UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

Ensembling and transfer learning for multi-domain microscopy image segmentation

Author:
Oleh MISKO

Supervisor:
Dmytro FISHMAN

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2020

Declaration of Authorship

I, Oleh MISKO, declare that this thesis titled, "Ensembling and transfer learning for multi-domain microscopy image segmentation" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

"This is the real secret of life — to be completely engaged with what you are doing in the here and now. And instead of calling it work, realize it is play."

Alan Watts

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Master of Science

Ensembling and transfer learning for multi-domain microscopy image segmentation

by Oleh MISKO

Abstract

A lot of imaging data is generated in medical, and particularly in the microscopy field. Researchers spend a lot of time analyzing this data due to slow algorithms and exhaustive manual work. Recent advancements in machine learning and especially deep learning areas resulted in methods that could be used to efficiently solve challenges in the microscopy imaging field. Image segmentation is one of the most common labor-intensive tasks that could be automated with deep learning approaches. One of the biggest challenges for computer algorithms in this domain is the problem of domain shift. The domain shift is the difference between the distribution of the data used for training and the distribution of the new upcoming data. In this work, we show that deep neural networks could efficiently segment microscopy images with the domain shift present. Moreover, we show that transfer learning from other medical tasks is an effective strategy to reduce the amount of required annotated data, whereas fine-tuning ImageNet models for microscopy segmentation gain little benefit.

Acknowledgements

First of all, I would like to thank my wonderful girlfriend Natalia for immense support during my study at Ukrainian Catholic University as well as my parents and grandparents for inexhaustible encouragement and faith in me.

I would like to acknowledge the University of Tartu, Institute of Computer Science and PerkinElmer company for supporting me and providing all the necessary resources for this work. I want to express my immeasurable gratitude to my supervisor and friend Dmytro Fishman who has found an unbelievable amount of energy and time to support, guide and review my work.

Furthermore, I would also thank the Ukrainian Catholic University team, especially Olexii Molchanovskyi, Rostyslav Hryniv, and Yaroslav Prytula for their tremendous effort to create, design and improve the best Computer Science program in Ukraine.

I am also very grateful to the team at Tartu University, namely Mohammed Ali, Tarun Khajuria, Sten-Oliver Salumaa, Kaupo Palo, Leopold Parts, Tõnis Laasfeld, and Mikhail Papkov for their support and valuable feedback.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

All essential biological processes that occur in living systems could be traced back to the microscopic scale. Microscopy imaging allows us to understand the life processes on this scale by analyzing the changes in cellular physical and biochemical properties and behavior under the influence of internal and external factors [1, 2].

A cell represents the smallest building block for any living organism. Normally, cells are grouped into cell lines (types of cells) based on their tissue of origin, physical shape, structure, density, and biochemical properties. These differences make the space of microscopy images vast and heterogeneous.

Microscopy imaging helps in drug discovery (reaction of the cells based on applied chemicals), allows a more reliable understanding of genetic perturbations (changes in cell properties over time) or a better interpretation of blood tests (live analysis of blood cells) [3].

Numerous image acquisition techniques (image modalities) introduce additional complexity into microscopy image processing. These modalities result in different types of images that require different approaches for the downstream analysis. In this thesis, we used two popular modalities - brightfield and fluorescence.

The brightfield modality technique is one of the most popular forms of microscopy imaging. This technique takes the dark specimen and contrasts it by the surrounding bright light field [4]. Brightfield images are cheap and relatively easy to acquire. However, these images often have low contrast values and thus are more challenging to analyze even for a human. An example of a brightfield image is presented in Figure 1.1.

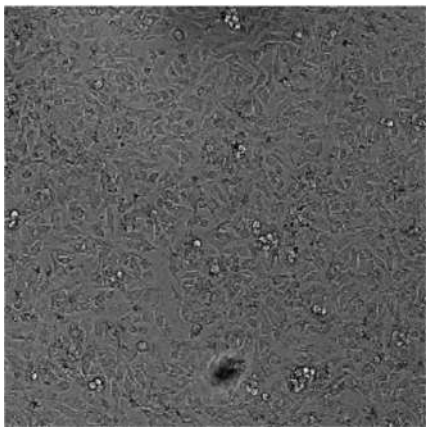

FIGURE 1.1: Brightfield microscopy image.

Fluorescent images, in contrast, are obtained using special fluorescent dyes that absorb incoming light and emit it back at a predefined wavelength. Using fluorescent signal biologists can experimentally highlight different parts of cells and produce noise-free images. However, obtaining such images is an expensive process, compared to the acquiring brightfield images, as one needs special chemical dyes. An example of a fluorescence image is presented in Figure 1.2.

Microscopy image analysis usually constitutes object segmentation, edge detection, object counting, and object area calculation [5]. At present, the majority of the aforementioned tasks are not fully automated, requiring practitioners to invest a lot of time into the manual work. In this work, we will address the problem of cell nuclei segmentation, which could be set as a semantic segmentation task. The main challenge is to accurately find distinct regions of the image that correspond to the nuclei and separate them from regions that correspond to the background [6].

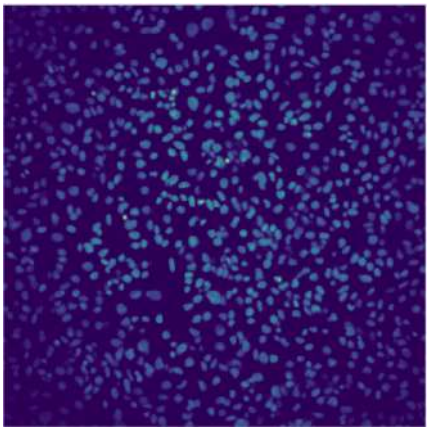

FIGURE 1.2: Cell nuclei highlighted by the fluorescent dye.

Artificial intelligence approaches and especially deep neural networks have become one of the most important technologies over the past decade. They demonstrate incredible results in different domains [1]. Deep neural networks have shown very prominent results in the field of microscopy image segmentation and could significantly decrease the amount of time spent by a human on these problems [7].

Neural networks produce remarkable results but require large amounts of annotated data. In the microscopy field, it is normally hard to obtain sufficient amounts of data labeled by a professional biologist. Therefore, the performance of the deep neural networks is highly dependent on a sufficient amount and good quality of the data. Additionally, after being trained on one dataset (source domain), neural networks could produce a significantly worse performance on the new data (target domain). This is caused by differences in the distributions of the target and source domain datasets which is called the domain shift. In the microscopy field this difference is caused by different intensity levels, the variability of patterns across different types of cells and noise [8].

Transfer learning is a popular strategy to reduce the amount of annotated data required for model training. It is normally done by fine-tuning the pre-trained network. The data, that the network was previously trained could be from a general domain [7] or other medical domains [9].

Transfer learning has shown to yield better results when the tasks of the source and target network are similar [10]. Recent studies show that transferring weights

[1] 2010 – 2019: The rise of Deep Learning

from a distant domain, like ImageNet, into medical domains would not always give better results and may harm the performance of the model [11].

In this work, we will examine what training strategy is the most robust to the variations in the dataset for two modalities - brightfield and fluorescence. We will perform experiments for cases when the training and testing data was sampled from the same or different distributions. Moreover, since obtaining annotated data for microscopy segmentation is normally hard we will explore the feasibility of using transfer learning from two different medical domains and domain of natural objects (ImageNet).

1.2 Research questions and goals

- What training strategy achieves the best generalization performance on hold-out data in case hold-out data was sampled from:
  - The same dataset (no domain shift)
  - Different dataset (domain shift is present)
- How transferring model weights, obtained from
  - Similar medical domain (small domain shift is present)
  - Different medical domain (significant domain shift is present)
  - Distant natural objects domain

  affects training and therefore, generalization performance of the model on hold-out data?

Chapter 2

Related work

2.1 Rule-based approaches

Some of the most popular approaches for automated nuclei segmentation belong to the family of rule-based methods. These methods use the experts' knowledge to create a set of tunable rules that would help an algorithm to identify pixels that belong to nuclei.

One of the most well-established methods used for cell nuclei segmentation is Otsu's thresholding [12]. This algorithm iterates over threshold values from 0 to 255 on a grayscale image. At each step, the algorithm divides image pixels into two distinct classes - foreground and background. Then, the variance of pixel intensities for both classes is calculated. The algorithm searches through threshold values until it finds the one that minimizes the within-class variance for both foreground and background pixels

Typically, Otsu's thresholding is used in combination with the watershed transformation algorithm [13]. This algorithm treats the image as a "topographic map" where the intensity of each pixel is representing the "height of the point". For example, dark areas can be interpreted as valleys and lighter parts as hills or mountains [14].

Otsu's thresholding is easy to implement but it is computationally inefficient due to exhaustive search through all possible threshold values. Moreover, it has shown to be sensitive to noise and variations in pixel intensity levels [15]. Finally, Otsu's thresholding commonly requires manual parameter tuning from dataset to dataset or even from image to image.

Further work aimed at improving the execution time of Otsu's method [16]. The authors proposed a sequence of steps that replaced the exhaustive threshold value search. First, the color histogram values of the input image are obtained. A color histogram is a representation of the distribution of colors in an image. This histogram is then used to calculate the number of gray shades (unique pixel intensities between white and black) in the image. Initial global threshold value then is calculated as

$$T = \frac{\sum h \times shades}{\sum h} \tag{2.1}$$

, where h is the color histogram values and *shades* is the number of gray shades in the image

At each iteration, the image is segmented using a threshold value of T. New threshold values T_{fg} and T_{bg} for two classes are calculated using (2.1). Finally, the global threshold value is updated as

$$T = \frac{T_{fg} + T_{bg}}{2} \tag{2.2}$$

These steps are repeated until the difference in T in the successive iterations is not zero.

Authors state that their modification works a hundred times faster than the original algorithm. They use accuracy as the ratio between the number of correct predictions to the total number of predictions. Improved version achieves 0.84 pixel accuracy compared to classical Otsu's thresholding with a value of 0.79

To summarise, rule-based approaches are easy to implement, but they require a lot of human interaction and fine-tuning. In addition, they yield poor results with images with significant amounts of noise or with high variance in pixel intensity values.

2.2 Machine learning approaches

Machine learning approaches were the next step in the attempt to automate nuclei segmentation [15]. Usually, these methods are used in combination with thresholding or image gradient extraction to produce more accurate results. Machine learning algorithms could be used for segmentation in either supervised or unsupervised mode.

Classification models belong to the family of supervised learning algorithms. They predict the class for each pixel in the image using the knowledge about neighboring pixels and ground truth generated by experts. Frequently, classification models are combined with thresholding algorithms. Using thresholding as the first step to classification algorithms, allows the model to concentrate on important features like object contours. Therefore, this combination should increase the effectiveness and performance of the whole pipeline.

Popular classification models for biological images segmentation are: k-nearest neighbors (KNN), decision trees, support vector machines, and random forests [17]. These models were shown to work well in segmentation [15]. Nonetheless, they require a reasonable amount of labeled data (normally produced manually) for training which is not always available. Moreover, classification models similar to thresholding methods that were described in the previous section may be sensitive to pixel intensity variations and noise [18].

Clustering models belong to the family of unsupervised learning algorithms. These models search for hidden patterns in the input data. Such patterns allow the algorithm to categorize the data into several distinct regions. Normally, these methods are combined with thresholding or color histogram extraction methods to improve the resulting performance [17].

Most of the reviewed works seem to consider K-Means clustering, DBSCAN and Expectation-Maximization (EM) algorithm with Gaussian Mixture Models (GMM) [15]. Clustering models do not require labeled data and could produce fair results. However, they are sensitive to initialization and the amount of noise in the data. [19, 20].

Another strategy is called Markov Random Fields (MRF) modeling. It is a statistical model that predicts the relationship between the neighboring pixels. MRFs use a Bayesian hypothesis that neighboring pixels should fall into the same class. Usually, this approach is combined with clustering algorithms. This combination gives better results than standalone clustering methods [15]. Despite obvious advantages, these approaches tend to depend a lot on initial parameter selection and are computationally inefficient [21].

Machine learning models have been shown to achieve very good results in medical image segmentation. They usually outperform rule-based methods and are more practical as they do not require a lot of manual tuning [15]. However, they are sensitive to noise and pixel intensity variations which may involve complex preprocessing of the images to generate good results.

2.3 Deep learning approaches

Deep learning approaches, such as deep artificial neural networks, have been successfully applied to solve numerous problems in the area of medical image processing [22]. Deep learning models require neither manual parameter tuning as rule-based methods nor complicated pre- and post-processing mechanisms as machine learning approaches. Artificial neural networks may be robust to noise, pixel intensity variations and distribution differences between the images [23, 24]. Moreover, these approaches normally yield better results than classical approaches [25].

In this work, we will consider deep learning approaches to solve the microscopy image processing task. Therefore, now we will review several noteworthy applications of deep neural networks in medical image processing. More specifically, we will talk about applications in segmentation problems.

One of the noticeable applications of deep learning in microscopy imaging is nerve fiber segmentation [26]. Authors use a U-Net [27] like architecture to segment huge biopsy images [26]. U-Net is a fully convolutional neural network (FCN) that was developed for biomedical image segmentation. This architecture consists of two parts: encoder and decoder. The encoder part performs image downsampling and learns feature representations. On the other side, the decoder part performs image upsampling and locates the features on the image. The network uses skip connections between the encoder and decoder sides to locate the features more accurately. In this study, smaller images (patches) were extracted from the original images and used as input to the network. For each input patch, the network generates a binary mask, where pixels that belong to the fiber tissue are separated from the background. Obtained binary masks are stitched together to create the full-size mask of the original image.

The authors use a pixel-wise F1 score as an evaluation metric. The F1 score metric is a harmonic mean between precision and recall, which is computed based on true positives (TP), false negatives (FN) and false positives (FP). In this work, true positives are the pixels that are correctly classified as tissues. False positives and false negatives denote pixels that are incorrectly classified as the opposite class. The results show that this approach outperforms classical machine learning models. Moreover, their strategy yields better results than a manual segmentation by a novice biologist and almost as good as segmentation by an expert level professional.

Another noticeable work [2] compares deep learning approaches like U-Net [27] and DeepCell [28] with Random Forest algorithm and CellProfiler rule-based approach [29] in a cell perturbation segmentation task. The authors use pixel-wise F1 score and Jaccard index to evaluate the models. Such a combination of metrics addresses both pixel and object level performance of the model. Jaccard index quantifies the percentage of overlap between the ground truth mask and model prediction output. The results show that deep learning produces better results than machine learning and rule-based approaches. Moreover, U-Net was shown to perform faster than DeepCell and make fewer errors.

Another work compares several deep learning approaches for natural light (bright-field) and fluorescent microscopy segmentation [30]. The authors compare the performance of DeepCell, U-Net, and Mask R-CNN [31]. Mask R-CNN is a deep neural network originally aimed to solve instance segmentation problems. This network operates in two stages. First, it generates predictions about the regions where the object might be on the image. Second, it predicts the class of the object, draws the bounding box and generates a pixel-level mask of the object based on the predictions from the first stage.

Pixel-wise F1 score and AUROC metrics were used to evaluate the strategies. Authors show that DeepCell segments one image more than 500 times slower than other approaches rendering it impractical. Consistently with the previous study [28], U-Net outperformed Mask R-CNN in terms of pixel-wise performance, which in its turn was still superior to DeepCell.

Deep learning approaches are widely adopted in different medical image processing tasks. They normally produce better results than machine learning approaches and require less manual interaction. Nevertheless, training neural networks requires a lot more labeled data comparing to classical machine learning approaches [22]. One of the possible solutions is to reuse knowledge from other domains - transfer learning. Deep neural networks may be sensitive to hyperparameter selection and require considerable computational resources for training.

2.4 Transfer learning

Training a deep neural network from scratch may be a hard and computationally expensive task. In this chapter, we will review some of the noticeable works that use a strategy called transfer learning, that may potentially make model training faster and decrease the required amount of labeled data.

A remarkable work [32] aims to understand the superior performance of deep learning approaches. The authors explored features that deep neural networks consider important by visualizing activations of hidden layers. These visualizations were compared between different architectures trained on the same set of images (ImageNet dataset). The results showed that all models learned strikingly similar features despite drastically different architectures. The authors suggest that the weights that the network learns may be reused in different tasks. Therefore, the authors try to adapt weights from the ImageNet pre-trained model to the Caltech Pedestrian Detection task [33]. The results show that pre-trained model weights could be efficiently transferred in the new domain with some adaptation. The process of adaptation is usually called fine-tuning. In essence, fine-tuning is a process to take some model that was trained for a given task and make it perform a second similar task. Assuming the original task is similar to the new task, using a network that has already been designed & trained allows us to take advantage of the feature extraction that happens in the front layers of the network without developing that feature extraction network from scratch [34].

This study shows us that transferring weights from ImageNet domain to the other might introduce several benefits. First of all, this approach requires less resources for training the model, as the fundamental parts of the network (first layers) are already trained. One only needs to fine-tune the weights on a new domain, by training the network for several epochs, which in practice takes much less time than training from scratch. This approach helps if you don't have a lot of labeled training data since you don't need to train the network from scratch. Other works show that

transfer learning might produce even better results than training the model from scratch [35].

One of the works examined the effect of the transfer learning on chest disease classification problem [36]. The authors compared pre-trained ResNet-50 [37] with the same model trained from scratch. The authors evaluated models separately using pixel-wise F1 score. The results show that the pre-trained model produces better results in the classification of seven out of eight different diseases. Moreover, it requires less data to perform on par with the model trained from scratch.

Further work aimed to improve the accuracy of chest disease classification with transfer learning approaches [38]. The authors use a DenseNet-121 model [39] pre-trained on the ImageNet dataset. They re-train the whole model on 98,000 frontal chest x-ray images. The results show that their approach produces from 1% to 9% better results than previous works with respect to pixel-wise F1 score [36, 40]. Moreover, the authors show that their approach yields results as good or even better than expert level practicing radiologists. The main feature of this work from previously mentioned work [36] is that authors are using different network architecture and more data for fine-tuning.

Other works mentioned the benefits of transfer learning for diabetic retinopathy (DR) detection [41]. The authors replaced random forest classifiers, used in the medical device for diagnosing DR, with a convolutional neural network. They used Inception-v3 [42] pre-trained on ImageNet dataset and fine-tuned it on 25,000 DR images. They show that this modification resulted in a 6% increase in pixel-wise F1 score, compared to the randomly initialized network.

One more interesting work examines the feasibility of transfer learning from ImageNet domain to medical domain tasks [11]. The authors compare the performance of pre-trained and randomly initialized models on two medical tasks - chest x-ray segmentation and DR detection. They select ResNet-50 and Inception-v3 as prominent ImageNet competitors. The results show that much smaller architectures achieve almost the same pixel-wise accuracy as larger ImageNet models. Authors explore the layer activations and model outputs and conclude that large models adapt to new domains much slower than the small ones. Therefore, they state that transfer learning from a natural domain like ImageNet doesn't always give an improvement. The results show that randomly initialized models achieve 96.4%, whereas pre-trained fine-tuned models achieve 96.7% of pixel-wise accuracy in both tasks. Moreover, the authors state if the number of weights is large, then transfer learning may even harm the performance, compared to training the model from scratch or using a smaller model.

Summarizing, transfer learning is easy to implement a technique that may improve the performance of the deep neural network. It may drastically reduce the amount of labeled data needed, which was presented in several works [36, 41]. We decide to use transfer learning in our work and compare the performance of a fine-tuned model with a randomly initialized model. If the fine-tuned model will produce better results this may potentially have a huge impact on the biology field. This will mean that one doesn't need a lot of labeled microscopy data (which is usually hard to obtain) and expensive computational resources to perform automated microscopy data segmentation.

Chapter 3

Proposed approach and implementation

3.1 Dataset

For model training and evaluation two datasets of fluorescence and brightfield modalities provided by PerkinElmer company (measured by Opera Phenix microscope) [43] were used. These datasets represent the results of corresponding cell measurements of seven cell types. An example of the images is presented in Figure 3.1. The ground truth masks are generated from the fluorescence modality using PerkinElmer Harmony software with manual parameter tuning. The output of the software was manually evaluated by the human expert [30].

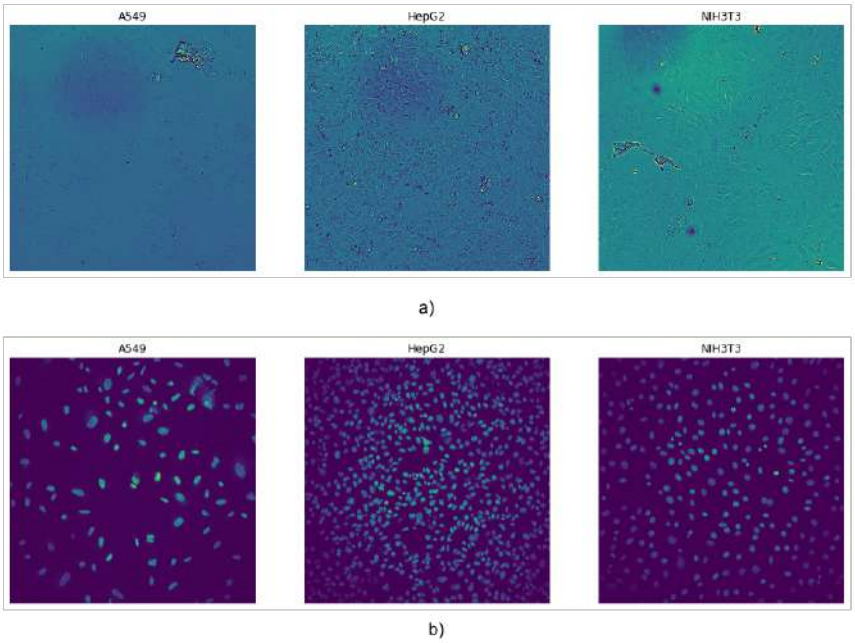

FIGURE 3.1: Differences between cell lines in a) brightfield and b) fluorescent images.

These datasets provide high-resolution images and quality-labeled ground truth masks of size 1080 × 1080 px. The distribution of the data across the cell lines is presented in Table A.1.

Eight random overlapping patches of size 288 × 288 px. are extracted from each image in the training set. This approach allows the network to efficiently process the data and also serves as an augmentation strategy. In the inference phase, each

test image is divided into several sequential patches of the same size. The trained model then predicts each patch separately. Next, the predicted patches are stitched back to form the input image probability map, where each pixel is assigned a probability score. Scores may be interpreted as the likelihood of the pixel to belong to foreground (nuclei).

Additionally, for transfer learning experiments we used the dataset from AstraZeneca [44] company, measured by CellVoyager microscope. This dataset provides high-resolution 2556 $\times$ 2156 px brightfield modality images of one additional cell line. The distribution of the data in this dataset is presented in Table A.2.

3.2 Learning strategies

3.2.1 Individual models

This strategy proposes to train a separate model for each type of cells as presented in Figure 3.2. The main benefit of this strategy is the high performance of the models on the respective cell lines. Each individual model is tested on the same type of cells, that is used for training. The major drawback is that for a new cell line one would need a reasonable amount of manually labeled data (which is usually hard to obtain). Moreover, if one doesn't have labeled data for new cell type it is unclear which model to select for segmentation. The problem arises from the fact that models are biased towards their respective cell lines and thus may produce poor results on different types of cells. Finally, it may be infeasible to train and maintain a vast amount of models in practice.

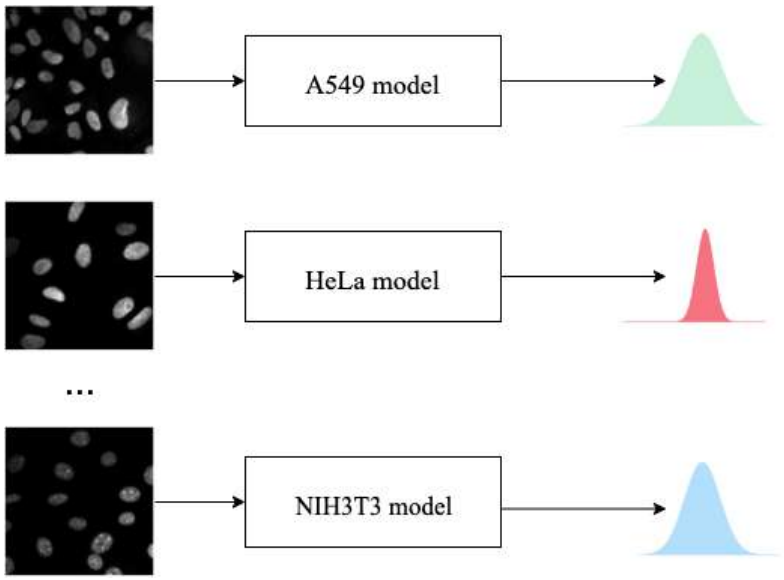

FIGURE 3.2: Individual models learn the distribution of a single cell line.

3.2.2 Master model

This strategy suggests training one model jointly on all types of cells as presented in Figure 3.3. During the training, the model combines the knowledge about different cells to learn a generalized representation. In practice, it's much easier to train and maintain only one model instead of separate model for each cell type. Furthermore,

new data could be used to fine-tune the model. Normally, you need less labeled data if you fine-tune the model instead of training a model from scratch [11, 45]. However, the standalone model may be sensitive to differences between cell representations on images and data imbalance [46].

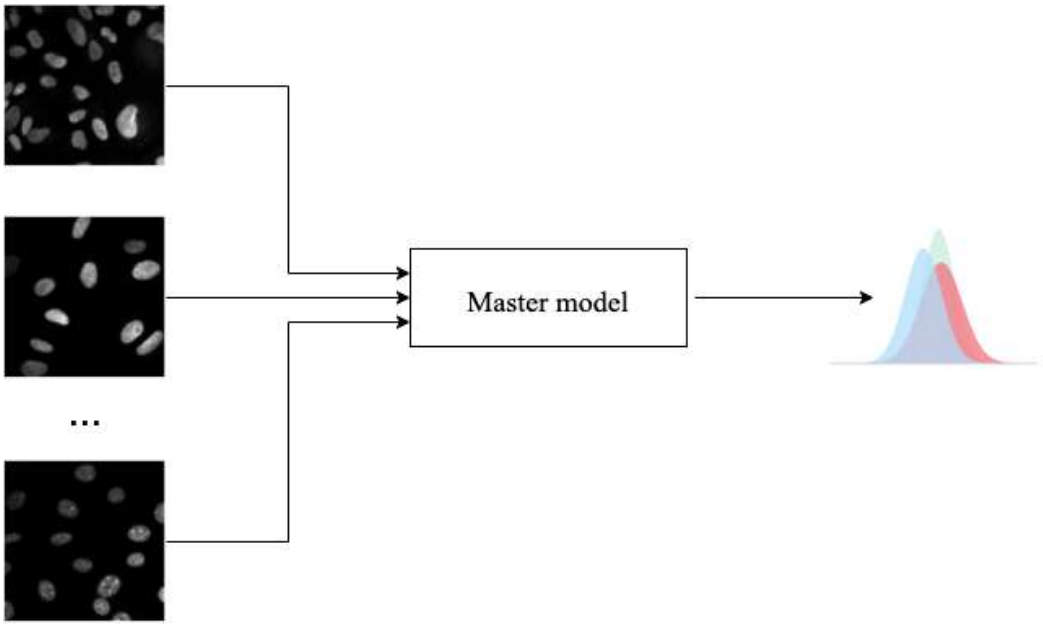

FIGURE 3.3: Master model learns a joint distribution from several cell lines.

3.2.3 Naive ensemble

Ensemble methods normally produce noticeable results in different tasks [47, 48]. These methods combine a set of models called "weak learners" to create a "strong learner". The naive ensemble approach suggests using individual models as ensemble members. The pipeline of this approach is presented in Figure 3.4. The input image is provided to each ensemble member to produce segmentation masks. Then these outputs are averaged to produce one final mask.

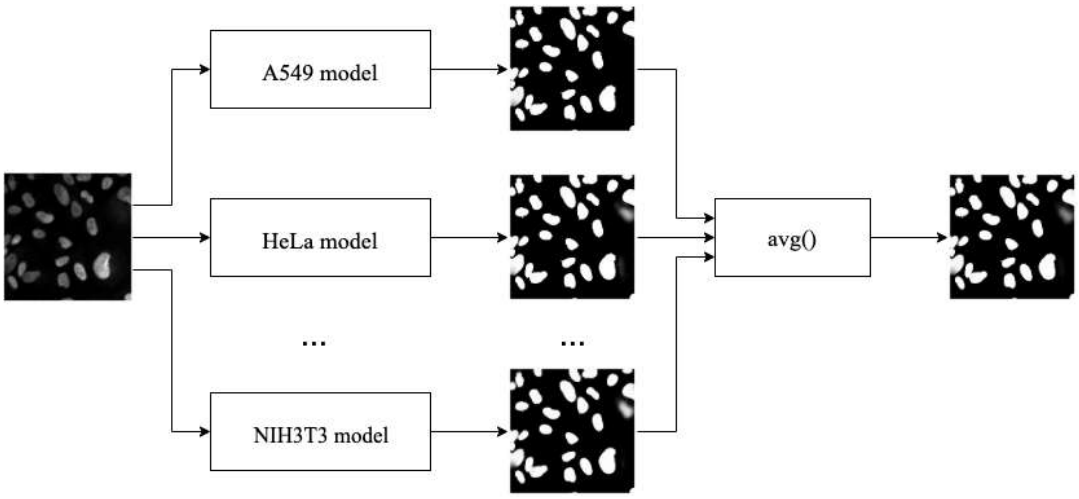

FIGURE 3.4: Naive ensemble averages individual model predictions.

This approach is more flexible than using standalone individual models and may yield prominent results. However, you still need trained individual models for this strategy. Moreover, the output of each model in the ensemble is given the same weight which may harm the performance. Some cells are more or less similar to others in terms of shape, size, and density. Therefore, some individual models may perform better on similar cell types, than others. The modification of the naive ensemble approach will be described in the next chapter.

3.2.4 Weighted ensemble

As stated previously some models may perform better on cell types, that are similar to their original training data in terms of shape, size or density of the nuclei. This strategy incorporates knowledge about the performance of individual models on different types of cells as presented in Figure 3.5. First of all, half of the validation set is hold-out from each cell line. The performance of each individual model is then measured on these hold-out sets separately by pixel-wise F1 score. Next, we square these scores to emphasize the differences between the performance levels and normalize them to sum to one. In the end, we obtain a weight vector of size $N \times 1$ for each cell line, where N is the number of individual models in the ensemble.

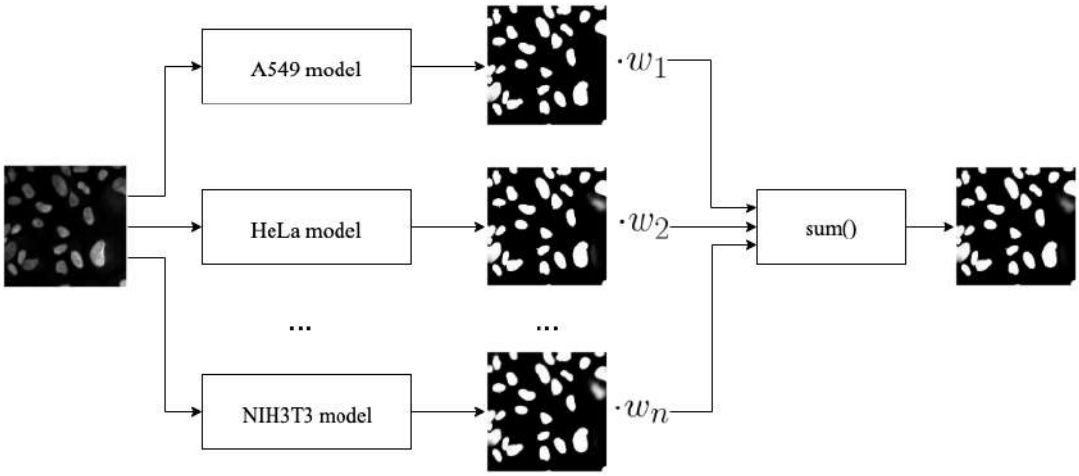

FIGURE 3.5: Weighted ensemble uses prior knowledge about cells similarity.

We load the corresponding ensemble weights with knowledge about cell type of the input image. Then we multiply the predictions of each ensemble member by the respective weight and summarize those predictions.

3.2.5 Stacking ensemble

A stacking ensemble is an approach where a new model, called meta-model, is trained on the ensemble predictions. The meta-model should combine the predictions from weak learners to produce a best-combined result. The predictions from the weak learners are packed into a tensor of shape $(K, 1080, 1080, 1)$, where K is the number of individual models in the ensemble. These predictions are used as inputs to the meta-model [49]. This strategy has shown to produce very good results in medical image processing tasks [50]. The pipeline of this approach is presented in Figure 3.6.

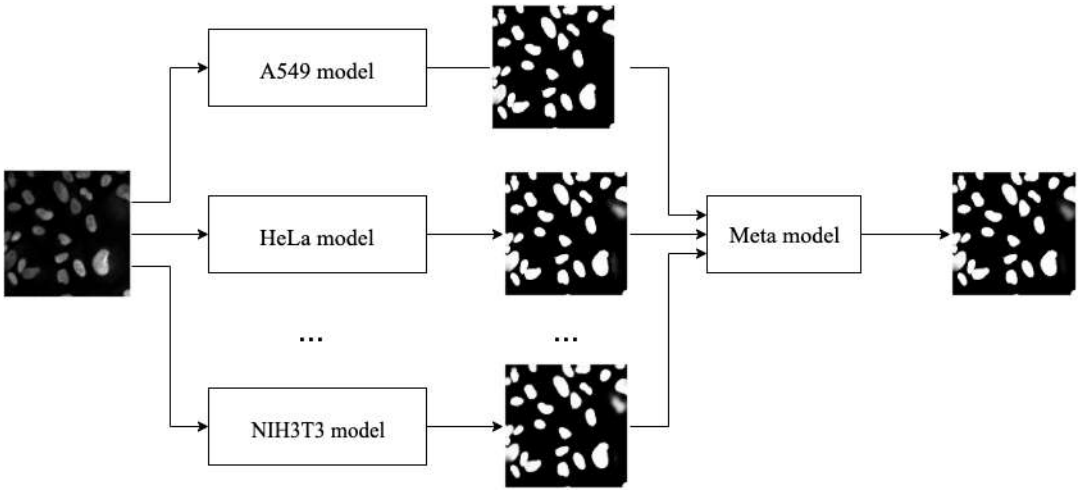

FIGURE 3.6: Stacking ensemble uses meta-model to combine ensemble predictions.

This approach requires nor the information about the cell type of the input image neither the pre-computed weights. However, it may be hard to deploy and maintain this strategy in practice [49].

3.3 Model training configuration

Based on the related work the U-Net architecture [27] is considered. This network produces good results across different medical segmentation tasks [2, 26, 30]. The weights of the network are initialized using Xavier initialization [51] which is a popular strategy that helps the network to converge faster [52]. ReLu is used as an activation function [53].

To tune the weights in the network the Adam optimizer [54] is used with a learning rate $\alpha = 1e-5$, the exponential decay rate for the first moment estimates $\beta_1 = 0.9$ and exponential decay rate for the second-moment estimates $\beta_2 = 0.99$. A learning rate scheduler is used to decrease the learning rate by a factor of 0.1 when the loss function reaches the plateau. Additionally, early stopping is used to stop model training when the difference in successive loss function values is less or equal to 0.01.

Binary cross-entropy loss function is used since the problem is stated as binary segmentation.

3.4 Experiments setting

First, we perform two experiments that assess the overall performance of the training strategies presented above. These experiments measure the impact of the domain shift on the performance of these strategies. In essence, domain shift represents the difference between the distribution of the data used for training (target domain) and the new data (source domain).

The first experiment evaluates the training strategies when the target domain and source domain data come from the same distribution. We use seven cell lines for both training and testing. This allows us to minimize the discrepancy between the domains. The master model was trained on 256 images from each cell line, resulting

in 1792 images in the training dataset. Individual models were trained on 256 images from the corresponding cell line. As stated in the previous chapter ensembles are constructed from seven trained individual models.

Additionally, we performed experiments to show how much data is necessary to train a model with reasonable performance. We train each strategy on a different number of images for both modalities separately. The master models are trained on $N \times M$ images from each respecting cell line. N here refers to the powers of two in the range from 0 to 8. M represents the number of cell lines which is equal to 7. The individual models are trained on N images from the corresponding cell line. In the case of the stacked ensemble, the meta-model was trained on $N \times M$ images/masks, predicted from the hold-out validation set by each ensemble member. In each experiment, we have trained new randomly initialized models. The results of these experiments could be found in Appendix C.

The second experiment evaluates training strategies when the target and source data come from different distributions. We conduct a series of experiments where we train the models using six cell lines (source domain) and one cell line is hold-out (target domain). This hold-out cell line testing set is used for evaluation. This experiment is performed for each of the seven cell lines in sequence. The master model here is trained on 256 images from six cell lines which yield 1536 images in the training set. The ensembles are constructed from the six individual models trained in the previous experiment and evaluated on images from the seventh, left out cell line.

Finally, we examine the impact of transfer learning using the master model as the main strategy. We use three different strategies in this experiment: no fine-tuning, fine-tune only on the target domain and fine-tune on both target and source domains.

In the first strategy, we use a model that was trained only on source domain data. This strategy should indicate how distant are the target and source domains and will produce good results if these domains are very similar (which is rarely the case).

The second strategy proposes to fine-tune a model only on the target domain data. Such situation may happen when one has a pre-trained model but doesn't have access to the data, that the model was originally trained on (from source domain). This strategy inevitably leads to the degradation of performance on the source but allows us to gain performance on the target domain. The results will show whether it's better to train a new model from scratch (for target domain) or fine-tune an already trained model.

The last strategy proposes to use the same amount of the data from both target and source domains for fine-tuning the model. This approach is applicable when one has access to the original model training data and labels. The motivation is to preserve the performance on the source domain while gaining on the target.

We compare the results from these strategies with a randomly initialized master model trained solely on the target domain to evaluate the applicability of the transfer learning approach.

For the experiments from a distant domain (ImageNet), we used U-Net like architectures with ResNet101 [37] and VGG-16 [55] backbones and compared these fine-tuned models with U-Net master model trained from scratch. In essence, a backbone represents what architecture is used for encoder and decoder parts in U-Net like systems.

We performed additional experiments to evaluate how the size of the dataset for fine-tuning impacts the performance of the strategies presented above. The results of these experiments are available in Appendix D.

3.5 Model evaluation

3.5.1 Pixel-wise metrics

To address the pixel-level performance of the models the pixel-wise F1 score is considered. This metric is used in various works dedicated to medical image segmentation [2, 26, 30] and makes the results comparable. The F1 score which is a harmonic mean between precision and recall is calculated using equation 3.1

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3.1}$$

where

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} \cdot \text{false positives}} \tag{3.2}$$

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} \cdot \text{false negatives}} \tag{3.3}$$

3.5.2 Object-wise metrics

The object-level performance of the model is a crucial part of microscopy image segmentation since we are mostly interested in the correct segmentation of distinct nuclei. To address the object-level performance of the models an object-wise F1 score based on the Intersection-over-Union (IoU) metric is considered. The IoU metric, also referred to as the Jaccard index quantifies the overlap between the ground truth mask and model prediction. First, the predictions of the model are thresholded by different values in range $[0.5, 0.95$ with step 0.05. At each step, the IoU metric is calculated using equation 3.4.

$$\text{IoU} = \frac{\text{ground truth} \cap \text{prediction}}{\text{ground truth} \cup \text{prediction}} \tag{3.4}$$

These values are then averaged. Next, we compute true positives, false positives and false negatives based on the IoU metric. Lastly, using equation 3.1 we compute object-wise F1 score.

3.5.3 Error metric scores

The results for both modalities would be presented in terms of pixel- and object-wise **F1 error**. Models trained on the fluorescence data usually achieve relatively big values of the F1 score close to 1. Therefore, it's hard to infer from the plots how much one training strategy is better than others in terms of the F1 score. To make the results for both modalities comparable we will use error scores. The F1 error is calculated using the equation 3.5.

$$F_{1\,\text{err}} = 1 - F_1 \tag{3.5}$$

Chapter 4

Experiments results

4.1 Performance of training strategies when source and target distributions are the same

To show which type of training is preferred when training and test data are sampled from the same distribution, we compared all five aforementioned strategies.

4.1.1 Fluorescence modality results

Fluorescent data is normally easy to segment for both biologists and computer systems. The cell nuclei on this type of data are well distinguishable as presented in Figure 3.1. Therefore all strategies achieve very small values of both pixel- (Figure 4.1) and object-wise (Figure 4.2) F1 error scores.

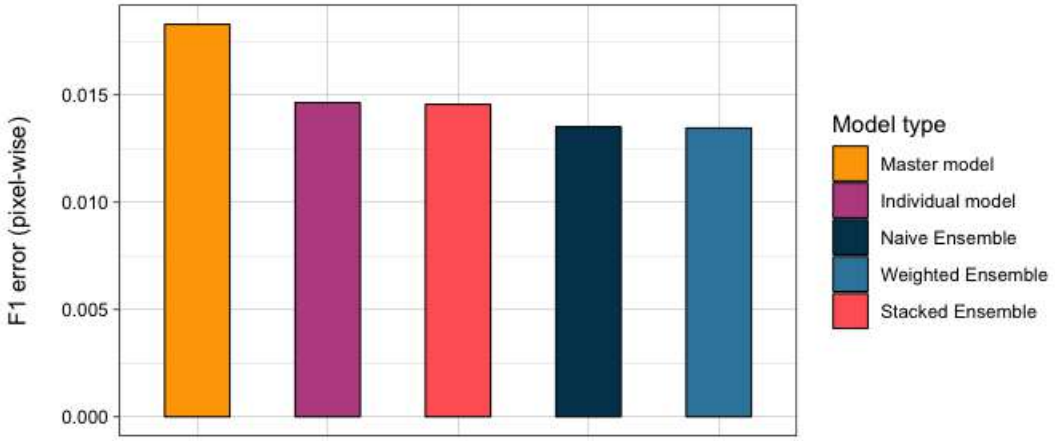

FIGURE 4.1: Ensemble strategies achieve the smallest pixel-wise F1 error.

The weighted and naive ensemble strategies show slightly better results due to the combination of individual models (which solely demonstrate excellent results). The stacked ensemble produces relatively moderate performance and the meta-model doesn't give an improvement in fluorescence case (compared to the naive ensemble which consists only of individual models). The master model performs the worst (in comparison), but the difference between the master model and weighted ensemble (which achieves the best performance) error scores is less than 0.01.

The results for each cell line separately could be found in Figure B.1 and Figure B.2 respectively. All strategies show relatively the same performance across cell lines. The models make most errors on HepG2 and MCF7 cell lines due to the

high density (and therefore heterogeneity) of the cells on these images. In contrast, HT1080, MDCK and NIH3T3 are less complex and thus models perform better on these cell lines in fluorescence modality.

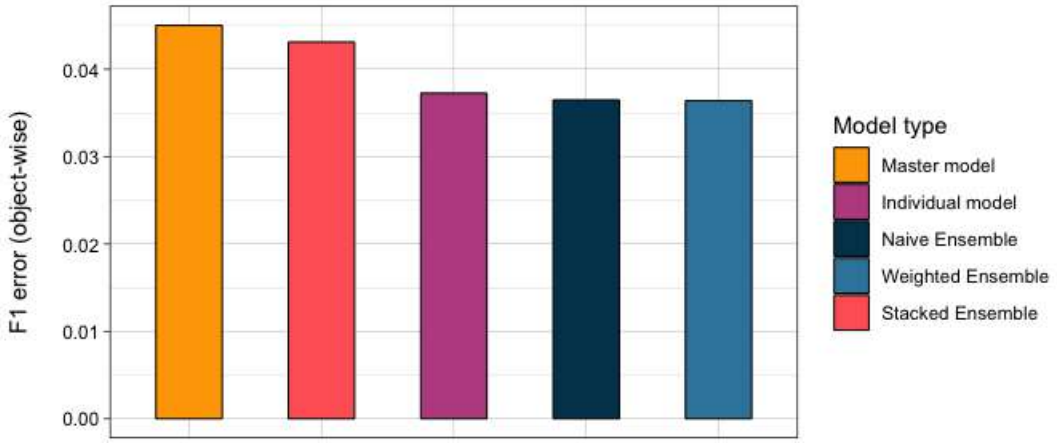

FIGURE 4.2: Individual models slightly outperform stacked ensemble approach on object-level.

Despite having a slightly higher error, the master model doesn't require any additional information about the underlying dataset (cell lines info) and needs no intricate training pipeline. Therefore, it seems to be the most reasonable strategy for fluorescence data segmentation. In other words, if one expects to segment the same or similar type of fluorescent data in the future it might make sense to train one big network (master model) what will perform this task, rather than building a complex system of individual models (which may not yield a significant performance gain).

Additionally, we evaluated how the size of the training dataset affects the performance of the strategies. The results of this experiment are presented in Figure C.1 and Figure C.2. In essence, one training image is enough for all strategies to produce satisfactory results with average error scores of 0.025 for pixel- and 0.035 for object-wise metrics. The naive, weighted and stacked ensemble achieve steady performance increase (as the number of training images increases), whereas individual and master model oscillates starting from 8 training images from each cell line due to limited standalone model capacity.

4.1.2 Brightfield modality results

In contrast to fluorescence modality, brightfield data is much harder to segment for both biologists and computer algorithms. The cell nuclei are much less distinguishable compared to fluorescence, which may be noticed in Figure 3.1. Therefore, both pixel- (Figure 4.3) and object-wise (Figure 4.4) F1 errors are significantly higher than for the fluorescence.

The individual model strategy achieves the best performance in both metrics. From this point onwards this strategy will be considered as the upper bound of achievable performance on brightfield segmentation task for a separate cell line. However, as stated before this strategy requires a reasonable amount of annotated data (which is normally hard to get for brightfield modality). The stacked ensemble demonstrates prominent results and the positive effect of the meta-model here

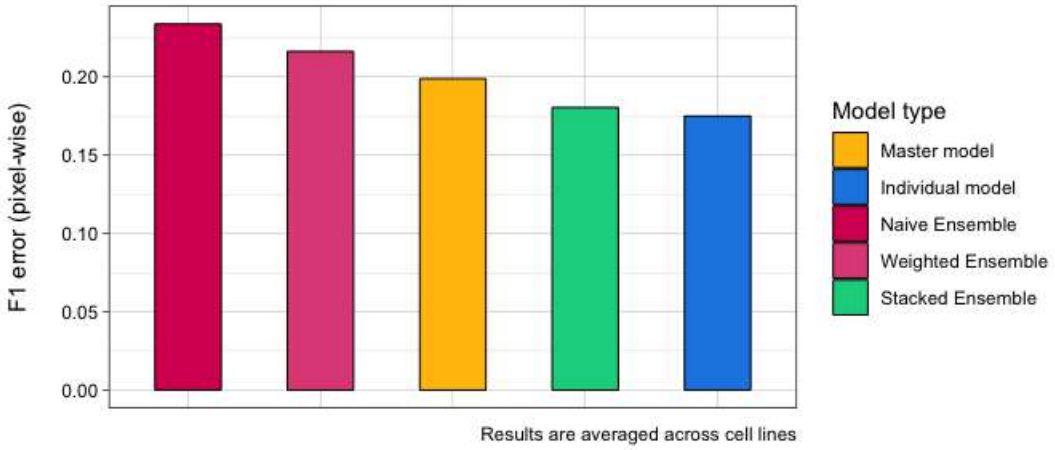

FIGURE 4.3: Individual models show superior performance in brightfield modality.

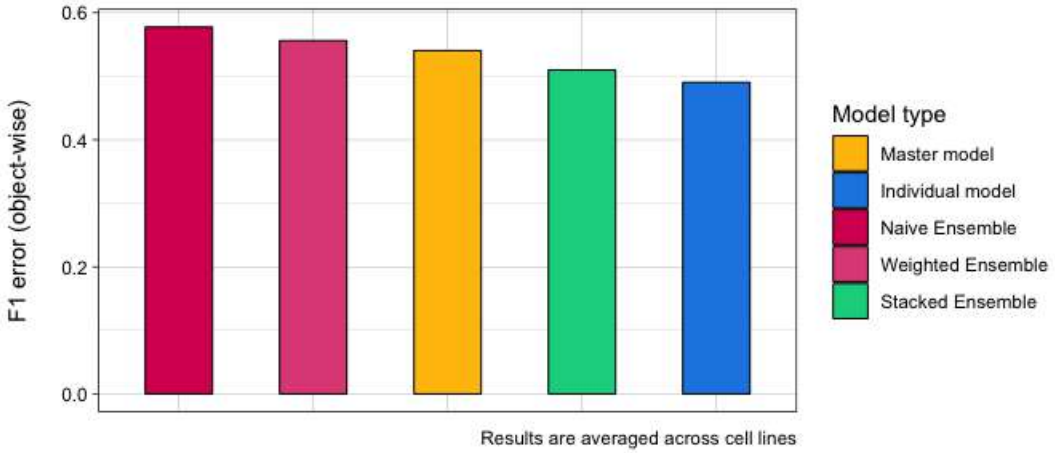

FIGURE 4.4: Stacked ensemble produces satisfactory results for brightfield modality in both pixel and object-wise levels.

is clearly visible (in contrast to fluorescent modality) as the naive ensemble demonstrates relatively unsatisfactory results. The weighted ensemble produces worse results than the master model and stacked ensemble and is impractical for complex brightfield modality. The master model achieves moderate error scores by using joint knowledge about different cell lines. It is easier to implement than other strategies and should be considered for brightfield data segmentation as well as for fluorescent.

The results for each cell line separately could be found in Figure B.3 and Figure B.4 respectively. The HepG2 and MCF7 seem to be the most complex cell lines (which was already seen in fluorescence experiments) with the addition of A549 for brightfield modality. The models produce reasonable results on HT1080, MDCK, NIH3T3 and HeLa cell lines. In general, the pattern of performance is relatively the same for both brightfield and fluorescence.

The effect of the training set size for brightfield is presented in Figure C.3 and Figure C.4. In contrast to fluorescence, brightfield models require much more annotated data to produce satisfactory results. All strategies show a continuous decrease in error metrics with the increase of training set size. Therefore, for brightfield modality, it's good to obtain as much good-quality labeled data as possible.

4.2 Performance of training strategies when source and target distributions are different

In this experiment, we compare the aforementioned training strategies when training and testing data are sampled from different distributions. The individual model strategy is not added as it could not represent target and source domains at the same time.

4.2.1 Fluorescence modality results

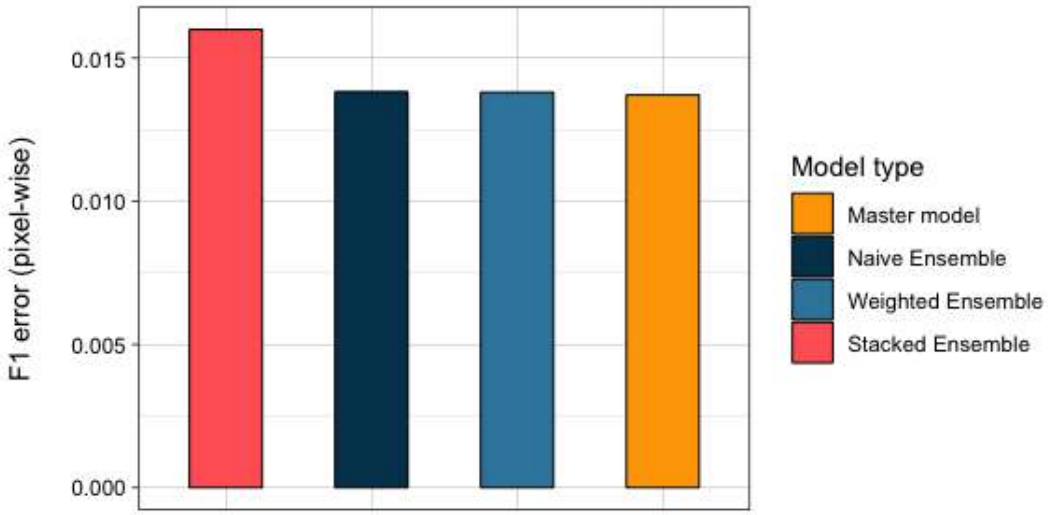

FIGURE 4.5: The stacked ensemble is the least domain shift robust strategy.

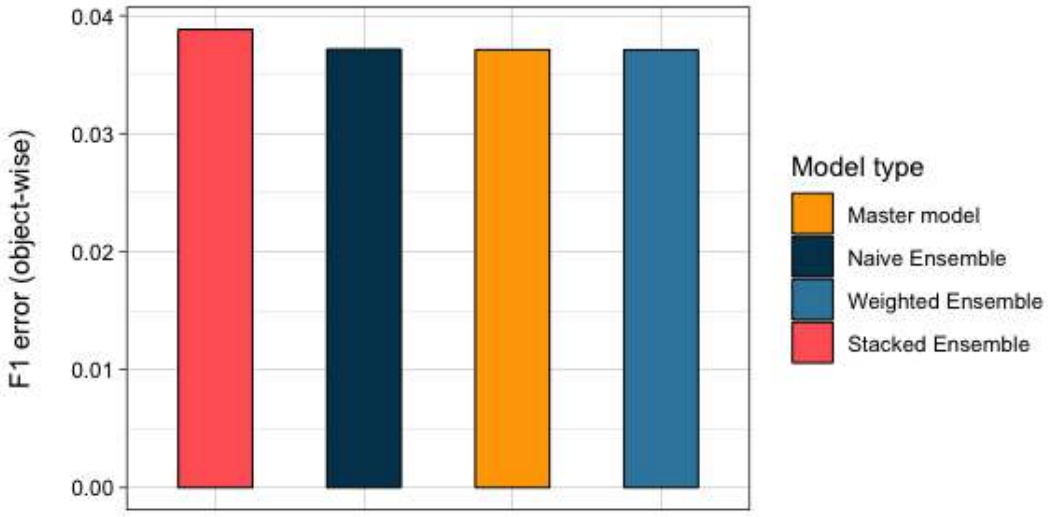

FIGURE 4.6: Master model and weighted ensemble show best results in both pixel- and object-wise scores.

As well as for experiments with no domain shift, all training strategies achieve low errors and the particular difference is minor. The weighted ensemble and master model show the best pixel- (Figure 4.5) and object-wise (Figure 4.6) performance.

However, the latter doesn't need the information about the input cell line and thus is simpler to use in practice. Moreover, the master model strategy is easier to adapt in case of new upcoming data (from different distributions). The stacked ensemble strategy renders impractical for fluorescence segmentation with domain shift, as the naive ensemble (which doesn't have meta-model) produces slightly better results.

The results for each cell line separately could be found in Figure B.5 and Figure B.6. The stacked ensemble (which achieves the highest error in general) performs the best on the complex HepG2 cell line. It could indicate that a stacked ensemble is a useful strategy for more complex cases. In essence, the strategies demonstrate the same pattern of performance as in the case without domain shift.

In conclusion, the master model or naive ensemble strategy should be considered for fluorescence data segmentation when the target and source domains data is sampled from different distributions and introduces domain shift.

4.2.2 Brightfield modality results

Stacked and weighted ensemble strategies achieve almost the same level of performance on the object level (Figure 4.8), but the stacked ensemble has a visibly lower pixel-wise error (Figure 4.7). This supports the statement from the previous chapter, that the stacked ensemble produces better results for complex tasks.

The master model strategy adapts to the cases with domain shift by the joint combination of the data from different cell lines in one model. This approach achieves almost the same level of performance as a weighted ensemble. Moreover, the difference between the error of the stacked ensemble (which is the best) and the master model is 0.05 for pixel- and 0.04 for object-wise errors which are relatively minor.

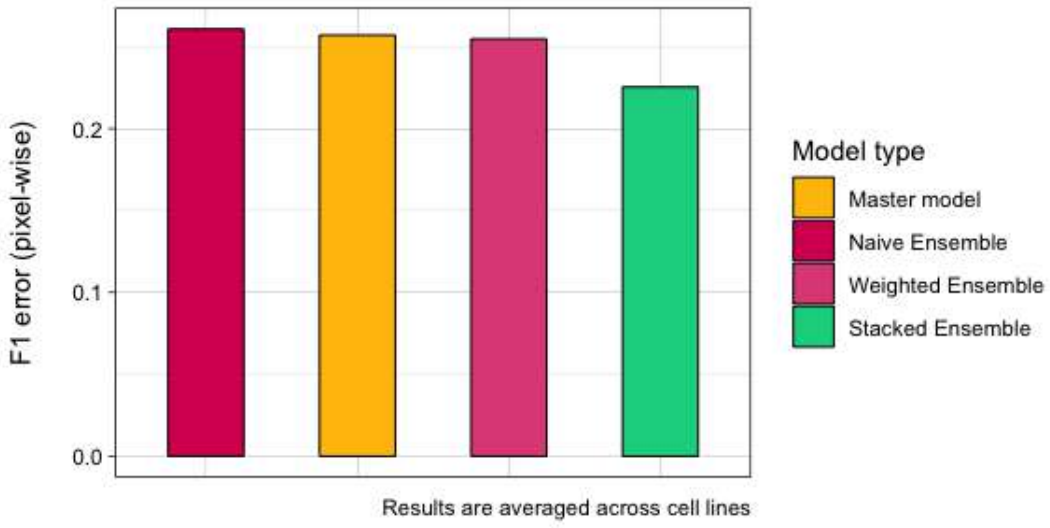

FIGURE 4.7: Stacked ensemble achieves the best pixel-wise results in brightfield modality with domain shift.

The results for each cell line separately could be found in Figure B.7 and Figure B.8 respectively. The general pattern follows the figures presented above. However, as well as for fluorescence modality, the stacked ensemble is significantly notably on a complex HepG2 cell line.

In contrast to the fluorescence modality, the stacked ensemble strategy proves to be the most effective in the brightfield modality when the domain shift is present.

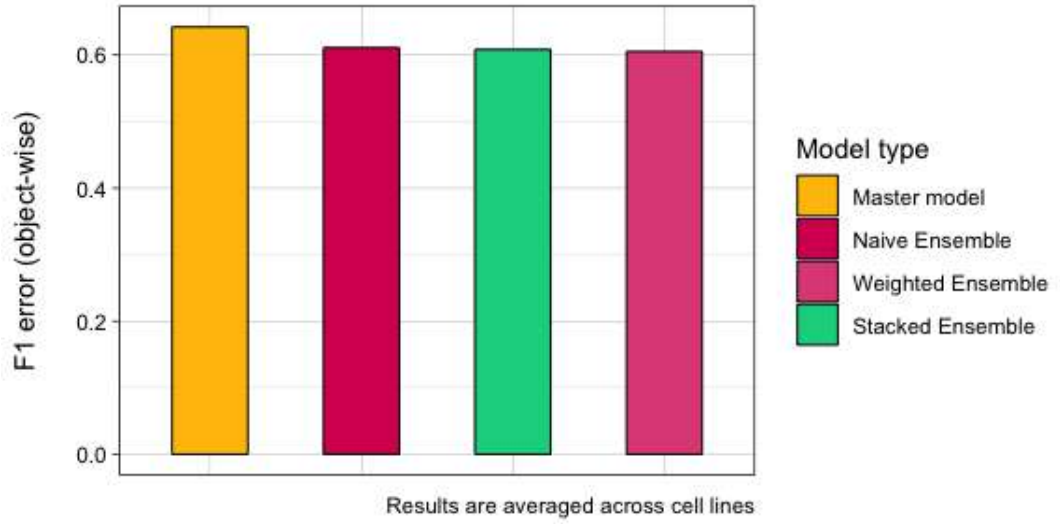

FIGURE 4.8: The weighted ensemble is more robust to domain shift on the object-level.

However, since the stacked ensemble is hard to implement, the master model strategy could be considered due to minor differences in error scores.

4.3 Transfer learning

Results presented in this chapter are focused on brightfield modality as models trained on fluorescent images seem to yield indistinguishable performance levels.

4.3.1 From medical domain with similar data distribution

In this experiment, we advanced the idea from the previous Chapter 4.2. We took models trained on the data from six cell lines (source domain) and used the remaining seventh cell line as the target domain. The difference between domains in this experiment is relatively small as all images come from the same distribution.

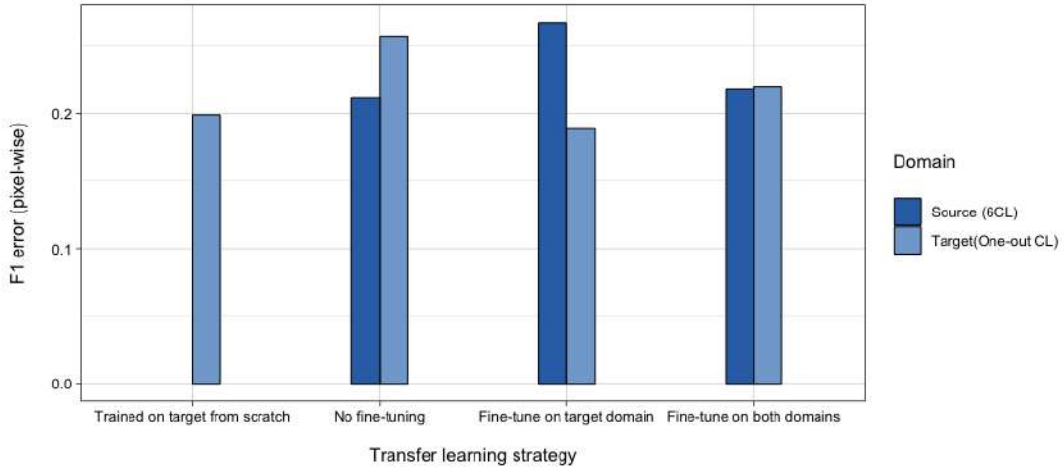

FIGURE 4.9: Fine-tuning on both domains produces slightly worse results than training a separate model for the target domain.

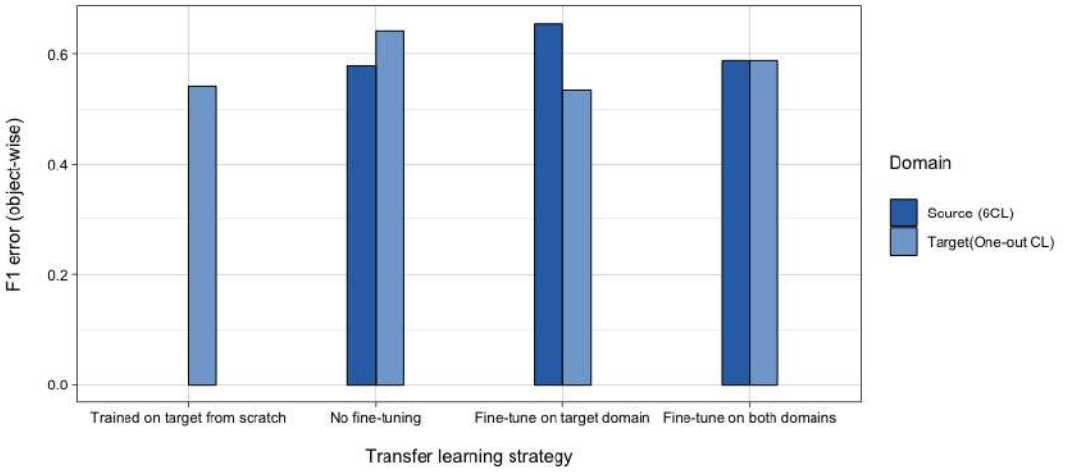

FIGURE 4.10: The object error scores of all strategies are not significantly different from a model trained from scratch.

The model trained on source domain data, without fine-tuning on the target domain, achieves the highest error scores (Figure 4.9 and Figure 4.10). The difference between the errors of the source domain model (no fine-tuning) and the target domain model (trained on target from scratch) is significant with value 0.1.

Fine-tuning only on the target domain allows us to reduce error on the target domain, compared to the model trained from scratch. This happens at the price of much worse model performance on the source domain. However, if performance on the target domain is the goal, this strategy produces better results, compared to training a randomly initialized model.

In contrast to the previous strategy, model fine-tuned on the data from both domains achieves reasonable performance on the target domain and mostly preserves the level of performance on the source domain. This is the most preferable strategy for similar tasks (domains) if one has access to the source domain data originally used for model training.

Additionally, we evaluated how the size of the dataset that is used for fine-tuning affects the performance of the model on both domains. We used the powers of two in the range from 0 to 8 as in the previous Chapter 4.1.

If we fine-tune only on the target domain a significant increase in error on the source domain is noticeable from the very start (Figure D.1 and Figure D.2). Nevertheless, the performance of the model on the target domain using only one image for fine-tuning is significantly better as the models trained from scratch on one image in previous chapters (Figure B.7). However, with the increasing number of images, the performance of the fine-tuned model converges to the same values as the model trained from scratch in the previous chapter. Therefore, fine-tuning only on the target data is reasonable when one has a relatively small amount of labeled data from the target domain.

In contrast, with fine-tuning on both domains the model loses the performance on source domain if we use few images (Figure D.3 and Figure D.4). However, with the increasing amount of images the model restores almost the same performance as originally on the source domain (with the difference of 0.01 for pixel- and 0.006 for object-wise error). Moreover, it continuously gains the performance on the target domain with increasing the number of images. On the peak number of 256 fine-tuning on both domains achieves almost the same level of performance as fine-tuning solely on target domain data.

4.3.2 From medical domain with different data distribution

In this experiment, we used an additional brightfield modality dataset provided by the AstraZeneca company as the source domain. The target domain is represented by seven cell lines dataset used in the previous experiments.

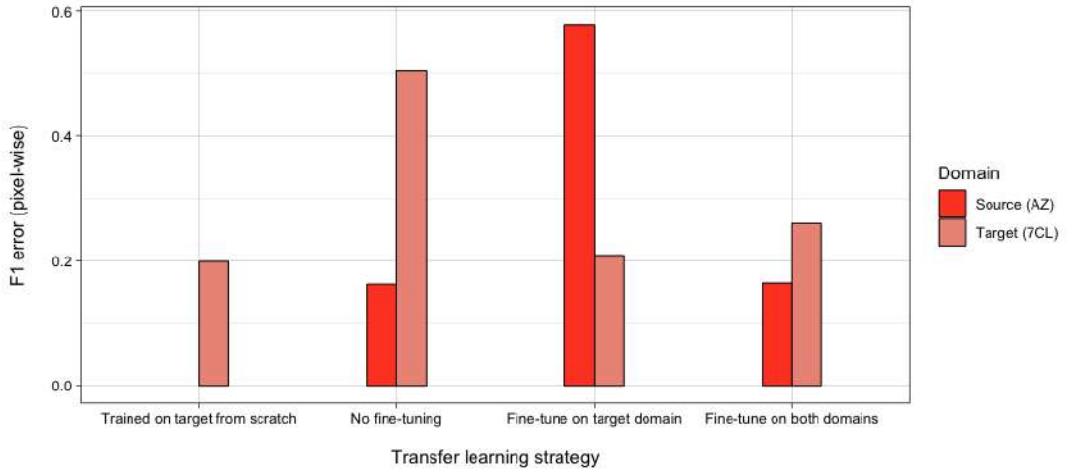

FIGURE 4.11: No fine-tuning provides unsatisfactory results due to the big domain shift.

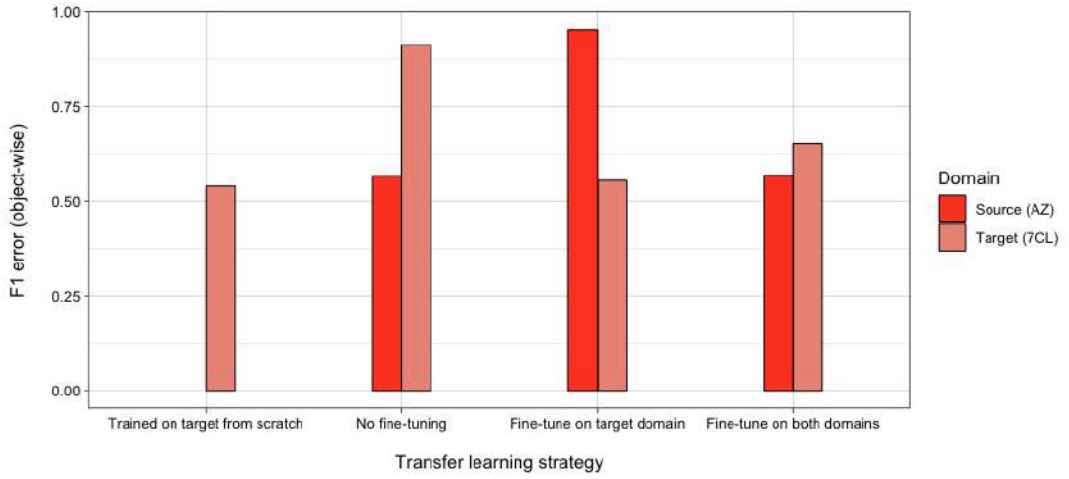

FIGURE 4.12: Fine-tuning only on target produces significant degradation on the source domain.

In contrast to the previous experiment with no domain shift, the source domain model without fine-tuning yields poor results on both pixel- (Figure 4.11) and object-wise (Figure 4.12) levels.

If we fine-tune the source domain model only on target domain data we achieve essentially the same performance as if we would train a target domain model from scratch. The same pattern is observed when no domain shift is present. In contrast, fine-tuning only on the target domain significantly reduces the model performance on the source domain. Therefore, this strategy should be applied if the performance of the model on the target domain is the main goal.

Fine-tuning the domain model on the data from both domains produces satisfactory results. The model preserves the performance on the source domain and

achieves reasonable performance on the target domain with a difference of 0.05, compared to the model trained from scratch.

With respect to the previous experiment, we evaluated how the size of the dataset used for fine-tuning affects the performance of the models on both domains.

The results for the strategy of fine-tuning only on target domain data are presented in Figure D.5 and Figure D.6 for pixel- and object-wise metrics respectively. The performance of the model on the source domain degrades more notable when the tasks are fairly different (compared to the case when tasks were similar). The error score for the source domain increases from 0.18 to 0.495 which means that half of the predictions of the model become incorrect. Moreover, the source model shows slower adaptation to the target domain, than in the case when the domain shift is not present.

Respectively, the results of different dataset size for the strategy of fine-tuning on both domain data are presented in Figure D.7 and Figure D.8 for pixel- and object-wise metrics. The performance of the model on the source domain almost doesn't change whereas the performance on the target domain slowly rises with the increasing number of images. For this strategy, it is preferable to obtain at least 64 or 128 annotated images (from both domains) to produce satisfactory results.

4.3.3 From natural objects domain (ImageNet)

The last part of transfer learning experiments evaluated the effect of transfer learning from the distant domain of natural objects (ImageNet). The results presented below correspond to 2016 images (whole training set) used for fine-tuning or training (in case of U-Net).

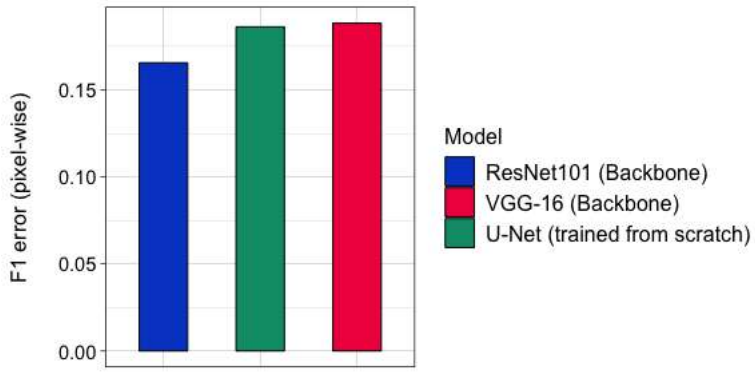

FIGURE 4.13: ResNet101 slightly outperforms U-Net trained from scratch in both pixel and object metrics.

From the results, the ResNet101 as a backbone has the smallest pixel- (Figure 4.13 and object-wise (Figure 4.14) error scores. The U-Net trained from scratch has 0.02 higher pixel and 0.04 higher object errors, than the ResNet101. However, the ResNet-101 system has more than 67.4 million parameters and the VGG-16 system has more than 37.8 million parameters. Whereas the U-Net model used in this experiment has almost 2.3 million parameters. Therefore, it renders the usage and further fine-tuning of large ImageNet models impractical for brightfield segmentation, as much smaller U-Net architecture yields almost the same performance.

ResNet101 gains the performance faster on pixel level with the increasing number of images (Figure D.9), but U-Net and VGG-16 (which demonstrate very similar performance throughout few images) perform better on the object level from the

start (Figure D.10). Additionally, the ResNet101 model demonstrates a surprising peak of error increase on 32 images which decreases thereafter. This is explained by previous work that states that the larget the model the slower it adapts to a new domain [11].

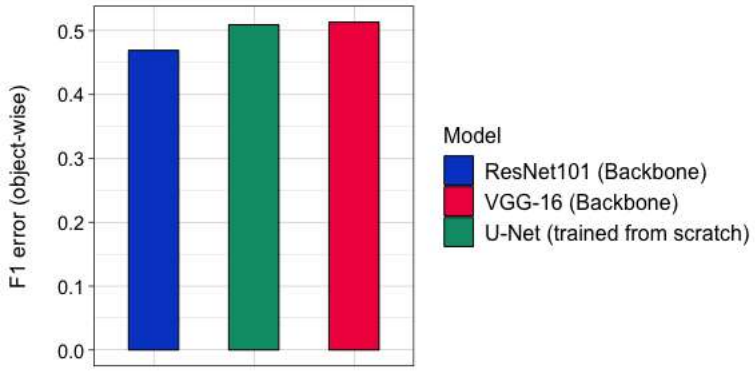

FIGURE 4.14: The VGG-16 yields the worst results for object and pixel errors.

Chapter 5

Conclusions

Microscopy segmentation is a labor-intensive task that requires a significant amount of manual human work. Institutions and private companies are using rule-based and machine learning algorithms to automate this process. However, most of these algorithms lack computational efficiency and produce relatively unsatisfactory results. Moreover, the high heterogeneity of the data introduces additional complexity for both biologists and computer algorithms.

In this thesis, our main focus was on exploring the effects of various possible training strategies and transfer learning approaches in microscopy segmentation.

The training strategies compared in this thesis are: training an individual model for each cell line, training one model jointly on all cell lines and ensemble strategies that combine individual models. We implemented three ensemble approaches: naive (basic averaging), weighted (weighted sum of predictions) and stacked (using an additional model to combine the predictions).

We compared the aforementioned strategies under two conditions: either training data (source domain) and testing data (target domain) are sampled from the same dataset, or from different datasets.

In the first case, naive and stacked ensembles produce noticeably better results but are harder to implement compared to a standalone model (that produces relatively close results). Consequently, a master model (trained jointly on all cell lines) proves to be the most reasonable training strategy for both modalities.

In the second case, the master model demonstrates superior performance in fluorescence whereas stacked ensemble proves to be the most effective strategy in brightfield.

The fluorescence models need fewer annotated images for accurate segmentation, whereas for brightfield modality it is preferable to obtain as much data as possible to produce sufficient performance. Moreover, some cell lines (HepG2 and MCF7) are more heterogeneous than others (MDCK, A549, and HeLa) and cause the models to produce a higher number of errors.

Transfer learning from a similar or different medical task showed to be an effective strategy that requires less annotated data to obtain the same or superior performance compared to a randomly initialized model. Fine-tuning the model on the data from both domains is the preferred strategy as it preserves most of the performance on source and yields reasonable performance on the target domain.

Transferring weights from a distant domain of natural objects (ImageNet) seem to be inefficient as it requires significantly bigger models (compared to the U-Net model used throughout the work) to produce slightly better results.

Further work could consider performing the aforementioned experiments with different deep artificial network architectures. Moreover, one of the possible improvements is to perform transfer learning from brightfield to fluorescent modality and vice versa.

Appendix A

Data distribution

Cell line	Training set size	Validation set size	Testing set size	Total
A549	286	66	80	432
HT1080	284	78	70	432
HeLa	293	58	81	432
HepG2	283	82	67	432
MCF7	290	70	72	432
MDCK	292	79	61	432
NIH3T3	288	71	73	432
Summary	**2016**	**504**	**504**	**3024**

TABLE A.1: Distribution of the data in primary dataset from PerkinEllmer

Training set size	Validation set size	Testing set size	Total
628	78	78	**784**

TABLE A.2: Distribution of the data in AstraZeneca dataset

Appendix B

Performance on separate cell lines

B.1 Source and target domain from the same distribution

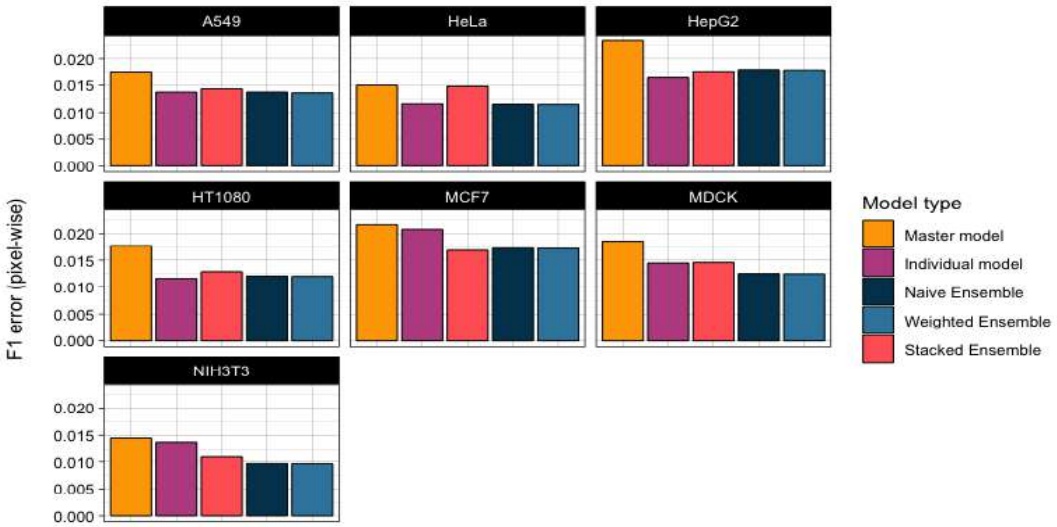

FIGURE B.1: Pixel-wise performance of the strategies in fluorescence modality (no domain shift).

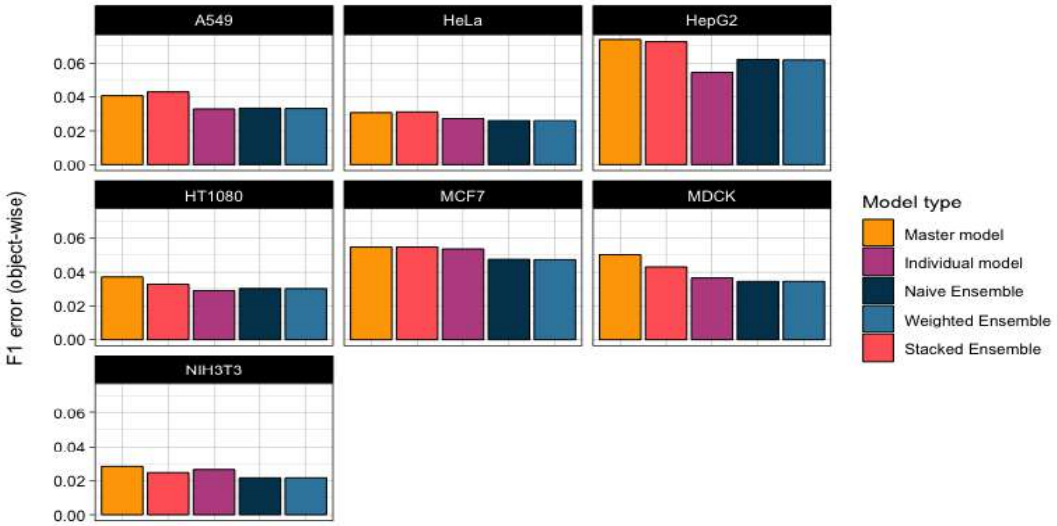

FIGURE B.2: Object-wise performance of the strategies in fluorescence modality (no domain shift).

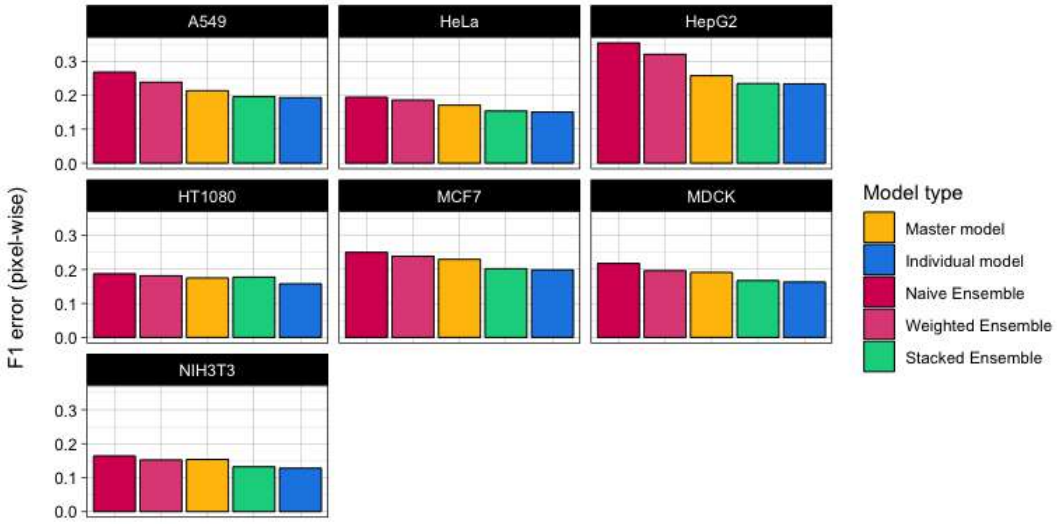

FIGURE B.3: Pixel-wise performance of the strategies in brightfield modality (no domain shift).

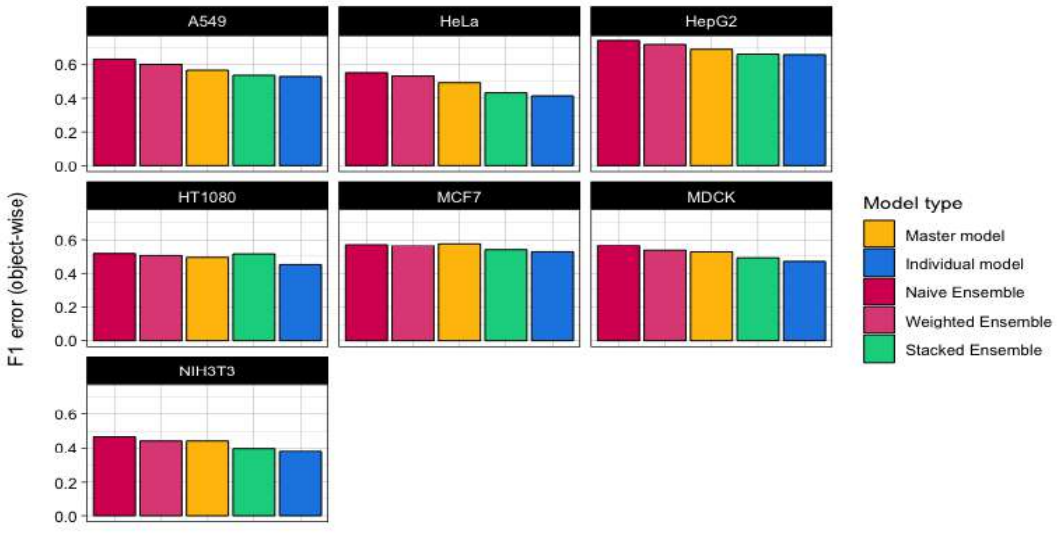

FIGURE B.4: Object-wise performance of the strategies in brightfield modality (no domain shift).

B.2 Source and target domain from different distributions

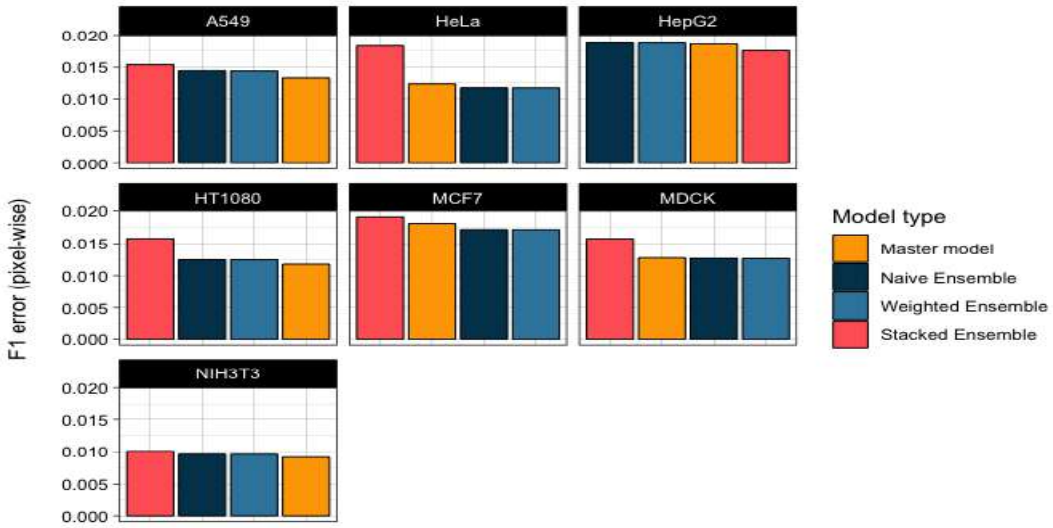

FIGURE B.5: Pixel-wise performance of the strategies in fluorescence modality (domain shift present).

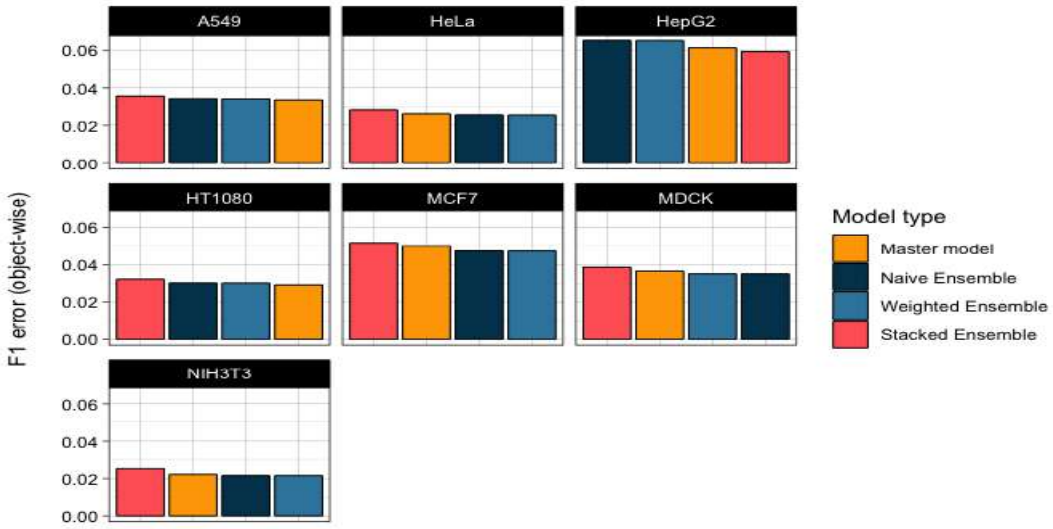

FIGURE B.6: Object-wise performance of the strategies in fluorescence modality (domain shift present).

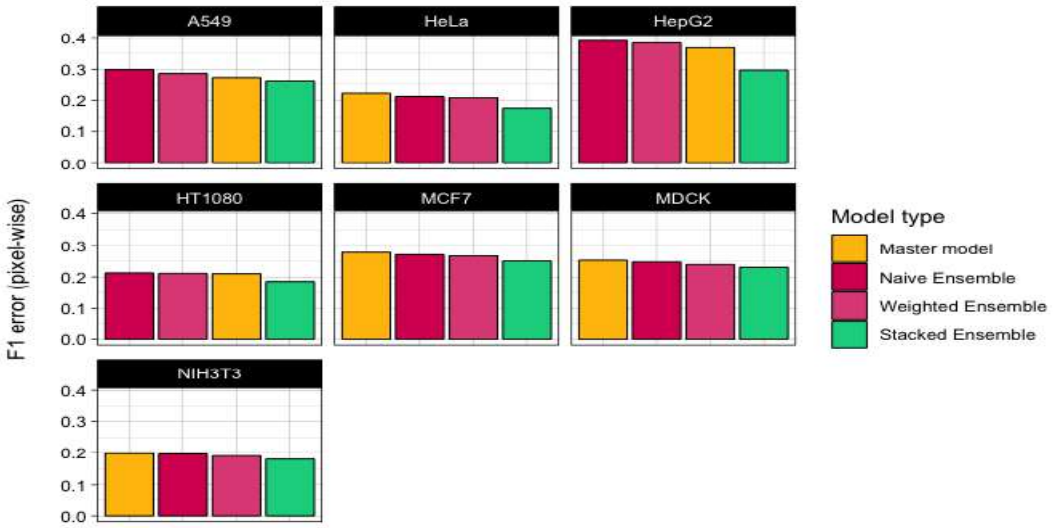

FIGURE B.7: Pixel-wise performance of the strategies in brightfield modality (domain shift present).

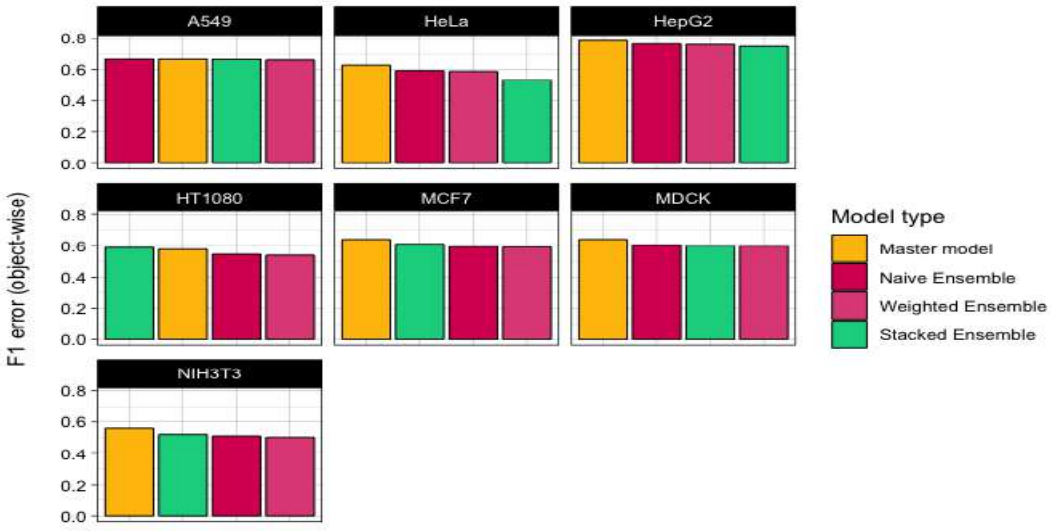

FIGURE B.8: Object-wise performance of the strategies in brightfield modality (domain shift present).

Appendix C

The effect of training set size

C.1 Fluorescence modality

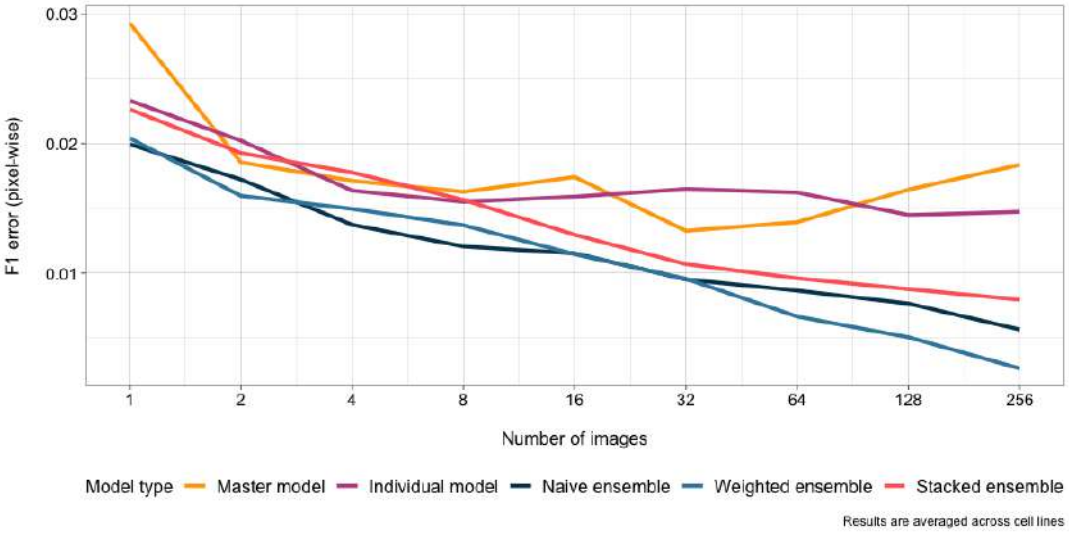

FIGURE C.1: Effect of training set size on pixel-wise performance of the fluorescence models.

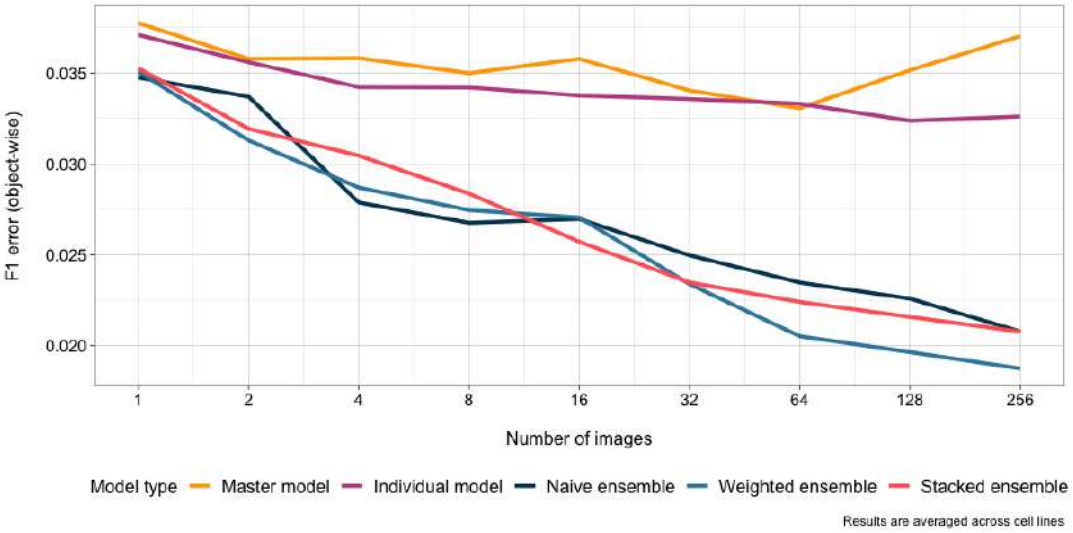

FIGURE C.2: Effect of training set size on object-wise performance of the fluorescence models.

C.2 Brightfield modality

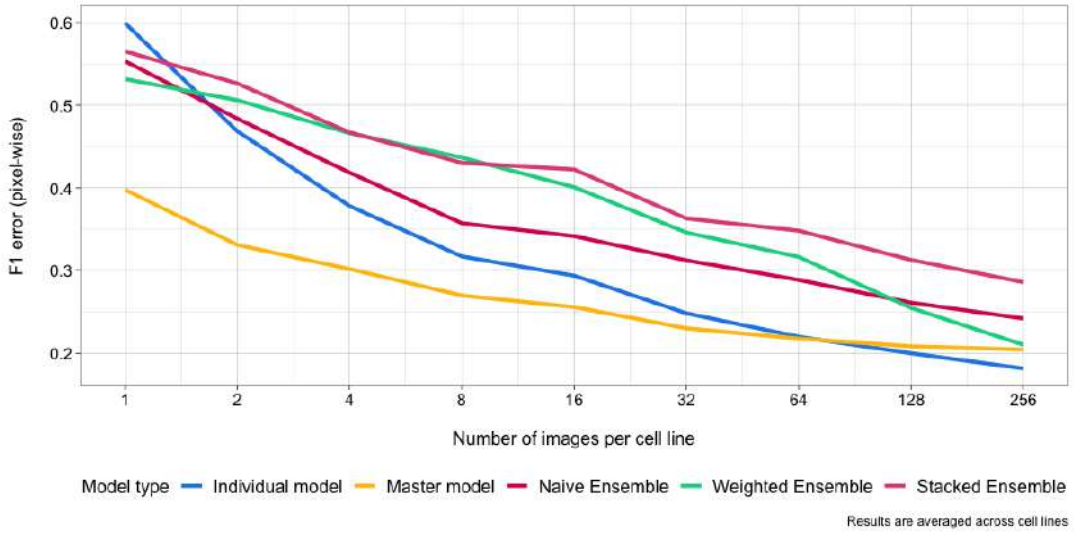

FIGURE C.3: Effect of training set size on pixel-wise performance of the brightfield models.

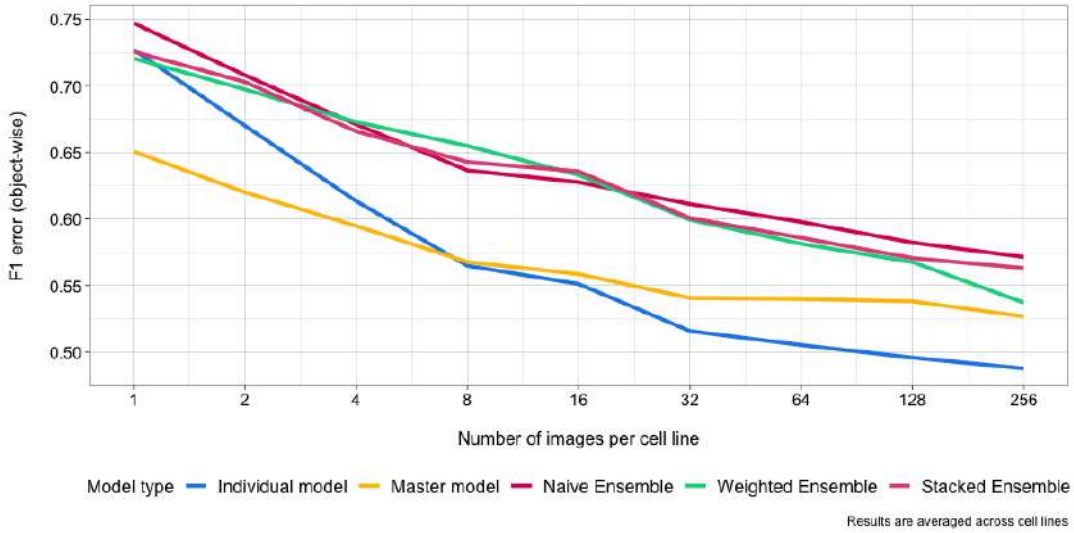

FIGURE C.4: Effect of training set size on object-wise performance of the brightfield models.

Appendix D

Effect of fine-tuning dataset size

D.1 [Similar domains] Fine-tune only on target domain

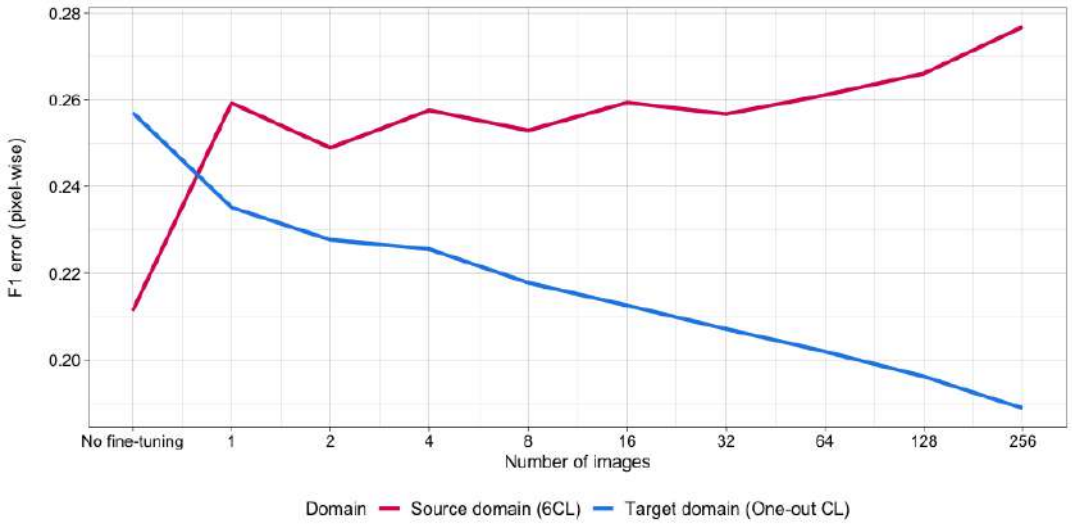

FIGURE D.1: Pixel-wise metrics for fine-tuning on target with different number of images (no domain shift).

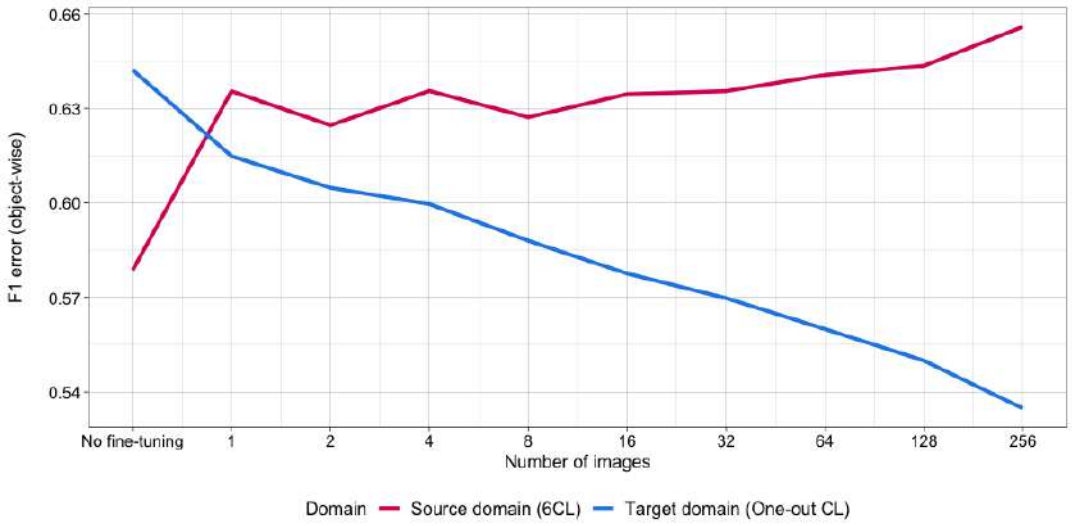

FIGURE D.2: Object-wise metrics for fine-tuning on target with different number of images (no domain shift).

D.2 [Similar domains] Fine-tune on both target and source domains

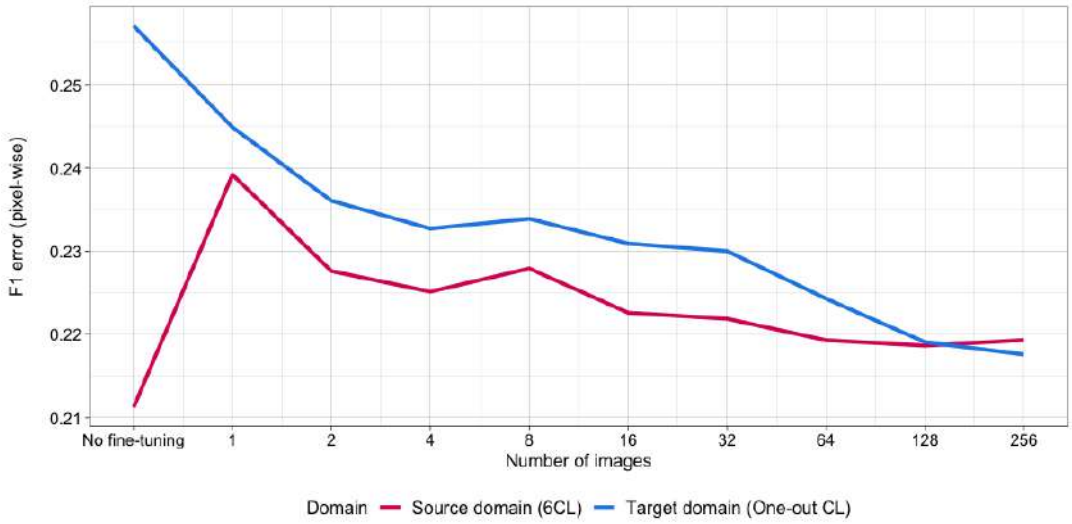

FIGURE D.3: Pixel-wise metrics for fine-tuning on both domains with different number of images (no domain shift).

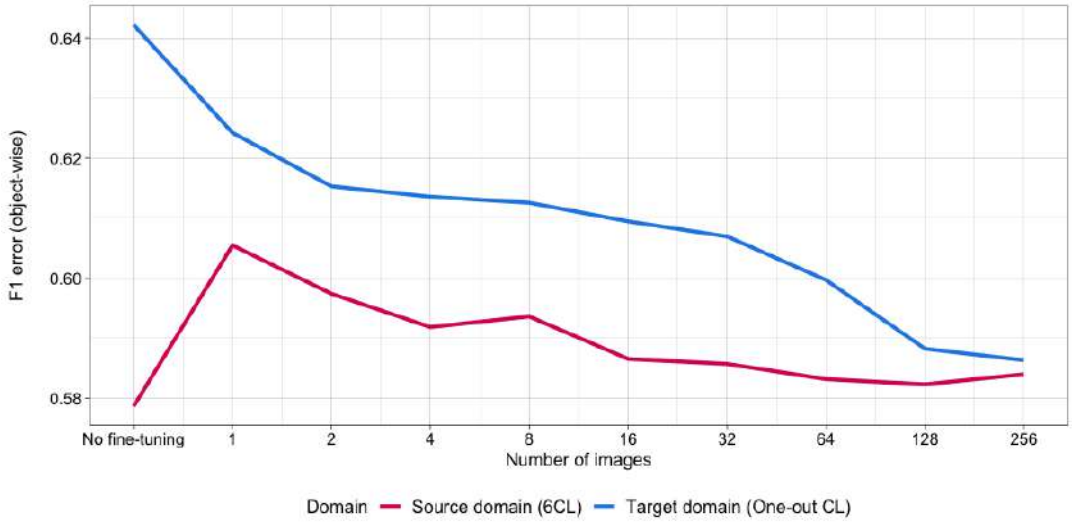

FIGURE D.4: Object-wise metrics for fine-tuning on both domains with different number of images (no domain shift).

D.3 [Different domains] Fine-tune only on target domain

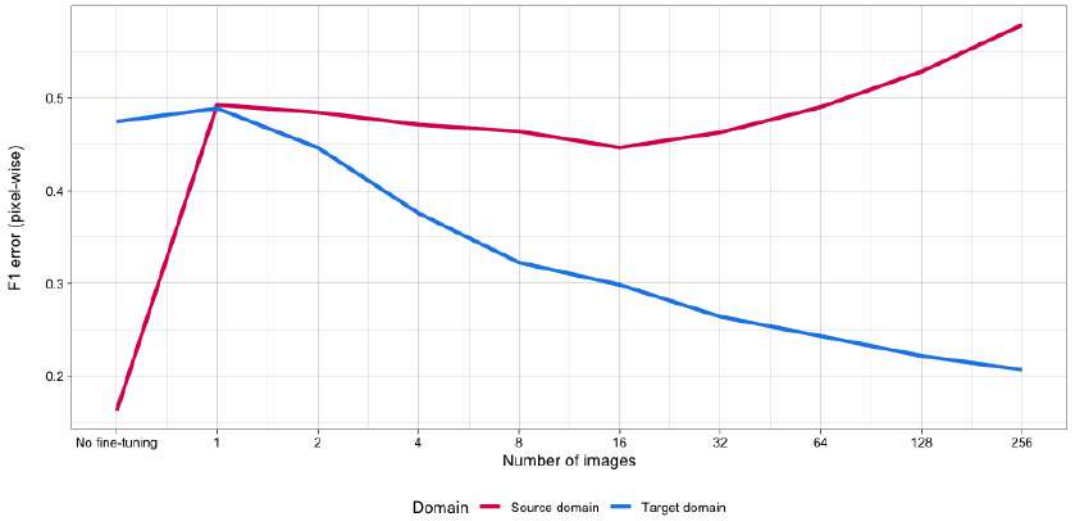

FIGURE D.5: Pixel-wise metrics for fine-tuning on target with different number of images (domain shift present).

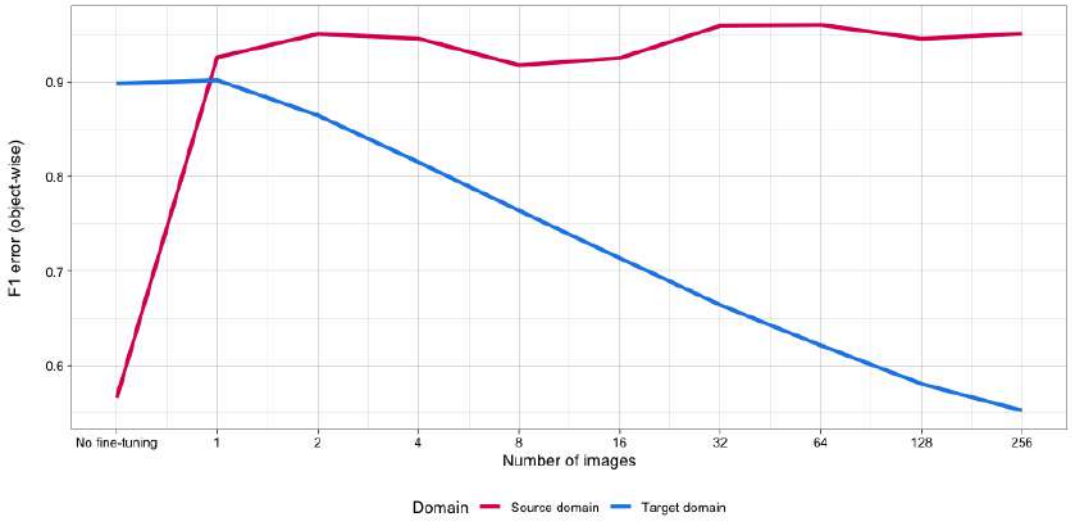

FIGURE D.6: Object-wise metrics for fine-tuning on target with different number of images (domain shift present).

D.4 [Different domains] Fine-tune on both target and source domains

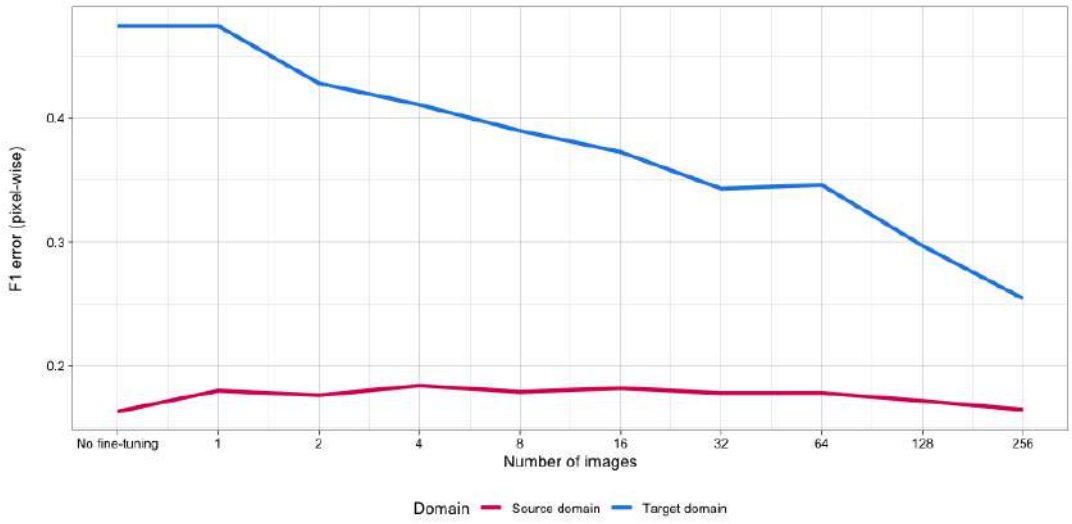

FIGURE D.7: Pixel-wise metrics for fine-tuning on both domains with different number of images (domain shift present).

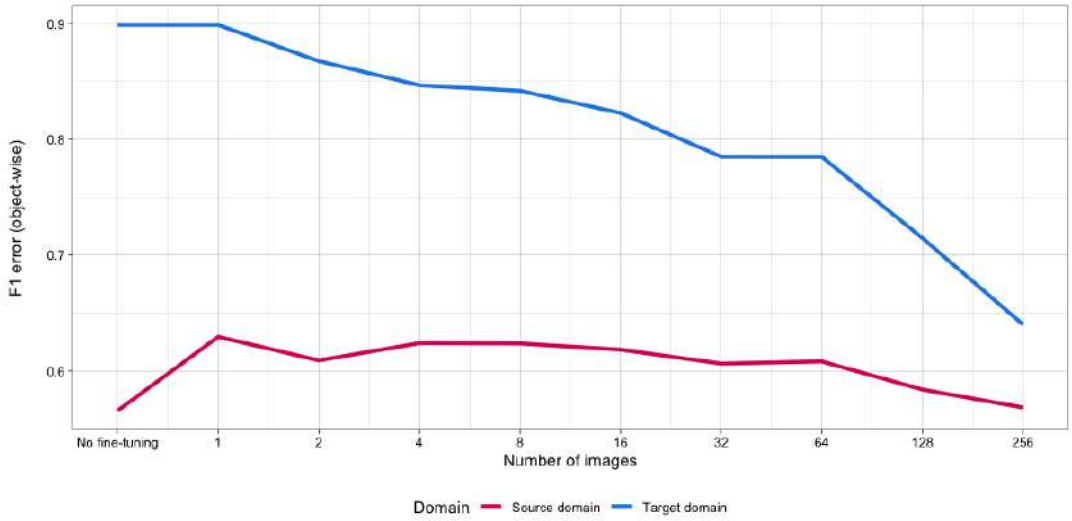

FIGURE D.8: Object-wise metrics for fine-tuning on both domains with different number of images (domain shift present).

D.5 [Distant domain] Effect of fine-tune and train set size

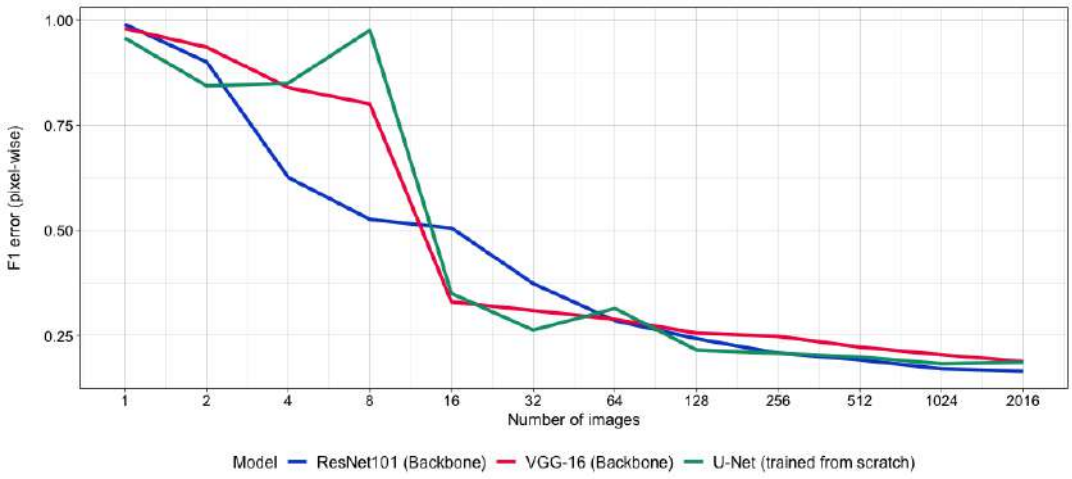

FIGURE D.9: Pixel-wise metrics for fine-tuning distant domain model with different number of images.

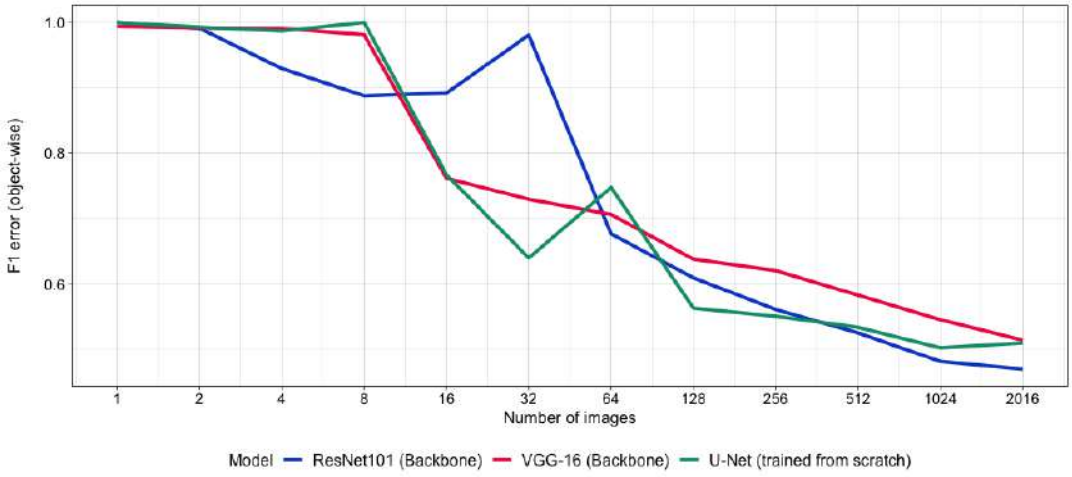

FIGURE D.10: Object-wise metrics for fine-tuning distant domain model with different number of images.

Bibliography

[1] N. Bougen-Zhukov, S. Y. Loh, H. K. Lee, and L.-H. Loo. "Large-scale image-based screening and profiling of cellular phenotypes". en. In: *Cytometry A* 91.2 (Feb. 2017), pp. 115–125.

[2] J. C. Caicedo, S. Singh, and A. E. Carpenter. "Applications in image-based profiling of perturbations". en. In: *Curr. Opin. Biotechnol.* 39 (June 2016), pp. 134–142.

[3] *Nikon: Microscopy imaging techniques*. https://www.nikon.com/products/microscope-solutions/explore/microscope-abc/imaging/index.htm. Accessed: 2020-01-02.

[4] *Microscopy Imaging Techniques*. https://www.microscopemaster.com/microscopy-imaging-techniques.html. Accessed: 2020-01-02.

[5] *Wikipedia Medical Imaging*. https://en.wikipedia.org/wiki/Medical_imaging. Accessed: 2019-09-30.

[6] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. "Simultaneous Detection and Segmentation". In: (July 2014). arXiv: 1407.1808 [cs.CV].

[7] H.-C. Shin et al. "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning". en. In: *IEEE Trans. Med. Imaging* 35.5 (May 2016), pp. 1285–1298.

[8] W. Alomoush et al. "A survey: Challenges of image segmentation based fuzzy c-means clustering algorithm". In: *Journal of Theoretical and Applied Information Technology* 96 (Aug. 2018), p. 18.

[9] N. Tajbakhsh et al. "Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?" en. In: *IEEE Trans. Med. Imaging* 35.5 (May 2016), pp. 1299–1312.

[10] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. "How transferable are features in deep neural networks?" In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., 2014, pp. 3320–3328. URL: http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf.

[11] M. Raghu, C. Zhang, J. M. Kleinberg, and S. Bengio. "Transfusion: Understanding Transfer Learning with Applications to Medical Imaging". In: *CoRR* abs/1902.07208 (2019). arXiv: 1902.07208. URL: http://arxiv.org/abs/1902.07208.

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

[13] S. Beucher. "Use of watersheds in contour detection". In: *Proceedings of the International Workshop on Image Processing*. CCETT. 1979.

[14] D. Vayada. *Intuitive image processing — Watershed segmentation*. https://medium.com/@dhairya.vayada/intuitive-image-processing-watershed-segmentation-50a66ed2352e. Accessed: 2019-11-28.

[15] D. L. Pham, C Xu, and J. L. Prince. "Current methods in medical image segmentation". en. In: *Annu. Rev. Biomed. Eng.* 2 (2000), pp. 315–337.

[16] H. Bindu and K. Prasad. "An Efficient Medical Image Segmentation Using Conventional OTSU Method". In: *International Journal of Advanced Science and Technology* 38 (Jan. 2012).

[17] J. C. Bezdek, L. O. Hall, and L. P. Clarke. "Review of MR image segmentation techniques using pattern recognition". en. In: *Med. Phys.* 20.4 (July 1993), pp. 1033–1048.

[18] T. Kapur et al. "Enhanced spatial priors for segmentation of magnetic resonance imagery". In: *Medical Image Computing and Computer-Assisted Intervention — MICCAI'98*. Springer Berlin Heidelberg, 1998, pp. 457–468.

[19] A. F. Goldszal et al. "An image-processing system for qualitative and quantitative volumetric analysis of brain images". en. In: *J. Comput. Assist. Tomogr.* 22.5 (Sept. 1998), pp. 827–837.

[20] D. L. Pham and J. L. Prince. "An adaptive fuzzy C-means algorithm for image segmentation in the presence of intensity inhomogeneities". In: *Pattern Recognit. Lett.* 20.1 (Jan. 1999), pp. 57–68.

[21] D. L. Pham, C. Xu, and J. L. Prince. "A Survey of Current Methods in Medical Image Segmentation". In: 1999.

[22] G. Litjens et al. "A survey on deep learning in medical image analysis". In: *Medical Image Analysis* 42 (2017), pp. 60 –88. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2017.07.005. URL: http://www.sciencedirect.com/science/article/pii/S1361841517301135.

[23] L. Perez and J. Wang. "The Effectiveness of Data Augmentation in Image Classification using Deep Learning". In: *CoRR* abs/1712.04621 (2017). arXiv: 1712.04621. URL: http://arxiv.org/abs/1712.04621.

[24] S. Zheng, Y. Song, T. Leung, and I. Goodfellow. "Improving the Robustness of Deep Neural Networks via Stability Training". In: *CVPR'2016*. 2016.

[25] G. J. S. Litjens et al. "A Survey on Deep Learning in Medical Image Analysis". In: *CoRR* abs/1702.05747 (2017). arXiv: 1702.05747. URL: http://arxiv.org/abs/1702.05747.

[26] H. Bergwerf. "Nerve fiber tracing in bright-field images of human skin using deep learning". PhD thesis. June 2018. DOI: 10.13140/RG.2.2.17593.88168.

[27] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597 (2015). arXiv: 1505.04597. URL: http://arxiv.org/abs/1505.04597.

[28] D. Bannon et al. "DeepCell 2.0: Automated cloud deployment of deep learning models for large-scale cellular image analysis". In: *bioRxiv* (2018). DOI: 10.1101/505032. eprint: https://www.biorxiv.org/content/early/2018/12/22/505032.full.pdf. URL: https://www.biorxiv.org/content/early/2018/12/22/505032.

[29] C. McQuin et al. "CellProfiler 3.0: Next-generation image processing for biology". In: *PLOS Biology* 16.7 (July 2018), pp. 1–17. DOI: 10.1371/journal.pbio.2005970. URL: https://doi.org/10.1371/journal.pbio.2005970.

[30] D. Fishman et al. "Segmenting nuclei in brightfield images with neural networks". In: *bioRxiv* (2019). DOI: 10.1101/764894. eprint: https://www.biorxiv.org/content/early/2019/09/10/764894.full.pdf. URL: https://www.biorxiv.org/content/early/2019/09/10/764894.

[31] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. "Mask R-CNN". In: *CoRR* abs/1703.06870 (2017). arXiv: 1703.06870. URL: http://arxiv.org/abs/1703.06870.

[32] M. D. Zeiler and R. Fergus. "Visualizing and Understanding Convolutional Networks". In: *CoRR* abs/1311.2901 (2013). arXiv: 1311.2901. URL: http://arxiv.org/abs/1311.2901.

[33] *Caltech Pedestrian Detection Benchmark*. http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians. Accessed: 2019-11-21.

[34] *Fast.ai fine-tuning deep neural networks*. http://wiki.fast.ai/index.php/Fine_tuning. Accessed: 2019-11-21.

[35] S. J. Pan and Q. Yang. "A Survey on Transfer Learning". In: *IEEE Trans. on Knowl. and Data Eng.* 22.10 (Oct. 2010), pp. 1345–1359. ISSN: 1041-4347. DOI: 10.1109/TKDE.2009.191. URL: https://doi.org/10.1109/TKDE.2009.191.

[36] X. Wang et al. "ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases". In: *CoRR* abs/1705.02315 (2017). arXiv: 1705.02315. URL: http://arxiv.org/abs/1705.02315.

[37] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[38] P. Rajpurkar et al. "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning". In: *CoRR* abs/1711.05225 (2017). arXiv: 1711.05225. URL: http://arxiv.org/abs/1711.05225.

[39] G. Huang, Z. Liu, and K. Q. Weinberger. "Densely Connected Convolutional Networks". In: *CoRR* abs/1608.06993 (2016). arXiv: 1608.06993. URL: http://arxiv.org/abs/1608.06993.

[40] L. Yao et al. "Learning to diagnose from scratch by exploiting dependencies among labels". In: *CoRR* abs/1710.10501 (2017). arXiv: 1710.10501. URL: http://arxiv.org/abs/1710.10501.

[41] M. Abramoff et al. "Improved Automated Detection of Diabetic Retinopathy on a Publicly Available Dataset Through Integration of Deep Learning". In: *Investigative Ophthalmology and Visual Science* 57 (Oct. 2016), pp. 5200–5206. DOI: 10.1167/iovs.16-19964.

[42] C. Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *CoRR* abs/1512.00567 (2015). arXiv: 1512.00567. URL: http://arxiv.org/abs/1512.00567.

[43] *PerkinElmer - For the better*. https://www.perkinelmer.com/. Accessed: 2019-11-21.

[44] *AstraZeneca - Research-Based BioPharmaceutical Company*. https://www.astrazeneca.com/. Accessed: 2019-11-21.

[45] M. Ghafoorian et al. "Transfer Learning for Domain Adaptation in MRI: Application in Brain Lesion Segmentation". In: *CoRR* abs/1702.07841 (2017). arXiv: 1702.07841. URL: http://arxiv.org/abs/1702.07841.

[46] W. Yan et al. *The Domain Shift Problem of Medical Image Segmentation and Vendor-Adaptation by Unet-GAN*. 2019. arXiv: 1910.13681 [eess.IV].

[47] A. Kumar et al. "An Ensemble of Fine-Tuned Convolutional Neural Networks for Medical Image Classification". In: *IEEE Journal of Biomedical and Health Informatics* 21 (Jan. 2017), pp. 31–40. DOI: 10.1109/JBHI.2016.2635663.

[48] T. G. Dietterich. "Ensemble Methods in Machine Learning". In: *Proceedings of the First International Workshop on Multiple Classifier Systems*. MCS '00. London, UK, UK: Springer-Verlag, 2000, pp. 1–15. ISBN: 3-540-67704-6. URL: http://dl.acm.org/citation.cfm?id=648054.743935.

[49] *Medium Boosting, bagging and stacking ensemble methods with sklearn and mlens*. https://medium.com/@rrfd/boosting-bagging-and-stacking-ensemble-methods-with-sklearn-and-mlens-a455c0c982de. Accessed: 2019-12-15.

[50] H. Zheng et al. "A New Ensemble Learning Framework for 3D Biomedical Image Segmentation". In: (Dec. 2018). arXiv: 1812.03945 [cs.CV].

[51] X Glorot and Y Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference* (2010).

[52] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. "On the importance of initialization and momentum in deep learning". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by S. Dasgupta and D. McAllester. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, 2013, pp. 1139–1147. URL: http://proceedings.mlr.press/v28/sutskever13.html.

[53] A. F. Agarap. "Deep Learning using Rectified Linear Units (ReLU)". In: (Mar. 2018). arXiv: 1803.08375 [cs.NE].

[54] D. P. Kingma and J. Ba. "Adam: A Method for Stochastic Optimization". In: (Dec. 2014). arXiv: 1412.6980 [cs.LG].

[55] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014). URL: http://arxiv.org/abs/1409.1556.