.

The second public dataset we use is the Electron Microscopy Dataset from EPFL (Lucchi et al., 2013). See Figure 4b for an example image. This dataset offers a predefined split into training and testing data. The training set consists of 165 images of mitochondria, with dimension of 1024x768 pixels. The image recognition task is to semantically segment the mitochondria. The testing set also contains 165 images of same dimensions. For performance evaluation, the training set is split into 80 % training, 10 % validation, and 10 % testing data.

Subsequently, both *UKB* and *iPSC* datasets are used to finally evaluate *AIxCell*. To this end, the meta-model is only trained on the meta-data of the seven other datasets. In analogy to the other datasets, a train, validation, and testing split of 80 %, 10 %, and 10 % is used. Since *UKB* is relatively small, we provide a compute budget of 4 hours to train the top configuration for 1.09 hours. *iPSC* contains 40 images of comparatively high dimensions. Given that the dataset comprises six different classes, it is reasonable to assume that longer training times might be needed. We therefore provide an increased compute budget of 6 hours for this dataset.

Finally, after completing the training and utilizing the final pipeline for inference, we compare the results to manual analysis results of a human expert and to those of a default pipeline configuration. Based on experience, we define the hyperparameters for the default pipeline configuration to be: batch size = 16, patch size = 128, learning rate = 0.001 and augmentations = Flip. To eliminate performance differences induced by the dataset split, the default pipeline and the best determined pipeline are evaluated using an identical split of 80 % training and 20 % testing data.

5. Results

In this section, we present the results from the individual experiments regarding the construction of the meta-base, radiomics features for optical appearance, meta-learning and the entire system are presented. Finally, we present the validation results of the overall system.

5.1 Meta-Base

A total of 1,458 different configurations were trained on various datasets, resulting in 1,423 data points in the meta-base after removing 35 outlier configurations based on training times and validation set IoU-scores. This accounts for 30.8% of the possible data points in the design space. The distributions of validation set IoU-scores, excluding outliers, are displayed in Figure 5. The majority of the datasets show a normal distribution of scores, but the distributions of *RUB* and *NB* are skewed towards favorable configurations. The average standard deviation for validation set IoU-score is 0.039, while the within-dataset standard deviation varies among the datasets.

To evaluate the potential of radiomics in converting images into informative vectors, the 95-dimensional image vectors were compressed into two dimensions using t-distributed stochastic neighbor embedding (T-SNE) Hinton and Roweis (2002). First, the resulting visualization (see Figure 6) clearly shows that images of a specific dataset are clustered in the neighborhood of the images from the exact same dataset. Second, the results reveal that the clusters belonging to the respective datasets are distributed in the cluster space. The proximity or distance of these clusters appears plausible when compared with the images: visually similar datasets are closer to each other in the cluster space

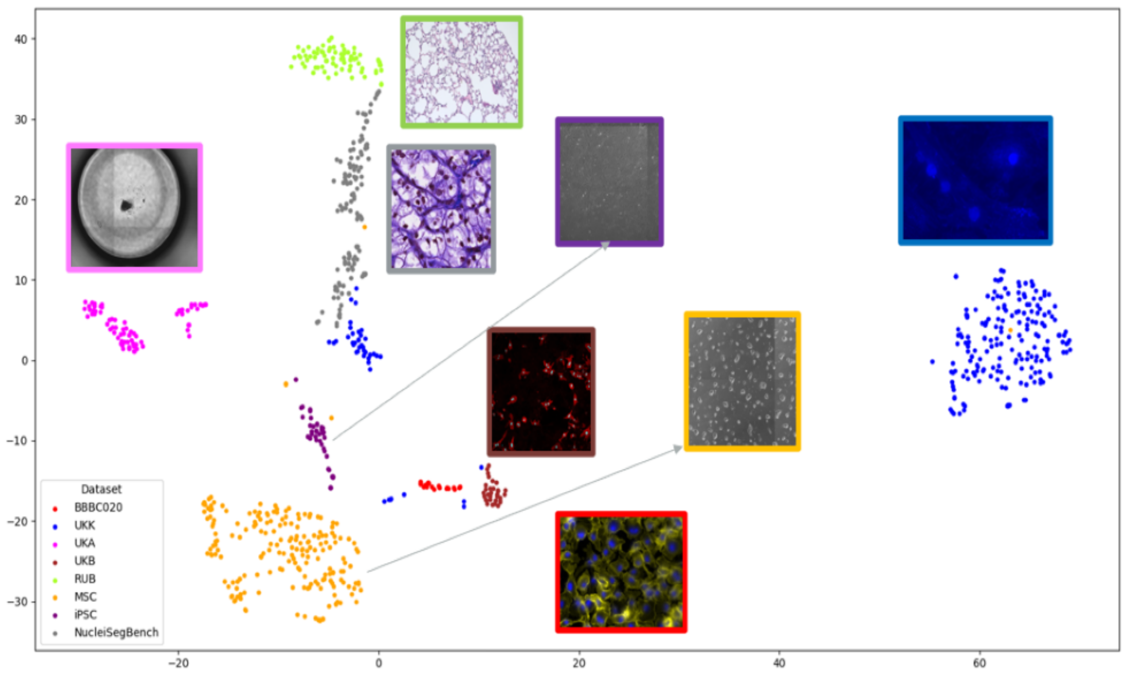


Figure 6: Radiomics vectors of all images of the meta-base visualized in two dimensions. The colors represent the dataset the vectors belong to, examples images for each dataset are also shown. Before applying T-SNE to the images, feature-wise normalization was applied.

in comparison with visually dissimilar datasets. For example, *MSC* and *iPSC* datasets that both contain images of stem cells obtained through phase-contrast microscopy, cluster closely together. The inherent limitations of T-SNE to preserve global distance information, limit the conclusions that can be drawn from this analysis. Therefore, we also analyze the cosine similarities between dataset vectors.

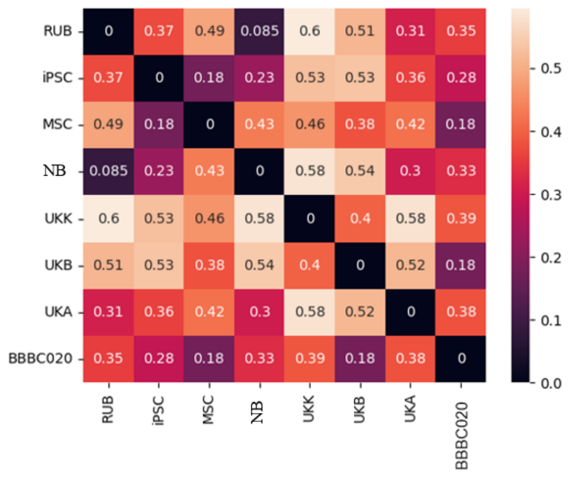


Figure 7: Cosine distance scores between the dataset level radiomics vectors. A value of zero indicates vector directions to be identical and a value of one indicates orthogonal vectors.

5.2 Meta-Features for Optical Appearance

A similar pattern emerges when examining the cosine distance scores between dataset-level radiomics vectors, as illustrated in Figure 7. Datasets that share visual similarities exhibit lower cosine distance scores, while those with different visual characteristics have higher scores. For instance, *RUB* and *NB*, being visually similar, exhibit the lowest cosine distance score of 0.085. Conversely, *UKK* and *RUB*, being visually distinct, display the highest cosine distance score.

5.3 Meta-Features

We use the performance of the entire meta-learning pipeline to assess the importance of the different meta-feature sets. As seen in the next section, the most performant model consistently uses the handcrafted meta-features and combinations of these with the hyperparameter space. Other meta-features, such as user-defined or optical-appearance are not used by the best models. We further validated meta-feature importance using permutation with the best meta-model, this analysis confirmed the results of the meta-pipeline comparisons.

5.4 Meta-Learning

The best five configurations measured by z-score are shown in Table 5 (for a definition of the metrics please refer to Section 3.6). The best performance is obtained using a

Table 5: The top 5 combinations of meta-features and meta-model, measured by z-score. The number in the Meta-Features column refers to the representation in Table 4.

Meta-Model	Meta-Features	Z-score	Pearson	Diff IoU	RMSE
Random Forest	2	0.643	0.295	0.028	0.138
LamdaMART	1	0.608	0.299	0.030	-
LamdaMART	3	0.606	0.291	0.028	-
Random Forest	3	0.583	0.308	0.030	0.157
Random Forest	1	0.582	0.242	0.030	0.153

Table 6: The best configurations found by the AutoML-system for each evaluated dataset.

Dataset	Patch Size	Batch Size	Learning Rate	Backbone	Augmentations
Fluocells	256	16	0.001	ResNet-50	None
EM EPFL	256	8	0.0005	ResNet-18	None
UKB	128	4	0.0005	ResNet-18	Flip
iPSC	256	8	0.0005	ResNet-18	None

random forest regressor, trained to predict IoU-scores based on handcrafted and engineered handcrafted meta-features. A selection based on the top 10% predicted configurations by this meta-pipeline is 0.643 standard deviations better than a random selection. When interpreting these results, it has to be considered that the best possible z-score equals 1.41.

Looking at the metrics, it becomes apparent that the z-score correlates with the Pearson correlation, difference in IoU-score and RMSE-score. While the largest z-score is not always associated with the best score on these metrics, configurations with large z-scores always have one of the best scores on these metrics as well.

Overall, the metrics do not indicate good performance by the meta-model, however, this illustrates the difficulty of evaluating a meta-model in this context. Many of the configurations only perform marginally different on some datasets and it is therefore not important to accurately rank these. For the RMSE, there are large differences in the average IoU-score between datasets (see Figure 5). Due to the small number of datasets in the meta-base, it is infeasible for the model to accurately estimate the mean performance on a given dataset. This leads to relatively large RMSE. Nonetheless, this is not necessarily relevant to evaluate the meta-model since it is of higher importance that the model is able to differentiate between bad and good configurations.

For the best performing meta-pipeline, we evaluated different maximum tree-depths for the random forest meta-model. This is motivated to limit the capacity of the model and to thereby reduce the risk of overfitting. The best maximum tree depth was found to be 7. Training the meta-pipeline again with the maximum tree-depth set to 7, improved the results. Averaged over 20 random seeds, this results in an average z-score of **0.688**, an average RMSE of **0.136**, an average Pearson correlation of **0.291** and a difference in IoU of **0.262**. Depending on the random seed. the obtained z-scores ranged from 0.64 to 0.74, with a standard deviation of 0.045. The performance of this meta-pipeline on each of the validation datasets is shown in Figure 8.

Obviously, the model performs similarly well across all datasets even though the z-scores vary. This shows that the z-scores always have to be assessed in the context of the specific dataset and are not a suitable measure to compare model performance between two datasets. The performance plots reveal that on most datasets the model does well at predicting bad configurations. However, the variance increases in predicting the best configuration. This is especially the case for *BBBC020*, *MSC*, and *UKB*, where the meta-model predicts low IoU scores for some of the best true configurations. On *MSC*, one of the true top 10% configurations is predicted to be the worst configuration. On all datasets except for *UKB*, there is some overlap between the predicted top 10% and true top 10% (green points). Overall, the model performs worst on *UKB*. The best performance is achieved on the *RUB*. Almost all points in the true and predicted top 10% overlap.

Table 6 shows the best configurations found by *AIxCell*. Oftentimes, these configurations comprise large patch sizes, low learning rates and no augmentations steps.

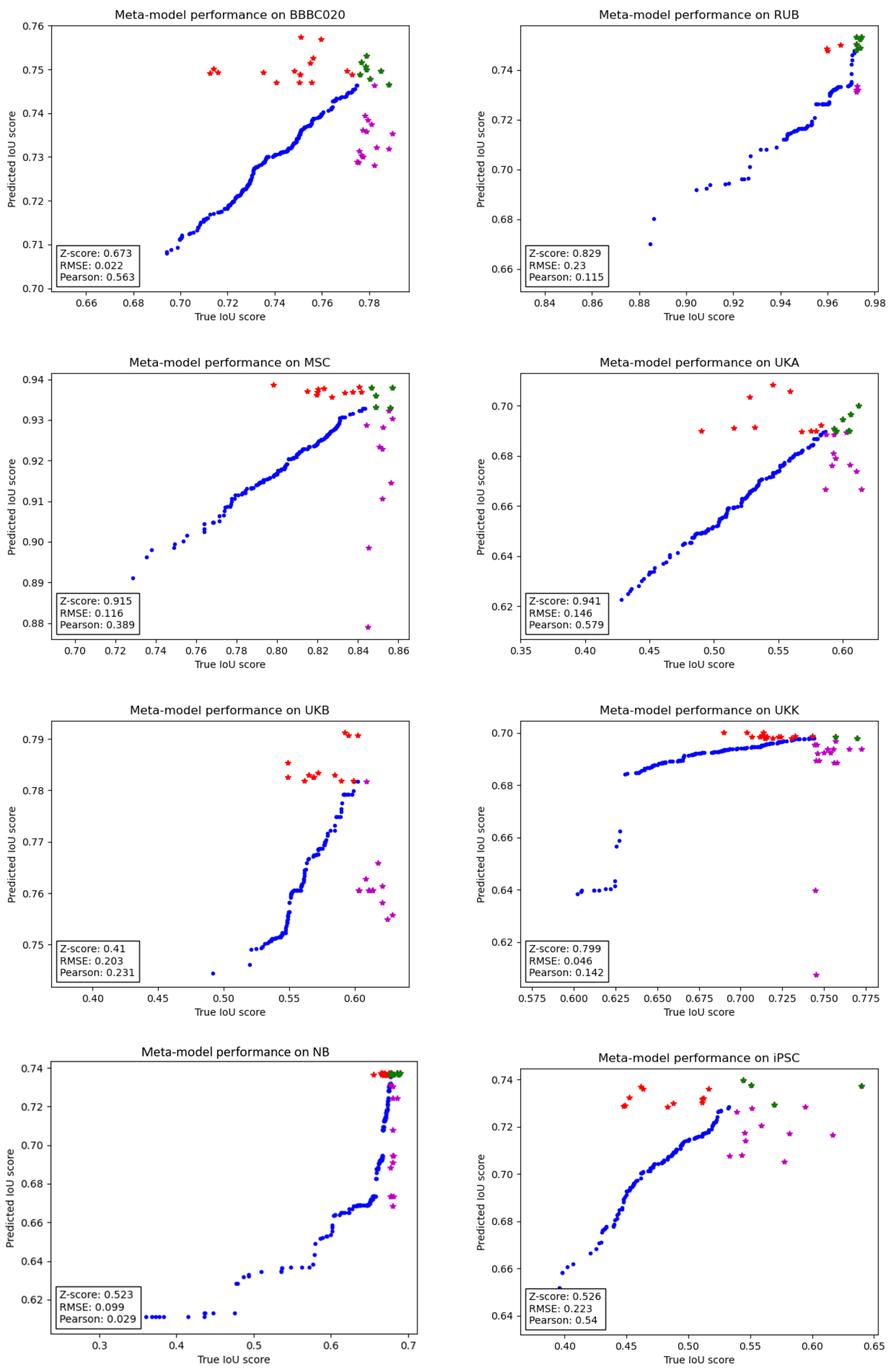
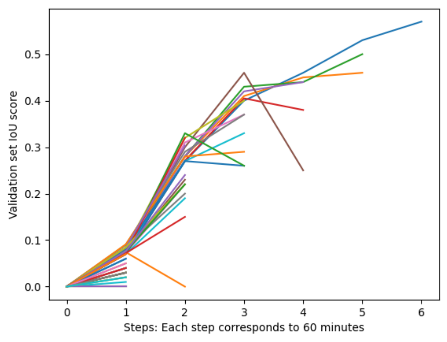
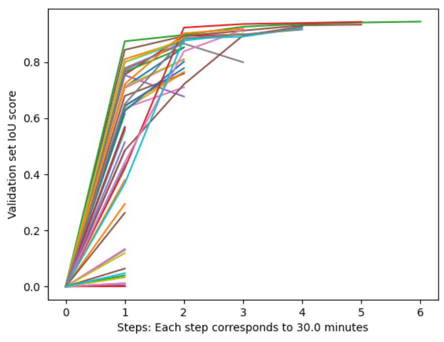


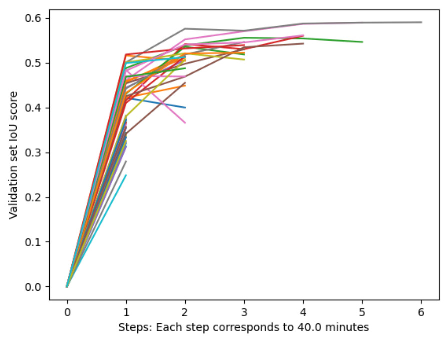
Figure 8: Validation set performance of the best meta-model and feature engineering strategy (random forest, handcrafted and engineered handcrafted) on all datasets. Points are colored blue by default, red is used for the top 10% of predicted configurations, magenta for the true top 10% and green for configurations that are both in the predicted and true top 10%.



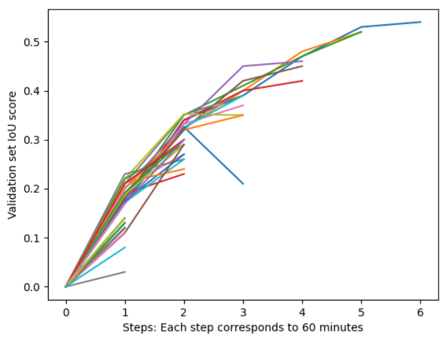
(a) Fluocells



(b) Electron Microscopy EPFL



(c) UKB



(d) iPSC

Figure 9: The best 50 predicted configurations evaluated on the given datasets using successive halving. At each step, the worst performing half of configurations is dropped.

Table 7: Comparison of the baseline pipeline, *AIxCell*, *NN-UNet* and SOTA solutions developed by human experts. In the identical splits column the same train/test split was used for the pipelines and in the unknown splits column the data was split randomly. An exception of this is EM EPFL where a pre-defined split is provided.

	Identical Splits				Unknown Splits		
Dataset	Baseline	*AIxCell*	NN-UNet	SOTA	SOTA	*AIxCell*	Metric
Fluocells	0.6438	0.6522	0.5691	-	0.8149	0.7621	F1-Score
EM EPFL	0.8021	0.7832	0.8421	0.8672	-	-	IoU
UKB	0.6017	0.6201	0.6072	0.6212	0.64	0.60	IoU
iPSC	0.5415	0.5720	0.5530	0.5824	0.7344	0.6431	IoU

5.5 Evaluation of the Meta-Learning-Based and Domain-Specific AutoML-System

Using the outlined best meta-pipeline, a random forest model limited to max tree depth value of 7 and handcrafted and engineered handcrafted meta-features, the overall AutoML-system was evaluated as described in section 4.2.

In Figure 9, the training progressions of the $n = 50$ best predicted configurations are illustrated. The plots reveal that the training time needed to achieve good performance varies across datasets. On the one hand, for *UKB* and *EM-EPFL*, some configurations attain performances close to the optimal configuration after only one successive halving step. On the other hand, for *Fluocells* and *iPSC* datasets, the performances increase significantly at each step and the best configuration might have further improved given a longer training budget. Nonetheless, the plots indicate that a close to optimal configuration is found for each dataset.

Table 7 shows the overall results. The solutions in the SOTA column refer to segmentation solutions developed by human experts. For *Fluocells*, this pipeline uses a patch size of 512 and six image augmentations which notably include a cropping approach for over-sampling. The model used is a ResUnet adopted to allow for a larger field of view (Clissa et al., 2021). The solution for *EM-EPFL* uses a working set based approximate subgradient descent method for learning graphical models for structured prediction (Lucchi et al., 2013). According to Leyendecker et al. (2022), for *UKB*, the reference solution relies on a feature-pyramid network (Lin et al., 2017) with a VGG16 backbone, a patch size of 256, a batch size of 4, and categorical cross-entropy loss. The best performance on *iPSC* dataset was achieved by a pyramid scene parsing network (Zhao et al., 2016) and a patch size of 384.

For all datasets without a pre-defined split, the system outperforms the static baseline configuration and NN-UNet. We note the performance differences between the identical splits and the unknown data splits, indicating that performance greatly depends on the data split. As expected, the individually developed and tuned SOTA solutions outperform *AIxCell* on all datasets. NN-UNet performs competitively on all datasets and outperforms *AIxCell* on EM EPFL.

6. Discussion

We showcase *AIxCell*, a domain-specific meta-learning-based AutoML-system for cellular image analyses. Our analyses of the meta-learning system shows that, even with a limited number of datasets and data points, the meta-model can predict good configurations for a new dataset. Leveraging this meta-model and successive halving, *AIxCell* always outperforms a baseline and *NN-UNet* a generic HPO optimization system for image segmentation. In this way *AIxCell* enables biomedical experts to construct, train, and utilize optimized DL-pipelines for their laboratory-specific use cases.

6.1 Meta-Learning

Analyzing the influence of different meta-feature sets on pipeline performance shows that handcrafted meta-features and combinations of these with the hyperparameter space are assigned the highest feature importance by the most performant meta-models. On the other hand, our results show that the optical appearance of images represented using radiomics is not considered of high importance by the best-performing meta-learning pipelines. They also do not utilize user-defined meta-features about the biomedical analyses task.

The main reason for the model's preference to a small set of the most important meta-features could be the limited size of the meta-base. This decreases the number of features that can be optimally used by the model (Hua et al., 2005; Figueroa et al., 2012). Furthermore, the meta-features are not independent since they are always the same for a dataset. This increases the correlation between meta-features which further limits the number of optimal features.

For the optical meta-features, our qualitative evaluation shows their promise in representing datasets. Nonetheless, the relationship between the optical appearance and our hyperparameter space is ambiguous. We suspect that at least the patch size and augmentations depend on the optical appearance. However, the patch size also depends on the statistics of the ROI's which are included in the handcrafted meta-features. The augmentations are rarely proposed by the meta-model, which may result from these depending on the optical appearance and being specific to each dataset. Reducing the radiomics vector to eight (cosine scores) or four (PCA) dimensions could remove the information required to predict the given augmentations. While the optical meta-features are not used by the best meta-pipeline, they are used in the 3^{rd} and 4^{th} best configurations. Combined with the results of our qualitative analysis, this shows the potential of radiomics features to quantify image datasets.

To improve the dataset-specific meta-features, future work could explore different methods to reduce the dimensionality of the radiomics vectors. Moreover, to increase the informational content and therefore the quality of the meta-features, tools such as MetaBU (Rakotoarison et al., 2022) could be used to construct new and better meta-features based on all extracted meta-features. In MetaBU this is done via an Optimal Transport procedure, aligning the manually designed meta-features with the space of distributions on the hyperparameter configurations.

Compared to other meta-learning-based AutoML-systems, such as AutoBagging (Pinto et al., 2017) and RankML (Laadan et al., 2019), *AIxCell's* meta-model learns based on a much smaller meta-base. While the meta-model also performs worse than AutoBagging or RankML, it is still able to identify good configurations for a new dataset and therefore outlining the potential of limiting the domain in meta-learning.

Using successive halving allows *AIxCell* to evaluate n out of N pipelines using the same time budget that would be required to perform around five full evaluations. The results indicate that for each dataset the best pipeline is trained until convergence (see Figure 9). However, we note that the time budget required for each dataset varies and that for *Fluocells* and *iPSC*, a larger budget might have improved final performance. This time budget is a parameter that needs to be chosen by the biomedical expert, however, even when choosing the budget conservatively, *AIxCell* still runs within a reasonable time with regard to laboratory practice.

In summary, our meta-learning experiments demonstrate that despite having a relatively small meta-base, meta-learning effectively guides the search space towards configurations that yield high performance significantly outperforming random selection. Doing so across multiple random seeds further highlights the robustness of our system. This demonstrates the potential of meta-learning in domain-specific scenarios, even with the high computational cost of evaluating individual configurations.

6.2 *AIxCell*

The evaluations of *AIxCell* show that SOTA solutions outperform it, however, it outperforms the baseline solutions and NN-UNet (see Table 7). Outperforming NN-UNet (Isensee et al., 2021), a generic heuristic-based system, on three datasets demonstrates the potential of a domain-specific system. We note that NN-UNet might outperform *AIxCell* if it is run on a larger GPU since it could then use larger patch and batch sizes. Nonetheless, both were run on similar GPUs so it is a fair comparison.

As described, the SOTA solutions entail complexity, with human experts experimenting with various configurations and performing dataset specific adaptations. Given the search space and time constraints of *AIxCell*, it is to be expected that it performs worse. Nonetheless, since *AIxCell* outperforms baseline solutions it provides biomedical experts the benefits of dataset-specific hyperparameter tuning. While the improvements over the baseline solutions are small, taking into context the meta-base statistics (see Figure 5) shows that the best configurations found by *AIxCell* are in the top quintile of all sampled configurations.

A noteworthy observation pertains to the performance variations between identical and unknown data splits. This indicates that in cellular image segmentation, the performance is significantly influenced by the data split. Using a stratified splitting, potentially also taking into account visual features, might lead to more uniform results.

In comparison to the state-of-the-art of AutoML for image data and automatic methods for cellular segmentation, *AIxCell*, presents a novel approach by using meta-learning and focusing on a specific domain. The benefit of this approach is that it greatly reduces the compute required to find an optimal configuration for a novel dataset. A downside is that *AIxCell* cannot compete with SOTA solutions developed by human experts. Nonetheless, the results show reliable improvement over baseline solutions and therefore the benefits of task-specific configuration tuning.

Integrating these task-specific adaptations could improve the performance of systems relying on a generalist segmentation solution such as CellProfiler (Jones et al., 2008) or Ilastik (Berg et al., 2019). Compared to T-AutoML (Yang et al., 2021) *AIxCell* uses a similar methodology but requires significantly less compute for the optimization process. Nonetheless, T-AutoML considers a larger search space and, in principle, works for all semantic segmentation tasks.

7. Conclusion

Motivated by the current practice in biomedical laboratories of manually analyzing increasing amounts of microscopy images, we propose *AIxCell*, a domain-specific meta-learning-based AutoML-system for analyzing cellular image data. With *AIxCell* and its four-stage AutoML-system architecture, we aim to provide a tool allowing biomedical experts to automate the otherwise difficult task of developing entire DL-based image processing pipelines. To meet the high requirements of biomedical image analyses, we propose a domain-specific approach. According to that and aiming to balance the generality-specificity trade-off, *AIxCell* is trained on a portfolio of cellular and tissue image datasets with varying analysis tasks. We utilize a portfolio-based meta-learning-based approach for predicting top-performing pipeline configurations for a given dataset. Subsequently, *AIxCell* utilizes a multi-fidelity approach in the form of successive halving according to which n out of N top-performing pipelines are trained under a fixed time budget until the best-performing DL-pipeline is identified. We perform a throughout evaluation of the meta-learning setup and evaluate *AIxCell* on four datasets. Our results show that meta-learning is suitable to limit the search space towards good-performing configurations, with handcrafted meta-features being the most important. *AIxCell* always outperforms a baseline DL-pipeline and outperforms NN-UNet, a generic solution, on three out of four datasets. This underscores the benefit of dataset-specific DL-pipelines and the domain-specific approach. However, by automating the development process of DL-pipelines for biomedical image analysis and by trading-off specificity for generality, *AIxCell* faces high computational costs in comparison with manually developed pipelines. Limitations of *AIxCell* include its static but growing post-processing library that requires the implementation of custom post-processing functions for solving novel analysis tasks. For training a new primary pipeline on a novel task, *AIxCell* still relies on manual labeling and its performance is dependent on label quality. Furthermore, *AIxCell's* ability to propose a suitable pipeline for a novel task depends on the task's representation within the meta-data. However, with increasing diversity of the meta-data, we consider that the performance of *AIxCell* will further increase across a growing variety of biomedical analysis tasks. In summary, we conclude that limiting the application domain and utilizing meta-learning within this domain, is a promising approach in designing AutoML-systems for image data. In the application area of biomedical microscopic image analyses, our system provides biomedical experts with the benefits of dataset-specific DL-pipelines, without requiring knowledge of deep learning.

Future research in meta-learning-based image AutoML-systems could explore more effective meta-features, particularly emphasizing optical appearance to potentially better predict dataset-centric adaptations, such as image augmentations. Including more data-centric adaptations could yield larger gains for dataset-specific optimization. Also, further research should investigate the sensitivity of segmentation algorithms with respect to the size and shape of cell structures and the resulting generalizability of the models. Additionally, examining different multi-fidelity methods, including early stopping, could further enhance the efficiency and effectiveness of these systems.

Acknowledgments

The publication was written within the research project AIxCell (IGF 21361 N). The project of the Research Association for Precision Mechanics, Optics and Medical Technology is funded by the Federal Ministry for Economic Affairs and Energy via the AiF within the context of the programme for the promotion of Industrial Cooperative Research (IGF) based on a resolution of the German Bundestag.

Ethical Standards

The work follows appropriate ethical standards in conducting research and writing the manuscript, following all applicable laws and regulations regarding treatment of animals or human subjects.

Conflicts of Interest

We declare we don't have conflicts of interest.

Data availability

The meta-dataset and some of the cellular image analysis datasets are proprietary and unfortunately not publicly available.

References

Vasileios Anagnostidis, Benjamin Sherlock, Jeremy Metz, Philip Mair, Florian Hollfelder, and Fabrice Gielen. Deep learning guided image-based droplet sorting for on-demand selection and analysis of single cells and 3d cell cultures. *Lab Chip*, 20:889–900, 2020. . URL http://dx.doi.org/10.1039/D0LC00055H.

Anusha Aswath, Ahmad Alsahaf, Ben N.G. Giepmans, and George Azzopardi. Segmentation in large-scale cellular electron microscopy with deep learning: A literature survey. *Medical Image Analysis*, 89:102920, 10 2023. ISSN 1361-8415. . URL https://www.sciencedirect.com/science/article/pii/S1361841523001809.

Stuart Berg, Dominik Kutra, Thorben Kroeger, Christoph N. Straehle, Bernhard X. Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, Kemal Eren, Jaime I Cervantes, Buote Xu, Fynn Beuttenmueller, Adrian Wolny, Chong Zhang, Ullrich Koethe, Fred A. Hamprecht, and Anna Kreshuk. ilastik: interactive machine learning for (bio)image analysis. *Nature Methods*, 16(12):1226–1232, December 2019. ISSN 1548-7091, 1548-7105. . URL http://www.nature.com/articles/s41592-019-0582-9.

Luca Clissa, Roberto Morelli, Fabio Squarcio, Timna Hitrec, Marco Luppi, Lorenzo Rinaldi, Matteo Cerri, Roberto Amici, Stefano Bastianini, Chiara Berteotti, Viviana Lo Martire, Davide Martelli, Alessandra Occhinegro, Domenico Tupone, Giovanna Zoccoli, and Antonio Zoccoli. Fluorescent Neuronal Cells, 2021. URL http://amsacta.unibo.it/6706.

Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural Architecture Search: A Survey. 2018. . URL https://arxiv.org/abs/1808.05377.

Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. AutoGluon-Tabular: Robust and Accurate AutoML for Structured Data. 2020. . URL https://arxiv.org/abs/2003.06505.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Tobias Springenberg, Manuel Blum, and Frank Hutter. Auto-sklearn: Efficient and Robust Automated Machine Learning. In Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors, *Automated Machine Learning*, pages 113–134. Springer International Publishing, Cham, 2019. ISBN 978-3-030-05317-8 978-3-030-05318-5. . URL http://link.springer.com/10.1007/978-3-030-05318-5_6.

Rosa L Figueroa, Qing Zeng-Treitler, Sasikiran Kandula, and Long H Ngo. Predicting sample size required for classification performance. *BMC Medical Informatics and Decision Making*, 12(1): 8, December 2012. ISSN 1472-6947. . URL https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/1472-6947-12-8.

Salvatore Gitto, Renato Cuocolo, Alessio Annovazzi, Vincenzo Anelli, Marzia Acquasanta, Antonino Cincotta, Domenico Albano, Vito Chianca, Virginia Ferraresi, Carmelo Messina, Carmine Zoccali, Elisabetta Armiraglio, Antonina Parafioriti, Rosa Sciuto, Alessandro Luzzati, Roberto Biagini, Massimo Imbriaco, and Luca Maria Sconfienza. CT radiomics-based machine learning classification of atypical cartilaginous tumours and appendicular chondrosarcomas. *EBioMedicine*, 68:103407, June 2021. ISSN 23523964. . URL https://linkinghub.elsevier.com/retrieve/pii/S2352396421002000.

Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002. URL https://proceedings.neurips.cc/paper_files/paper/2002/file/6150ccc6069bea6b5716254057a194ef-Paper.pdf.

J. Hua, Z. Xiong, J. Lowey, E. Suh, and E. R. Dougherty. Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, 21(8): 1509–1515, April 2005. ISSN 1367-4803, 1460-2059. . URL https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/bti171.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors. *Automated Machine Learning: Methods, Systems, Challenges*. The Springer Series on Challenges in Machine Learning. Springer International Publishing, Cham, 2019. ISBN 978-3-030-05317-8 978-3-030-05318-5. . URL http://link.springer.com/10.1007/978-3-030-05318-5.

Fabian Isensee, Paul F. Jaeger, Simon A. A. Kohl, Jens Petersen, and Klaus H. Maier-Hein. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, 18(2): 203–211, February 2021. ISSN 1548-7091, 1548-7105. . URL https://www.nature.com/articles/s41592-020-01008-z.

Haifeng Jin, François Chollet, Qingquan Song, and Xia Hu. AutoKeras: An AutoML Library for Deep Learning. *Journal of Machine Learning Research*, 24(6):1–6, 2023. URL http://jmlr.org/papers/v24/20-1355.html.

Thouis R Jones, In Han Kang, Douglas B Wheeler, Robert A Lindquist, Adam Papallo, David M Sabatini, Polina Golland, and Anne E Carpenter. CellProfiler Analyst: data exploration and analysis software for complex image-based screens. *BMC Bioinformatics*, 9(1):482, December 2008. ISSN 1471-2105. . URL https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-9-482.

Lars Kotthoff, Chris Thornton, Holger H. Hoos, Frank Hutter, and Kevin Leyton-Brown. Auto-WEKA: Automatic Model Selection and Hyperparameter Optimization in WEKA. In Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors, *Automated Machine Learning*, pages 81–95. Springer International Publishing, Cham, 2019. ISBN 978-3-030-05317-8 978-3-030-05318-5. . URL http://link.springer.com/10.1007/978-3-030-05318-5_4.

Neeraj Kumar, Ruchika Verma, Sanuj Sharma, Surabhi Bhargava, Abhishek Vahadane, and Amit Sethi. A Dataset and a Technique for Generalized Nuclear Segmentation for Computational Pathology. *IEEE Transactions on Medical Imaging*, 36(7):1550–1560, July 2017. ISSN 0278-0062, 1558-254X. . URL http://ieeexplore.ieee.org/document/7872382/.

Doron Laadan, Roman Vainshtein, Yarden Curiel, Gilad Katz, and Lior Rokach. RankML: a Meta Learning-Based Approach for Pre-Ranking Machine Learning Pipelines. 2019. . URL https://arxiv.org/abs/1911.00108.

Philippe Lambin, Ralph T.H. Leijenaar, Timo M. Deist, Jurgen Peerlings, Evelyn E.C. de Jong, Janita van Timmeren, Sebastian Sanduleanu, Ruben T.H.M. Larue, Aniek J.G. Even, Arthur Jochems, Yvonka van Wijk, Henry Woodruff, Johan van Soest, Tim Lustberg, Erik Roelofs, Wouter van Elmpt, Andre Dekker, Felix M. Mottaghy, Joachim E. Wildberger, and Sean Walsh. Radiomics: the bridge between medical imaging and personalized medicine. *Nature Reviews Clinical Oncology*, 14 (12):749–762, December 2017. ISSN 1759-4774, 1759-4782. . URL http://www.nature.com/articles/nrclinonc.2017.141.

Trang T Le, Weixuan Fu, and Jason H Moore. Scaling tree-based automated machine learning to biomedical big data with a feature set selector. *Bioinformatics*, 36(1): 250–256, January 2020. ISSN 1367-4803, 1367-4811. . URL https://academic.oup.com/bioinformatics/article/36/1/250/5511404.

Lars Leyendecker, Julius Haas, Tobias Piotrowski, Maik Frye, Cora Becker, Bernd Fleischmann, Michael Hesse, and Robert H. Schmitt. A modular deep learning pipeline for cell culture analysis: Investigating the proliferation of cardiomyocytes, 2022. URL https://publica.fraunhofer.de/handle/publica/430778.

Lars Leyendecker, Anna Weltin, Florian Nienhaus, Michaela Matthey, Bastian Nießing, Daniela Wenzel, and Robert Schmitt. Deep learning based automation of mean linear intercept quantification in copd research. *Frontiers in Big Data*, 8, 06 2025. .

Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid

Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, Honolulu, HI, July 2017. IEEE. ISBN 978-1-5386-0457-1. . URL http://ieeexplore.ieee.org/document/8099589/.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, February 2018. ISSN 0162-8828, 2160-9292, 1939-3539. . URL https://ieeexplore.ieee.org/document/8417976/.

Vebjorn Ljosa, Katherine L Sokolnicki, and Anne E Carpenter. Annotated high-throughput microscopy image sets for validation. *Nature Methods*, 9(7):637–637, July 2012. ISSN 1548-7091, 1548-7105. . URL http://www.nature.com/articles/nmeth.2083.

Aurelien Lucchi, Yunpeng Li, and Pascal Fua. Learning for Structured Prediction Using Approximate Subgradient Descent with Working Sets. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1987–1994, Portland, OR, USA, June 2013. IEEE. ISBN 978-0-7695-4989-7. . URL http://ieeexplore.ieee.org/document/6619103/.

Erick Moen, Dylan Bannon, Takamasa Kudo, William Graf, Markus Covert, and David van Valen. Deep learning for cellular image analysis. *Nature methods*, 16(12): 1233–1246, 2019. .

Ty Nguyen, Tolga Ozaslan, Ian D. Miller, James Keller, Giuseppe Loianno, Camillo J. Taylor, Daniel D. Lee, Vijay Kumar, Joseph H. Harwood, and Jennifer Wozencraft. U-Net for MAV-based Penstock Inspection: an Investigation of Focal Loss in Multi-class Segmentation for Corrosion Identification. 2018. . URL https://arxiv.org/abs/1809.06576.

Randal S. Olson, William La Cava, Zairah Mustahsan, Akshay Varik, and Jason H. Moore. Data-driven Advice for Applying Machine Learning to Bioinformatics Problems. 2017. . URL https://arxiv.org/abs/1708.05070.

Fábio Pinto, Carlos Soares, and João Mendes-Moreira. Towards Automatic Generation of Metafeatures. In James Bailey, Latifur Khan, Takashi Washio, Gill Dobbie, Joshua Zhexue Huang, and Ruili Wang, editors, *Advances in Knowledge Discovery and Data Mining*, volume 9651, pages 215–226. Springer International Publishing, Cham, 2016. ISBN 978-3-319-31752-6 978-3-319-31753-3. . URL http://link.springer.com/10.1007/978-3-319-31753-3_18.

Fábio Pinto, Vítor Cerqueira, Carlos Soares, and João Mendes-Moreira. autoBagging: Learning to Rank Bagging Workflows with Metalearning. 2017. . URL https://arxiv.org/abs/1706.09367.

Herilalaina Rakotoarison, Louisot Milijaona, Andry Rasoanaivo, Michèle Sebag, and Marc Schoenauer. Learning Meta-features for AutoML. April 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, volume 9351, pages 234–241. Springer International Publishing, Cham, 2015. ISBN 978-3-319-24573-7 978-3-319-24574-4. . URL http://link.springer.com/10.1007/978-3-319-24574-4_28.

Can Shi, Jinghong Fan, Zhonghan Deng, Huanlin Liu, Qiang Kang, Yumei Li, Jing Guo, Jingwen Wang, Jinjiang Gong, Sha Liao, Ao Chen, Ying Zhang, and Mei Li. Cellbindb: a large-scale multimodal annotated dataset for cell segmentation with benchmarking of universal models. *GigaScience*, 14:giaf069, 06 2025. ISSN 2047-217X. . URL https://doi.org/10.1093/gigascience/giaf069.

Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature Methods*, 18(1):100–106, January 2021. ISSN 1548-7091, 1548-7105. . URL http://www.nature.com/articles/s41592-020-01018-x.

Joost J.M. van Griethuysen, Andriy Fedorov, Chintan Parmar, Ahmed Hosny, Nicole Aucoin, Vivek Narayan, Regina G.H. Beets-Tan, Jean-Christophe Fillion-Robin, Steve Pieper, and Hugo J.W.L. Aerts. Computational Radiomics System to Decode the Radiographic Phenotype. *Cancer Research*, 77(21):e104–e107, November 2017. ISSN 0008-5472, 1538-7445. . URL https://aacrjournals.org/cancerres/article/77/21/e104/662617/Computational-Radiomics-System-to-Decode-the.

David A. van Valen, Takamasa Kudo, Keara M. Lane, Derek N. Macklin, Nicolas T. Quach, Mialy M. DeFelice, Inbal Maayan, Yu Tanouchi, Euan A. Ashley, and Markus W. Covert. Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLoS computational biology*, 12(11):e1005177, 2016. .

Joaquin Vanschoren. Meta-Learning. In Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors, *Automated Machine Learning*, pages 35–61. Springer International

Publishing, Cham, 2019. ISBN 978-3-030-05317-8 978-3-030-05318-5. . URL http://link.springer.com/10.1007/978-3-030-05318-5_2.

Junde Xu, Donghao Zhou, Danruo Deng, Jingpeng Li, Cheng Chen, Xiangyun Liao, Guangyong Chen, and Pheng Ann Heng. Deep learning in cell image analysis. *Intelligent Computing*, 2022, 2022. . URL https://spj.science.org/doi/abs/10.34133/2022/9861263.

Dong Yang, Andriy Myronenko, Xiaosong Wang, Ziyue Xu, Holger R. Roth, and Daguang Xu. T-AutoML: Automated Machine Learning for Lesion Segmentation using Transformers in 3D Medical Imaging. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3942–3954, Montreal, QC, Canada, October 2021. IEEE. ISBN 978-1-66542-812-5. . URL https://ieeexplore.ieee.org/document/9710946/.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. MedMNIST v2 - A large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Scientific Data*, 10(1):41, January 2023. ISSN 2052-4463. . URL https://www.nature.com/articles/s41597-022-01721-8.

Bingtao Zhang and Peng Cao. Classification of high dimensional biomedical data based on feature selection using redundant removal. *PLOS ONE*, 14(4):e0214406, April 2019. ISSN 1932-6203. . URL https://dx.plos.org/10.1371/journal.pone.0214406.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid Scene Parsing Network. 2016. . URL https://arxiv.org/abs/1612.01105. Publisher: arXiv Version Number: 2.

Lucas Zimmer, Marius Lindauer, and Frank Hutter. Auto-PyTorch Tabular: Multi-Fidelity MetaLearning for Efficient and Robust AutoDL. 2020. . URL https://arxiv.org/abs/2006.13799.