UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

Incorporating Metadata for Semantic Segmentation by employing Channel Attention Mechanism

Author:
Iaroslav PLUTENKO

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 2021

Declaration of Authorship

I, Iaroslav PLUTENKO, declare that this thesis titled, "Incorporating Metadata for Semantic Segmentation by employing Channel Attention Mechanism" 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: _____

UKRAINIAN CATHOLIC UNIVERSITY

Faculty of Applied Sciences

Master of Science

**Incorporating Metadata
for Semantic Segmentation
by employing
Channel Attention Mechanism**

by Iaroslav PLUTENKO

Abstract

The meta-information accompanying data from image acquisition devices has limited use in microscopy image processing techniques involving Deep Learning. This project aims to incorporate the supplementary metadata for semantic segmentation by employing a channel selection mechanism in convolutional networks outlining its potential benefits and practical applications where metadata can be used for switching tasks within a master model. The results of conducted experiments show that meta-information is helpful, and the phenomenon is more expressed with incompatible segmentation tasks, where a multi-head model or separate models are required otherwise. Overall, we have achieved a slight increase in scores for similar tasks as well and demonstrated the applicability of the CNN model for separate tasks, forcing it to work as an ensemble, leveraging the beneficial effect of multi-task learning.

Acknowledgements

The research became possible thanks to the Credit mobility traineeship within Erasmus+ program funded by the EU and thanks to the enthusiastic collaboration between the University of Tartu and the Ukrainian Catholic University. Special thanks to *PerkinElmer, Inc* for the provided samples and counseling. The praise for organizational skills should be given to Academic Program Director Olexii Molchanovskyi.

I'm grateful for the support and comments from the goal-oriented team consisting of students and researchers of the Institute of Computer Science and PerkinElmer experts, namely Mohammed Ali, Sten-Oliver Salumaa, Kaupo Palo, Leopold Parts, Tõnis Laasfeld, Hartwig Preckel, Olavi Ollikainen, Dmytro Urukov, Kaspar Hollo, and Mikhail Papkov. I'm obliged to the latter for attracting my attention to Squeeze and Excitation techniques that determined the vector of my research.

And the special gratitude is reserved for talented young scientist Dmytro Fishman, whose experience, devotion to the scientific quest, and orchestration abilities provided guidance in my experiments and facilitated interaction with team members.

Contents

List of Figures

List of Tables

List of Abbreviations

CA	**C**hannel **A**ttention
CNN	**C**onvolutional **N**eural **N**etwork
GPU	**G**raphics **P**rocessing **U**nit
NN	**N**eural **N**etwork
MLNN	**M**eta **L**earning **N**eural **N**etwork
DL	**D**eep **L**earning
RAM	**R**andom **A**ccess **M**emory
SOTA	**S**tate **O**f **T**he **A**rt

Chapter 1

Introduction

This chapter provides a brief historical background for the semantic segmentation task and highlights the typical challenges the scientists and users face in the biomedical field of research and industry. We will also disclose our motivation and general objectives for using meta-information more actively to facilitate achieving better results in the processing of microscopy cell images.

1.1 Background

The machine-aided image processing and analysis of biomedical data have existed since the invention of digitalization and image acquisition techniques in medicine and microscopy. Early software methods of image processing focused primarily on image enhancement. Gradually with the development of Machine Learning (ML), the scope of tasks and applicable methods became more sophisticated, aiming for a partial or full replacement of stages handled by human experts in the image analysis workflow. The typical tasks in biomedical image processing are semantic segmentation (classifying each pixel belonging to two or more classes), object detection (lesions, tumors, abnormal formations, cell parts, etc.) and counting, or more advanced categorization resulting in medical diagnosis based on all visual data.

The essential skill for microbiologists was to segment microscopy images, manually painting regions of different cell types and their content. Durable and tedious work required a profound experience and strained all the attention of the expert to the limits. ML helped to automate a large part of it. The oldest techniques applied were "rule-based" routines like thresholding (Otsu, 1979) and watershed (Beucher, 1979) that separated regions using fixed or adaptive threshold. Despite labeled as outdated, these methods remain nonetheless applicable in modern image processing workflows as main or auxiliary steps, praised for high computational speed and simplicity. Then classic ML methods came to the industry: regression and clusterization techniques based on pixel neighboring analysis (Pham, Xu, and Prince, 2000; Bezdek, Hall, and Clarke, 1993). However, these techniques, despite being fast and tractable, are neither universal nor accurate enough. They also require manual adjustments, feature engineering, and rarely are transferable beyond the specific area. The transfer requires significant reprogramming of the software kit.

With the advent of Deep Learning (DL), Neural Nets (NN) gained recognition and a large share of medical and microscopy imaging applications. The Convolutional Neural Nets (CNN) and U-Net (Ronneberger, Fischer, and Brox, 2015) in particular became de facto a standard and a backbone for a plethora of working and experimental NN topologies and processing solutions. Such networks offer more

flexibility and robustness, eliminate the manual feature engineering stage and increase the accuracy for segmentation tasks comparing to preceding ML models (Litjens et al., 2017; Fishman et al., 2019). However, NNs still rely on human-annotated data during the training phase to achieve superior performance.

1.2 Project conception

One of the challenges typical for supervised learning is the high dependency on the training data, overfitting, and domain specificity. When the NN model trains on one type of data, it performs well on such type later but deteriorates when the new input data type differs from the trained one, the situation obtained the term *domain shift*. For the biomedical images, a different domain may be represented by images produced by various methods, other devices, or the same device with different settings. Major domain shift should be prevented by including a significant share of samples from the target domain into training data.

Needless to say, it is not always possible. The underrepresented or absent domain in training data makes predictions on this data for the model difficult. Nevertheless, a broad set of solutions addressing the domain shift exists, commonly termed *domain adaptation*. Domain adaptation encompasses various methods; one of the most studied nowadays is transfer learning (Pan and Yang, 2010). The model initially trained on one set of samples is trained afterward with the inclusion of the samples from the required domain - the process is referred to as fine-tuning. The common usage of transfer learning is the reduction of time needed for training: researchers take models already pretrained on the public dataset like *ImageNet* and then continue fine-tuning on field-specific data. Transfer learning for the task of semantic segmentation in biomedical imaging has its own specifics and complications (Ghafoorian et al., 2017; Raghu et al., 2019).

However, transfer learning is not among the objects of our research. Since we use similar data distributions during the training and inference time, we do not address domain shift in current research but rather the domain specificity. Thus, the usage of terms *domain* and *task* in the current thesis might also be a source of confusion as many authors tie these terms to transfer learning and provide definitions exclusively in such context (Pan and Yang, 2010). However, this terminology usage is broader and often less specific; a *domain* in our work is a synonym of a subset of data with feature distributions specific only to such subset. The same applies to the usage of the term *task* - in the current document, it refers to the segmentation of different objects from the same or similar source images, but in the experiment with heterogeneous domains, it refers to separate workflows as well. It's enough to claim that we will have multiple domains in our experiments, but the domains of the training set are not going to change during the inference time. There will also be no task transitions since we will not make the model learn a new task after initial training in the experiments.

Following these considerations, two corner strategies for designing an effective system working in a cross-domain environment are: training a single master model on a large diverse dataset and introducing multiple models: an individual ("expert") model for each domain separately. Our objective should reside somewhere in between these cases, incorporating master model properties and domain separation within a single system. Domain-specific models or subsystems should preserve the highest accuracy domain-wise and often serve as a ground-truth configuration when assessing the performance of competing systems. The decision of which model to

use is up to the user(operator), though uniting them in an ensemble is preferable for convinience and for the additional advantages offered by ensembles over the individual classifiers (Dietterich, 2000). In the literature, such ensembles of individual experts dividing the problem space is referred to as a Mixture of Experts (Masoudnia and Ebrahimpour, 2014). There were reports of successful implementation of meta-learning with ensembles in biomedical images segmentation (Zheng et al., 2019). Intuitively we expect that individual models perform better on respective subsets than a master model, especially when the distance between domains is significant. We find confirmations in earlier studies when generalization comes with decreased accuracy comparing to the domain-specific system (Misko, 2020). However, there is little evidence to prove this should be a rule of thumb, and we must state that it is just an assumption that will not always hold true in our experiments.

Microscopy bitmap images come with abundant meta-information depending on data acquisition devices or the methodology of samples preparation, in the industry this information is rarely used directly in NN training. The examples of metadata are given below with short comments:

- *exposure*: discrete integer values corresponding to the sensor activation time, affecting image brightness, contrast, color rendering and many other image properties directly and indirectly;
- *plane*: binary in biplane microscopy (upper/lower), affecting the quality (focus/blur);
- *magnification*: ordinal values (1.25x, 5x, 10x, 20x, 40x, 64x) which can be treated as numerical, affecting the bitmap features size;
- *imaging modality*: nominal values (Fluorescent, Brightfield, DPC-reconstruction), greatly affecting the image representation;
- *imaging condition(medium)*: binary(Air/Water), slightly affecting general properties, quality, and focus/blur;
- *imaging condition(focal)*: binary(focal/non-focal), affecting the image representation and quality (focus/blur).

In the context of our project, meta-information marks the subset of data in a discrete fashion. Corollary, the more specific is meta-information, the narrower subset it represents. Oftentimes we will use only a cell line name as meta-label to denote the origin of the subset of samples. Continuous variables from meta-information were not included in the current research, but obviously, in the light of the above definitions, they should be discretised to represent some subset within a range of values. In the concluding set of experiments, meta-information will serve as an attribute of a specific task.

We are looking for a semantic segmentation system still embedding the CNN as a primary engine, operating on bitmap microscopy images and incorporating sparse metadata to aid segmentation in diverse domain conditions. Every improvement in domain specialisation comparing to the generalized metadata-free model would be welcome.

Thus we can outline the main objectives of our research as follows:

1. Can meta-information as a domain marker help increase the NN performance in the task of semantic segmentation?

2. How can we incorporate meta-information in existing CNN models?

3. What might be the practical applications and benefits of such a system beyond the context of domain specialization?

The rest of the current thesis structure is organized as follows. In Chapter 2, we overview the ways to incorporate metadata based on the achievements in multimodal learning. Then in Chapter 3 we will describe the data, models, metrics, and other experiment settings. Later in Chapter 4 we will describe experiments in respective sections in chronological order. We will summarize our work with conclusions in the last chapter.

Chapter 2

Related Work

We started referring to the related works in the previous chapter when we provided the historical background and conceptualized our project. In this chapter, we proceed to explain the context and describe achievements in Deep Learning research areas related to or adjacent to our goals of combining sparse meta-information with high-dimensional bitmap data. We will elaborate a bit more about Deep Learning applications in biomedical community, cover multi-modal systems capable of receiving different types of input and describe data fusion strategies. We will mention the Manifold Learning approach that transforms high-dimensional input into an encoded representation of lower dimensionality, and we will also mention the meta-learning principle, which arguably overlaps with our field of research. Then we will dwell on the Dynamic Neural Net concept, which we favored before the beginning of the project but later dismissed considering compound issues with implementation. In the section dedicated to Squeeze and Excitation Blocks, we will provide more technical details since these units form a ground for our models and deductions. The decision to include the last section with multi-task learning appeared after conducting experiments with different segmentation outputs and observing the phenomenon of drastic improvement for one of the tasks controlled by metadata in the joint training.

2.1 Biomedical applications

At the end of the 1990s, the shift from a rule-based approach to supervised machine learning techniques happened in biomedical image processing. Computer algorithms became adopted and commercialized for analysis and segmenting X-Ray images, ultrasound snapshots, and histopathology data. However, the broad use of handcrafted features persisted until the development of efficient training techniques for Neural Network which happened in the mid of 2010s. From that time, the burst of scientific publications on using CNN for biomedical image processing signified the fact that Deep Learning techniques gathered momentum and permeated the field of medical image analysis. An overview of Deep Learning algorithms, contributions, and notable milestones are provided in the publication of Litjens et al (Litjens et al., 2017).

In histopathology images, the observations of internal cell organelles and nuclei are crucial for phenotyping individual cells and for the analysis of drug propagation and cell response to them. The accurate visualization of nuclei and mitochondria is possible with sophisticated methods of sample preparation involving the use of staining substances that penetrate the target organelles adhering to DNA and highlight organelles under the microscope induced by a specific wavelength. This type of image modality is called fluorescent; it's effective but has drawbacks resulting in

higher cost of processing, time consumption, and quite low ability to observe dynamics of cell internals due to limited timeframe for sample readiness. The staining substance may quickly dissolve beyond the required structure and is usually toxic to the cell. Brightfield microscopy images that record natural light transmission properties do not contain particular information about the internal cell structure and are considered complementary to the fluorescent data in the analysis. However, Deep Learning enabled the utilization of brightfield modality as an independent source of information. It is possible to train NN with target images produced from fluorescent data and obtain powerful instrumentation to process cheap brightfield data (Fishman et al., 2019).

2.2 Multi-modal systems

Digital images often are supplied with metadata in the textual and tabular format provided by acquisition devices. In the field of DL a combination of inputs of different formats is referred to as Multimodal Deep Learning (Ngiam et al., 2011). The models taking into account diverse sources usually perform better than separate processing units.

The process of combining data from different modalities has been termed as data fusion. SC. Huang, A. Pareek et al. in the comprehensive review (Huang et al., 2020) explain the concept of data fusion and provide examples of applications in medical imaging. Also, they describe the varieties of this technique. Data fusion strategies fall into three types: early, late, and joint fusion, although the boundaries between them may appear fuzzy.

Early fusion implies data stacking before entering the model and using this combined data in a single input. The late fusion suggests using separate data flows, for example, the image-only model and the text-only model producing independent predictions to be ranked by a final aggregation module at the decision stage. We can't consider late fusion in our experiments because metadata is not supposed to produce independent predictions. In the joint fusion, semi-processed dataflows from separate inputs interlace inside the main model in a fully connected layer. We would like to note in advance that the way we incorporate metadata in our models falls into the joint fusion category. We will elaborate later in this chapter on the reasons for considering so.

From the reports dealing with supplementary tabular data resembling metadata, we can note the work of Kawahara et al. (Kawahara et al., 2019), in which supplementary data consisted of patient's physiogical and health parameters. The team constructed a multi-task deep convolutional neural network for skin lesion classification using "multi-modal multi-task loss function that considers multiple combinations of the input modalities". Under close inspection this approach can be classified as joint fusion strategy.

We should also pay tribute to the efforts of N. Gessert et al. (Gessert et al., 2020) who incorporated patient meta-data into other cancer-detecting NN. The model combines the ensemble and meta-learning strategy, it can be classified as joint fusion as well. The pre-trained sublearners keep fixed weights during the meta-module learning phase.

2.3 Manifold Learning

Among early fusion implementations, common are systems where image features undergo extraction on the preliminary phase, usually by CNN. They turn into the same format as supplementary data. After concatenation, the combined data goes into the primary model, which is not necessarily a NN. There are many implementations with a similar pattern called Manifold Learning, circulating in the field of medical imaging (Belkin and Niyogi, 2003; Tenenbaum, Silva, and Langford, 2000).

This format found applications in processing 3D MRT medical images (Zhu et al., 2018; Brosch and Tam, 2013; Park, 2012). The output produced by MRT devices has a high spatial resolution. The resulting 3D images are represented by high dimensional vectors when ingested directly by NN models (but with significantly lower dimensions than initial images). Manifold Learning allows compressing 3D data producing tabular-like data in a descriptive format instead of large arrays of pixels and voxels (Gerber et al., 2010; Gray et al., 2011; Tao and Matuszewski, 2013).

The manifold format becomes suitable for incorporating metadata. Few reports found this approach feasible (Wolz et al., 2011; Wolz et al., 2012; Aljabar et al., 2010). The systems using this format differ in architecture from image-specific CNN. However, Manifold Learning is not always necessary when working with 2D data. So far, the planar image format has a broad use in microscopy, and state of the art CNN's perform well with images. We will explore the way of using tabular metadata and pixel input data simultaneously without significant disruption of CNN architecture, using the merits in the related field of applications.

2.4 Meta-learning

Meta-learning has a general meaning of learning on a higher level, sometimes the ability to learn how to learn. Concerning ML, meta-learning implies a combination of independent learning techniques, algorithms, and effective selection of the most appropriate one based on meta-knowledge in order to improve the overall system performance.

A meta-learning system consists of a learning subsystem adapting with experience. The learning subsystem is often referred to as the base-learner. The system gains this experience from meta-classifier based on previous episodes and high-level information extracted from training data (Lemke, Budka, and Gabrys, 2015; Chan and Stolfo, 1993). In our case, we supply such meta-information externally.

The ensembles mentioned earlier may serve as examples of a meta-learning system when the selection of results from the collection of base learners is automated through the entire process of learning.

2.5 Dynamic Neural Nets

Conventional NNs have a fixed set of parameters after completion of the learning process. The same applies to ensembles having a set of predefined submodels. In contrast, Feihu Zhang and Benjamin W. Wah in their publication (Zhang and Wah, 2017) assert that meta-learning NN (MLNN) should dynamically approximate in selection to the most suitable submodel. One of the central powers of a meta-learning system is the ability to operate in new and unseen scenarios. They implemented such a system with utilization of dynamic weights. The concept of dynamic NN is not new, and there have been applications of them. Every model with parameters

changing in the inference time can be considered dynamic. So the range of concepts and implementation may be broad, but under a closer inspection, we were not able to find further development of this tempting idea in the literature and a possible implementation casts some doubts regarding complexity and feasibility.

Arguments in favor of dynamic parameters: greater flexibility, possibility to alter convolution layer parameters, lower dependence on training set balance, expected capability to interpolate between modalities and extrapolate beyond them.

Arguments against dynamic parameters: complexity in implementation (inflating with the number of layers applied), difficulties in tractability, susceptibility to methodological pitfalls. Since the parameters are altered during the training and inference time, there are engineering challenges, because existing frameworks allow processing the input data in batches, and NN parameters cannot be altered within a batch. So either we have to adhere to batches of size 1 or populate batches with similar images having same meta-parameters. Both approaches decrease effectiveness and increase computational cost. The proposed system would wildly digress from conventional CNN topologies, impairing direct comparison, debugging and application of proven strategies.

2.6 Squeeze and Excitation Blocks

In general, the data fusion technique is another aspect of model generalization occurring at the expense of feature space expansion. The interpolation or extrapolation between domains is unlikely. This holds true for traditional CNN models but may be different for Transformer models that are genuinely multi-modal from the inception and capable of combining data on various levels of abstraction (Vaswani et al., 2017). Transformers may possess dynamic properties (yet the vulnerability to a training set imbalance still may persist). Transformers are the nascent star in the NN constellation, but their study should be a topic for different research.

Considering data fusion, we already spoke against the late fusion strategy as unsuitable. For early fusion, two inputs must have the same modality. Earlier, we mentioned that bitmap data could be preprocessed by CNN for semantic feature extraction or transformed into another representation by Manifold Learning. But such image preprocessing is not practical for pixel-level semantic segmentation tasks, where CNNs remain the proven inventory. Therefore in the data fusion paradigm, only joint fusion remains a suitable strategy. The weak point of such strategy concerning CNN - it requires the presence of fully connected layers - the only place where the data fusion is possible. Classic CNN's for semantic segmentation are Fully Convolution Networks having no fully connected layers.

However, the modifications of CNN architecture are possible, and we found a suitable approach that we can term as Channel Attention Mechanism. Many researchers focus on spatial dependencies proposing beneficial alterations to a network topology with region attention mechanisms to improve CNN performance. Meanwhile, interchannel dependencies also may play an essential role in CNN clockworks while being less discussed and overshadowed by papers dedicated to spatial attention. Forcing the network to pay closer attention to interchannel dependencies at the right stage of processing is similar to the spatial attention effect. It is also capable of providing a noticeable boost to performance.

That was the idea behind the Squeeze-and-Excitation (SE) blocks proposed by J. Hu et al. (Hu, Shen, and Sun, 2017). SE units are compatible with virtually any CNN architecture, giving the improvement for the range of tasks with a relatively

small computational burden. The novel approach allowed authors to win ILSVRC 2017 classification task with top-5 error of 2.251%, exceeding the previous winning 2016 record by a relative improvement of ≈25%.

SE blocks provide access to a full range of channel activations on the layer, reinforcing the useful ones and suppressing the irrelevant ones at the right time and on the appropriate image processing stage. At the beginning of the block, each channel is squeezed to one pixel by an average pooling layer, thus serving as an activation descriptor for the given channel. From feature maps $U \in \mathbb{R}^{H \times W \times C}$ we obtain a set $z \in \mathbb{R}^C$ through "squeezing" $F_{sq}(u_c)$. These descriptors are fed into a small network consisting of three layers of neurons: input layer, hidden layer, and output layer - a sort of a multi-layer perceptron with a ReLU function δ and gated at the output by a sigmoid activation σ. The hidden layer is reduced by half or more, as configured by reduction ratio r, so the weights have dimensions $W_1 \in \mathbb{R}^{\frac{C}{r} \times C}$ and $W_2 \in \mathbb{R}^{C \times \frac{C}{r}}$ respectively.

$$s = F_{ex}(z, W) = \sigma(W_2 \delta(W_1 z)) \tag{2.1}$$

After passing this block, the channel descriptors gain new values based on block parameters that are learnable through backpropagation. Then modified descriptors are applied back to the multi-channel features by multiplication, eventually recalibrating them - the whole process termed by authors as *excitation*.

$$\tilde{x} = F_{scale}(u_c, s_c) = u_c s_c \tag{2.2}$$

where $\tilde{X} = [\tilde{x_1}, \tilde{x_2}, ...\tilde{x_C}]$ and $F_{scale}(u_c, s_c)$ is a channel-wise multiplication between the scalar s_c and the feature map $u \in \mathbb{R}^{H \times W}$.

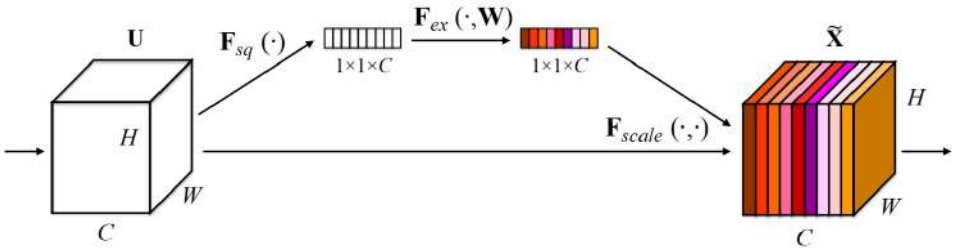

FIGURE 2.1: A Squeeze-and-Excitation block (Hu, Shen, and Sun, 2017).

The idea of SE blocks is further developed by Abhijit Guha Roy et al. (Roy, Navab, and Wachinger, 2018), who combined them with a spatial attention mechanism in a similar manner. However, the channel attention mechanism still contributed the most to the task scores improvement in their experiments. SE blocks become amenable for accepting encoded meta-information either by completely replacing channel descriptors or by concatenation with them. In case of replacement, we would witness the metadata-driven control of channel attention mechanism, so the system can suppress or bring forward the required channels depending on meta-labels. We will often refer to such blocks as Channel Attention units since they do not have squeezing operations any longer. In the case of concatenation, we would have a combination of this effect and channel recalibration. We will use both ways of incorporating metadata to compare the efficiency.

2.7 Multi-task learning

In the experiments on the dataset with different segmentation masks, we used meta-information to switch between tasks that otherwise would be incompatible in a single output. We observed an interesting effect of significant score improvement for one subset in the joint training compared to the individual expert model's performance in the condition of scarce training data for this particular subset. We realized that we dealt with multi-task learning finding the most likely explanation for the achieved results in publications dedicated to this type of DL. These findings also urged us to extend experiments and compare the performance of our model with classic multi-task and multi-head models.

Multi-task learning is a well-known technique having a multitude of implementations and applications in different areas. It naturally and almost simultaneously emerged from single task solutions offering several outputs designed for specific needs. In all such relevant literature, the effect of improving individual tasks from joint learning is noted, and sometimes it is specifically exploited. Rich Caruana, in his seminal paper (Caruana, 1997), provided an explanation of how such improvement occurs.

Sebastian Ruder, in his overview (Ruder, 2017), affirms that multi-task learning can appear and come in different guises. When the researcher optimizes more than one loss function, he certainly deals with multi-task learning. But that can be a disputable point when we try to apply it to our case because in metadata-driven model we have a single loss function; however, we can claim that it is optimized differently during the task change. So we are inclined to use a more general definition cited from the publication of Yu Zhang and Qiang Yang in a National Science Review (Zhang and Yang, 2017).

"Given m learning tasks $\{\mathcal{T}_i\}_{i=1}^{m}$ where all the tasks or a subset of them are related but not identical, multi-task learning aims to help improve the learning of a model for $\mathcal{T}_i$ by using the knowledge contained in the m tasks".

Such definition is even applicable to our metadata-driven model with a single output head. This comprehensive publication also covers the current state, the taxonomy of solutions, the progress, and possible ways in the development of multi-task learning approaches. Based on the classification from the paper, we can refer to our case as multi-task supervised learning.

Chapter 3

Experiment Settings

The conditions and environment for conducting experiments are indispensable and essential parts of the project. Therefore, this chapter describes the source data and ways of processing it along with the training of NN. The methods of metadata encoding are given in the respective section. The software framework for DL, backbone models overviews with peculiar hyperparameter settings follow next. Finally, we introduce the metrics for evaluating all results in our experiments, which appeared universal and straightforward for all cases.

3.1 Datasets

The data for the project was provided by *PerkinElmer, Inc*. The Primary dataset ("7 lines" dataset) consists of high-resolution microscopy images of cells from various human tissues and organs and respective ground truth masks of size 1080×1080 px. The total size of this dataset is 3024 samples split into training, validation, and test parts as 2016:504:504. The main objective in processing cell images with the aid of Deep Learning is segmenting nuclei for further analysis: count, size, shape, detection of dead cells, movement tracking, and many other observations that help investigate the cell internals and the response to various treatments. For some tasks, we also included a small dataset with anomalies occurring on the images that may help in refining the existing datasets, increasing the quality by removing the spoiled samples, or teaching the NN to ignore visual contaminations. The Primary dataset has seven subsets representing cells from various sources, having somewhat different appearances and morphology. The distribution of images across cell lines are given in Table A.1. We also used an additional dataset (Table A.2) from AstraZeneca (AstraZeneca, 2021) in some experiments, which we preprocessed for compatibility with "7 lines". Sometimes, the Primary dataset parts (specific cell lines) were used for intermediate experiments to form a smaller volume of data or to combine it with external data.

The methods for the preparation of biological samples may vary in complexity and ways how they impact cells, resulting in different appearances of digitized images. One requires the application of dye that penetrates cell nuclei and highlights them with high contrast. These spectacular fluorescent images often serve as a basis for constructing ground truth masks with the help of software and human supervision. However, this method is expensive, time-consuming, and invasive, literally killing the cells, making the observation of processes and movements developing over time in live cells impossible.

The more convenient way of studying cells is the brightfield method, where the samples are not exposed to invasive substances and are observed as is, highlighted by a light source. These images have low contrast and are harder for downstream

processing. But harnessed by the power of Deep Learning, microbiologists can automate the workflow and obtain the segmented images of good quality, provided the neural models were trained and tuned properly. The examples of brightfield and fluorescent modality images from Primary dataset, acquired by Opera Phenix microscope, are given on Figure 3.1 along with ground truth masks.

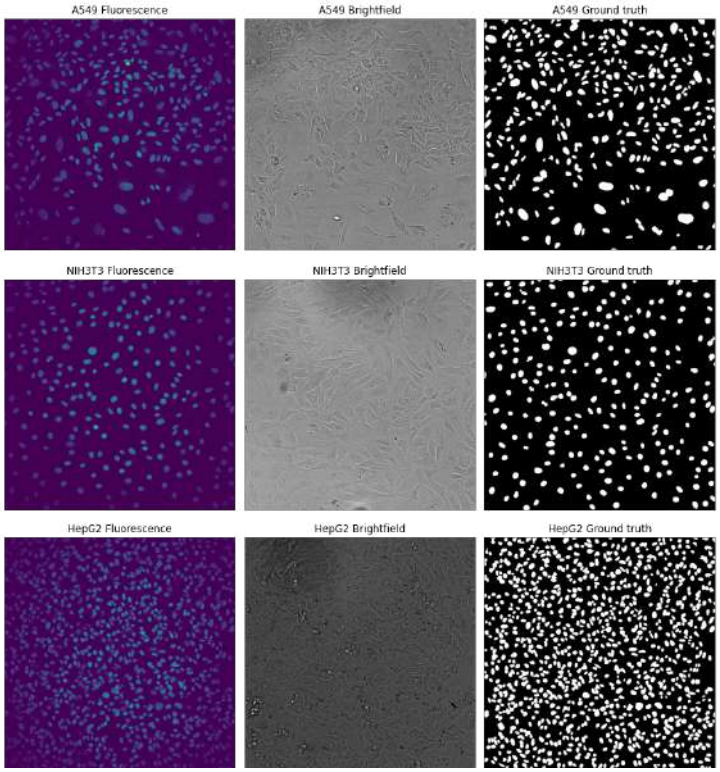

FIGURE 3.1: Examples of microscopy data taken from three cell lines (Source: PerkinElmer Primary dataset).

For our experiments we used only brighfield modality as more practical and challenging. The fluorescent sources we considered too easy for semantic segmentation task. The ground truth masks consist only from two classes: white foreground with segmented nuclei and black background.

Another dataset is named an Exhaustive dataset with more diverse meta-information, from where most notable is magnification. This dataset consists of 3888 images, split into the train/validation/test parts as 2577 : 642 : 669. More details will be provided in the respective section with the experiment involving this dataset.

We resorted to external and synthetic datasets in a few experiments to emphasize the researched phenomena. These datasets will be better described in dedicated sections.

The original size 1080×1080 is inconvenient for NN models leading to GPU memory outage, so the training images undergo random cropping of size 512×512 px synchronously with masks. Validation images were cropped only in the central part to preserve consistency across the experiments. During the inference, we applied tiling 512×512 px to the test data images of original size with overlapping tiles. These patches pass through the model, and output masks were stitched back in their original order discarding the overlapping margins. Disassembling and reassembling the test data is a good strategy for utilizing GPU resources without impacting the performance since the cellular patterns on the images are homogeneous, allowing the

described patchwork. Also, this strategy serves as a good augmentation tool, providing random patches during the training time. Apart from such augmentation, we didn't use any other. Cell images are relatively isotropic, so flipping and rotating augmentations would not bring positional diversity but rather expand the dataset size. However, random cropping was already considered sufficient for the diversification of images. Regarding the brightness and contrast changes, the dataset also contains a sufficient range of intensities. Color augmentation are not applicable to single-channel input images. Nevertheless, the most significant reason for not using other augmentation types was that they were not used for methods achieving SOTA results. In our work, we focused exclusively on NN topological search. Introducing new image augmentations into the pipeline could unnecessarily expand the possibilities in the common search space of solutions.

3.2 Metadata encoding

We encoded metadata with a vector of length n, where n amounts to the number of domains (subsets), which constitute the dataset. The components of this vector are populated with floating-point numbers having either of two values: 0.0 or 1.0. This vector is supplied to NN in a separate input. Further, we will refer to it as a meta-input. Meta-input data undergoes the same preprocessing as main input data. First, the formation of batches occurs where each vector in the meta-input batch corresponds to the source image in the main input batch. Then the casting to NN digestible format is scheduled (tensors in PyThorch framework). Finally, the metadata is delivered to GPU for synchronous feed with main data. The number of discrete values in metadata defines the number of components. For example, in the Primary dataset with seven lines, the metadata comprises 7-"bit" vectors. Each component is responsible for the single domain, taking the value 1 when the source image belongs to the particular cell line and 0 when it belongs to other cell lines. Thus, only one component is "active" (taking 1) in the supplementary metadata vector for the Primary dataset. Values are converted to to floating-point format in the pipeline. The example of dictionary with metadata vectors for each meta-label of the seven lines is below:

```
HeLa    [1, 0, 0, 0, 0, 0, 0]
MDCK    [0, 1, 0, 0, 0, 0, 0]
A549    [0, 0, 1, 0, 0, 0, 0]
HT1080  [0, 0, 0, 1, 0, 0, 0]
HepG2   [0, 0, 0, 0, 1, 0, 0]
MCF7    [0, 0, 0, 0, 0, 1, 0]
NIH3T3  [0, 0, 0, 0, 0, 0, 1]
```

Such encoding resembles a binary code or one-hot format, but it is not so because it allows the redundancy of data. For example, for experiments with two subsets, we used a two-component vector, where the value `[1, 0]` denoted one source, and `[0, 1]` another source, while the total number or combination is four:
`[0,0], [0,1], [1,0], [1,1].`
One-hot encoding is not mandatory because this vector is an input (or a part of the input) for Channel Attention block with a small neural network, having no requirements for avoiding multicollinearity. The other reason for not using one-hot encoding is better interpretability. It was convenient, for debugging especially, to hold a specific position in the metadata vector responsible for a particular domain.

This encoding acted as expected, as our experiments showed. For the testing, we often applied different modes of meta-label supply. Apart from “normal” mode with native meta-labels we used “shuffled” mode, inferring the test images with foreign meta-labels, “zeros” and “ones” modes when the meta-information vector consisted of all zeros or ones. Such cases were included for curiosity, and they confirmed the full (in case of "shuffled" mode) and partial (in case of "ones" mode) engagement of the channels and pathways from other domains; for "zeros" mode the biases of SE/CA blocks are exposed. In contrast, we observed the full engagement of correct pathways with a label native to the subset. This behaviour has been permanent for all experiments throughout the research.

Such encoding also allows the inclusion of continuous variables since we created the metadata-driven pipeline with floating-point format, reserving the possible development in future. However, in our experiments, we didn’t use continuous meta-parameters.

3.3 Models and Hardware

Two Convolutional Neural Net models were the main tool for semantic segmentation in the current project: Unet3 and PPunet. Both are the derivatives from the classic U-Net (Ronneberger, Fischer, and Brox, 2015). These models have five stages of downsampling and upsampling, triple convolution layers instead of double ones in the classic implementation, and a constant number of filters in each layer (64) regardless of the level. For the Unet3 this results in fewer parameters than the original U-Net without deterioration in segmentation performance. PPUnet contains Pyramid Pooling Layers (PPL) with five interpolation units operating on various scales. PPL gained recognition in the Computer Vision community by enhancing the accuracy and making the model more robust to inconsistent object sizes (He et al., 2015). PPUnet also features skip connections between encoder levels. This model (in "naked" form without SE blocks) is currently accepted as a state-of-the-art model in the Tartu research group, giving the highest scores on the Primary dataset.

Besides the declared benefits of Pyramid Pooling blocks, we pursued two other goals with PPUnet:

1. using the different software framework (PyTorch in our project, while the significant part of the code in Tartu group exercise Keras) and custom pipeline we could compare our results with SOTA records, which eventually appeared to be a fortunate decision, that allowed us to debug and increase the performance of new pipeline;

2. exceed, if possible, the highest results and set new records in semantic segmentation of Primary dataset.

By augmenting models with Squeeze and Excitation (SE) blocks, we obtained enhanced models which generally performed better than models without SE blocks (“naked” models), and also we obtained models capable of accepting encoded meta-information. Extra SE modules are lightweight and added only 3-7% to the initial number of parameters, e.g. “naked” Unet3 has 1.37 million parameters, and Unet3 equipped with SE blocks has 1.42 million (3% increase), source PPUnet has 2.10 million parameters, and PPUnet+SE has 2.25 million (7% increase). The schemes of both models and the positioning of SE blocks are illustrated on Figures 3.2 and 3.3.

In our experiments, we used several baseline models for comparison with meta-driven ones. For the master models trained on the whole dataset we took naked

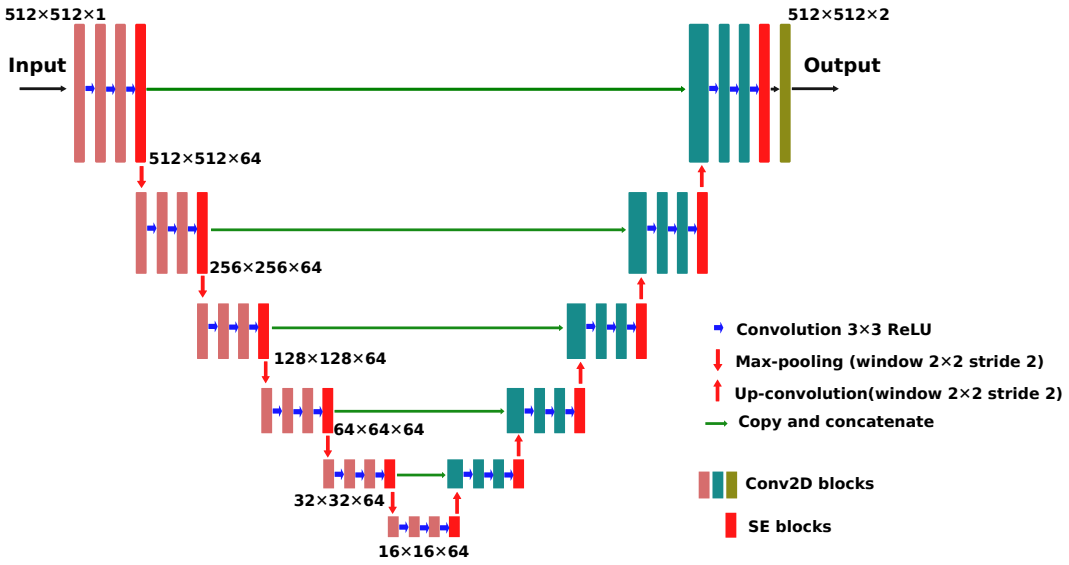

FIGURE 3.2: Unet3 schema with SE blocks.

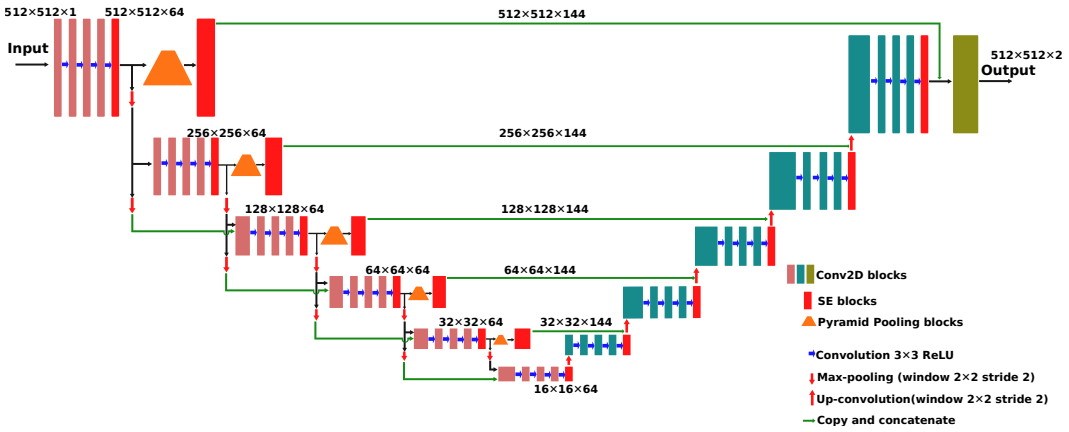

FIGURE 3.3: PPUnet schema with SE and Pyramid Pooling blocks.

configuration(without SE blocks) and configuration with SE blocks without meta-input. Models for investigating the effect of meta-informaion were equipped with legacy SE blocks taking exclusive meta-input (so technically, using "squeeze" in the naming of these blocks is no longer correct as no input channels accepted, so further we will refer to such units as Channel Attention (CA) blocks). Also, we used models with SE blocks and concatenated meta-input. We used SE blocks with concatenated dummy input (usually zeros) of the same size as meta-label vector in later experiments as an additional reference model. We trained individual models on a restricted subset for many experiments to obtain an "expert" model only in a specific domain. Such individual models featured conventional SE blocks.

At the beginning of the study, we used SGD optimizer with a starting learning rate of 0.01 and learning rate optimizer ReduceLROnPlateau, which reduced the learning rate by half when validation loss fails to improve for five consecutive epochs. The typical plot for learning rate with this optimizer is shown on Figure 3.4

After pilot experiments, the optimizer was replaced by Adam with a base learning rate of 0.0002 and an upper rate of 0.0008. The rate scheduler was CyclicLR which changes the rate in jigsaw manner with decaying amplitude, approaching to the basic learning rate (Figure 3.5). Adam optimizer itself decreases learning rate

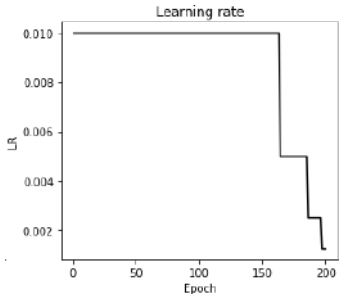

FIGURE 3.4: Learning rate with ReduceLROnPlateau scheduler.

adaptively to various parameter groups and uses it as an upper bound. So learning rate scheduler only limits the upper bound for Adam. There is no common opinion whether Adam requires a scheduler, but similar practice with a sinusoidal rate pattern is employed as SOTA method in the research group. We didn't conduct studies for the determining the benefit of the scheduler, but at least we assumed it to be harmless. The typical plot of the learning rate for Adam is given below in Figure 3.5. The frequency of changing the learning rate direction is set by a fixed number of cycles, it was adjusted for the entire training session, and the scheduler pattern scales on the number of epochs (accounting for training set size and batch size).

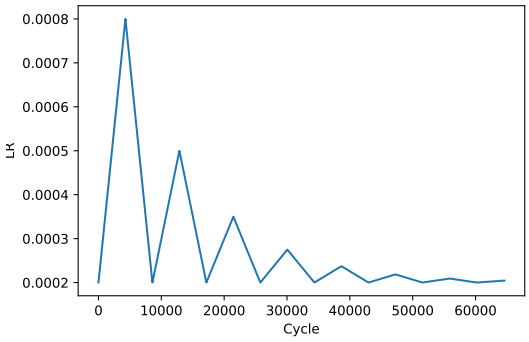

FIGURE 3.5: Learning rate with CyclicLR scheduler.

The usual number of epochs for all experiments is 100. In some experiments, we extended it to 200, and for quickly converging experiments with synthetic data in concluding ablation studies, we restricted it to 50. We saved the best model on the training range to use for downstream analyses. Usually, the majority of the models converged approximately after the 70th epochs, even with large datasets. If a further decrease in validation loss occurred, it was so minuscule that we considered it not worth investing more time than 200 epochs. That was why we used a fixed number of epochs instead of letting the model train with an early stopping setting. We performed a few runs on the Primary dataset with patience setting 20 (implying that training should quit when the loss does not decrease during twenty consecutive epochs). The model continued training more than 400 epochs but with negligible improvements, which were irrelevant compared to the best status of training session interrupted after 100 epochs.

The loss function used as criteria for minimization in the backpropagation algorithm is *CrossEntropyLoss*, having the general form as:

$$J = -\frac{1}{N}(\sum_{i=1}^{N} y_i \cdot log(\hat{y}_i)) \tag{3.1}$$

where y_i and $\hat{y}_i$ are target class and prediction probability respectively for each pixel. However, in DL framework the implementation for calculating this kind of loss might differ.

The DL framework for our project was PyTorch of version 1.6.0+cu92, CUDNN version: 7603. Computations were performed on GPU Tesla v100 16GB and 32 GB installed on High Performance Centre (HPC) servers in the Institute of Computer Science at University of Tartu. The amount of RAM in HPC allowed us to use in-memory datasets up to 20 GB.

3.4 Metrics

The pixel-wise and object-wise evaluation of accuracy and other metrics in semantic segmentation rely on a comparison with ground truth. Intersection-over-Union (IoU), also known as Jaccard Index, measures the overlap between the ground truth mask and model prediction.

$$IoU = \frac{target \cap prediction}{target \cup prediction} \tag{3.2}$$

Equally common is F1 score, also known as Sørensen–Dice Coefficient, which is a harmonic mean between precision and recall.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{2TP}{2TP + FP + FN} \tag{3.3}$$

where TP denote true positives - predicted foreground pixels which coincide with ground truth (GT) mask pixels, FP - false positives, predicted foreground pixels located on the wrong place relative to the GT, FN - "underpredicted" foreground pixels when predicted background pixels reside in place of GT foreground pixels.

Further we will use F1 score as a sole metrics for presenting results of our experiments because IoU and object-wise metrics were strongly correlated and consistent with each other. Since oftentimes F1 is relatively high and the difference between scores is minimal, we will use the **F1 error** metric to improve visual perception for plotting. F1 error is calculated as a value complementing F1 to 1:

$$F_{1\,err} = 1 - F_1 \tag{3.4}$$

Chapter 4

Experiments and Interpretations

The most voluminous chapter in this thesis is dedicated to experiments and their results. We described our activities in chronological order spreading over three months. We started from the first pilot experiments to test preliminary assumptions and to adjust the software kit. Then we proceeded to more extensive experiments with various datasets, each having a unique name for the convenience of referring and identification.

4.1 Pilot experiments

The first experiments aimed to investigate the parameter fusion effect described in the Related Work chapter as Dynamic Neural Nets 2.5. The 2-line dataset was formed, consisting of two cell lines (HeLa and MDCK). Each cell line was supplied with a random specific noise tensor of size 64×3×3 to be added to Unet3 middle layer parameters during the training and the inference time. This experiment showed that such an approach had no effect on the model's selective performance towards each cell line, serving as a sort of regularisation but regardless of the cell line. That is to say, the model adapted to such perturbation and inverting the labels (catering foreign label with images of particular cell line) didn't affect the inference results. Neither did supplying one constant particular label for both subsets nor adding zeros to parameters, which implies labels are useless at inference time. The pixel-wise scores for the test set were modest and much lower than SOTA scores in Tartu group on respective cell line subsets (as well as scores of our best model from further experiments). Our model scored 0.78 on HeLa subset while the SOTA results reached 0.9 and higher, the respective figures for MDCK are 0.73 and ≈0.83

Considering the unclear vision for developing the idea of dynamic neural networks in the frame of our research, and engineering challenges also mentioned in Dynamic Neural Nets section 2.5 we ceased conducting experiments and further planning in this direction and focused exclusively on SE blocks.

However, we also had a toy model accepting the meta-information in the form of an image (similar to QR-code used in commerce) into the dedicated channel along with the main microscopy image. This approach arguably failed: the model actually performed better when the black image was supplied at the inference time, so additional bitmap information only served as a detractor during the learning process.

We started exercises with SE blocks. The Unet3 model equipped with SE blocks on each level performed significantly better than "naked" Unet3 on the same dataset and with the same hyperparameter settings. In pilot experiments, we used a "two-bit" meta-input to denote the origin of the sample: `[1, 0]` for HeLa and `[0, 1]` for MDCK. Few implementations with topological variants of how to add meta-information to SE/CA blocks were tested, listed below and illustrated in Figure 4.1:

1. option when meta-input ("two-bit") was concatenated to 64-length input of SE block;

2. option when 20 additional duplicated meta-inputs were concatenated to the input of SE block to assess if the meta-input size makes a difference in concatenation with original 64-input;

3. option with additional flat layer in SE block to allow more complex non-linear patterns in channel selection (meta-input is concatenated to the conventional input of SE block in this configuration);

4. option when meta-information was concatenated not to the main input of SE block but to the middle layer where the number of neurons is reduced;

5. option with meta-information as an exclusive input to SE block without activated channels from the previous convolutional block. In this configuration, meta-input replaces the conventional SE input, but the middle layer remains as a legacy from SE topology;

6. option with exclusive meta-input and redundant middle layer removed. This configuration was tested chronologically later on other dataset.

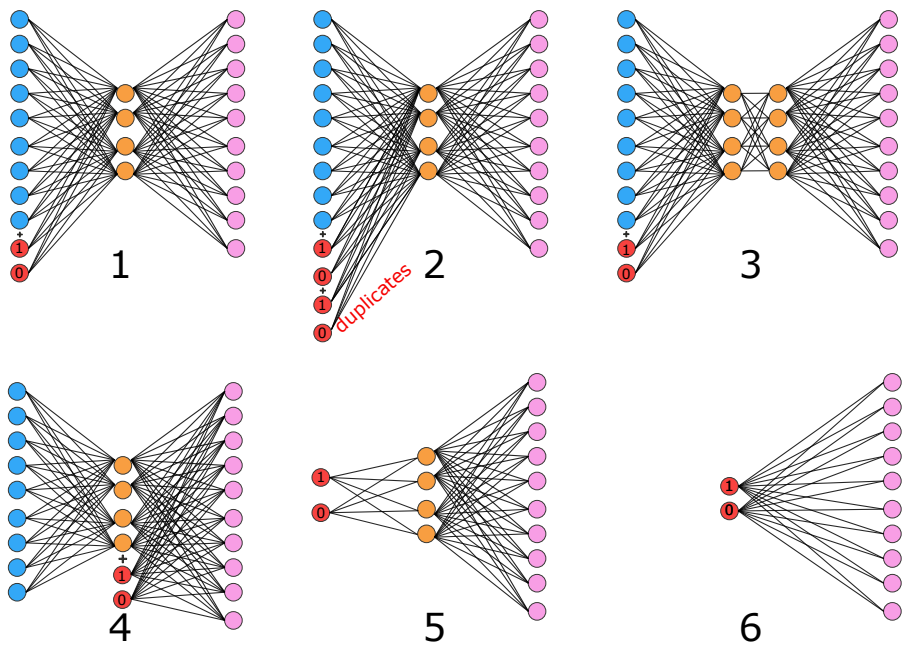

FIGURE 4.1: Variants of introducing meta-information to Channel Attention blocks (the downscaled version of blocks which originally have 64-length input). Red circles - meta-information, blue circles - conventional SE input of channel descriptors, orange circles - hidden layer(s), pink circles - output.

More experiments to determine the appropriate reduction ratio affecting the size of the middle layer were performed on another dataset and will be mentioned later. However, they did not change our approach, and in all models with combined SE+meta blocks, we used reduction ratio 2. We should note now that for the majority of models that were controlled exclusively by meta-information (without squeezing

the channels from the previous layer) we still used three flat layers inherited from SE block topology (variant 5 in Figure 4.1), though the middle layer between the input and output layer in such configuration becomes meaningless, carrying no useful functionality. The middle layer was discarded in final stages of the project, but until then, we kept it as legacy and to maintain the similar structure and number of parameters with model having a combined input SE+meta. For this concatenated variant the configuration 1 from Figure 4.1 was used throughout the project.

All options performed well and comparable with the baseline model with SE blocks (Table 4.1). It's worth noting that these models are responsive to labels swap - inverse labels cause the performance drop, and supplying only one label favors only that subset which is associated with this particular label (Table 4.2). We can assume that pathways from both domains have a lot of features in common - this is reflected during the inverse mode of label supply - the score for the domain with foreign meta-label doesn't drop significantly.

Models / Subset metrics	Master model naked (reference1)	Master model with SE (reference2)	Master model SE+meta with extra middle layer	Master model SE with duplicated meta-input	Master model with combined input SE+meta	Master model with exclusive meta-input	Expert model trained on HeLa	Expert model trained on MDCK
Combined F1	0.74	0.796	0.797	0.77	0.795	0.806	0.652	0.661
Hela F1	0.779	0.823	0.822	0.8	0.82	0.833	0.771	0.634
MDCK F1	0.675	0.753	0.757	0.723	0.755	0.761	0.468	0.699

TABLE 4.1: F1 scores for models from pilot experiments

Models & mode / Subset metrics	SE+meta normal	SE+meta inverse	Meta only normal	Meta only inverse
Combined F1	0.795	0.786	0.806	0.758
Hela F1	0.820	0.805	0.833	0.786
MDCK F1	0.755	0.758	0.761	0.720

TABLE 4.2: Example of F1 score response to test modes in two models with meta-information from pilot experiments. Normal mode means test images were paired with native meta-labels, in the inverse mode they went with foreign meta-labels (images from HeLa domain were paired with MDCK meta-labels and vice versa)

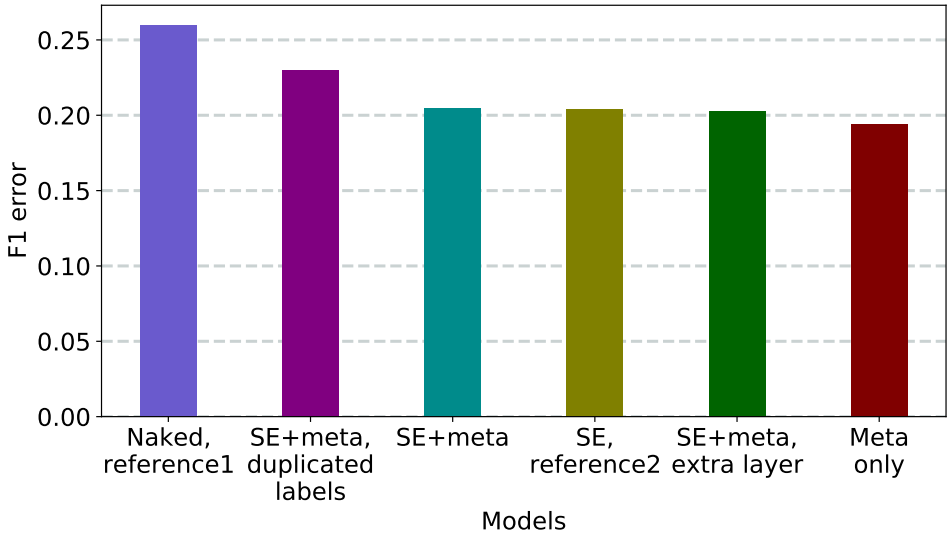

FIGURE 4.2: F1 error rate on the entire set for models in pilot experiments.

Another observation - a master model trained on both sets performs better on these sets than an individual model trained on a single subset - even for the naked model. This may be counterintuitive to our assumption that the model should be an expert in a narrow domain but lose some accuracy with generalization on multiple domains. But it turns out that both domains serve as good augmentation techniques and contribute to each other's scores in the master model. The pilot experiments served as a warm-up for more profound experiments and helped us to establish and tune the pipeline for the entire project. The ranking of the models from pilot experiments are shown in Figure 4.2.

4.2 Experiments with Primary dataset

We have described the Primary dataset in the preceding chapter with experiment settings details and the distribution across cell lines is provided in the Appendix A (Table A.1). We started experiments with Unet3 and PPUnet models using 200 epochs and SGD optimizer. Still, we could not reach SOTA records on this dataset that comprised the F1 score of 0.8523. However, the models with SE blocks and incorporated metadata showed better results than our baseline models, and metadata-driven models responded to label swap in test modes. After the scrutiny, we upgraded the optimizer to Adam, employed a CyclicLR scheduler, and applied post-processing of predicted masks: removing small stray holes and objects with an area below a certain threshold. With such optimization, the model started to show better results, and some of them exceeded the SOTA score for the Primary dataset, though the margin was relatively small.

Below in the Table 4.3 are the results after debugging the process, with scores for the entire dataset and specific subsets (rounded to the precision of the third digit after the dot). Also, in the last column, there are results from the models trained only on subset samples. These individual models had PPUnet topology with SE blocks. (We should point out that Unet3 with concatenated metadata was discarded from these results due to the discovered mistake during the training process)

The Figure 4.3 ranks the models by F1 error rate on the entire dataset.

Domain \ Model	PPUnet naked	PPUnet with SE	PPUnet meta	PPUnet meta + SE	Unet3 naked	Unet3 with SE	Unet3 meta	Individual models
Combined	0.850	0.854	0.855	0.855	0.849	0.854	0.854	-
HeLa	0.890	0.899	0.899	0.899	0.897	0.898	0.901	0.901
MDCK	0.858	0.864	0.864	0.867	0.856	0.865	0.866	0.830
A549	0.847	0.852	0.853	0.852	0.846	0.851	0.853	0.841
HT1080	0.864	0.867	0.868	0.869	0.862	0.868	0.866	0.857
HepG2	0.803	0.806	0.806	0.805	0.800	0.805	0.804	0.806
MCF7	0.836	0.839	0.840	0.839	0.834	0.839	0.839	0.825
NIH3T3	0.883	0.890	0.891	0.891	0.886	0.887	0.891	0.891

TABLE 4.3: Resulting scores of master models and individual models(last column) from experiments on Primary dataset

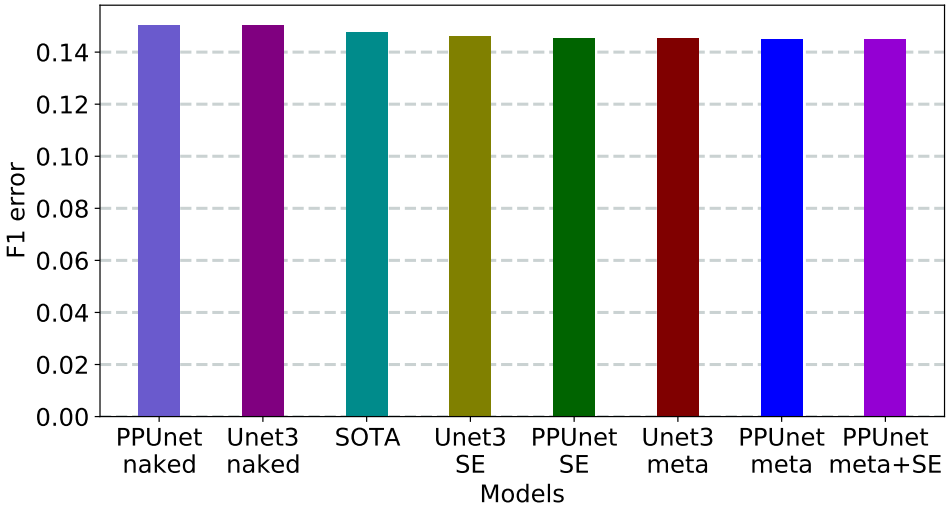

FIGURE 4.3: F1 error rate on the entire set for models in Primary dataset experiments and comparison with SOTA result.

Reference model results approached SOTA records, and configuration with SE and SE+meta exceeded them by a small margin. The difference between the naked model and SE model is noticeable, and the difference between SE and meta is subtle but still statistically significant according to paired t-test using individual F1 scores with 503 degrees of freedom and H1 hypothesis that the first set scores are greater (Table 4.4).

Test	p-value
SE results vs naked	3.5E-56
Meta results vs naked	4.3E-82
Meta results vs SE	0.00057

TABLE 4.4: Paired t-test on results from Primary dataset experiments

The advantage of metadata-driven models over reference SE model, however, is not quite convincing. We can attribute this to the fact that domain-specific models are not superior to the metadata-driven models. Only HeLa and NIH3T3 models can be considered the better "experts" in the respective domains. For the rest of the cell lines, the master model without meta control (SE model) demonstrates higher scores on the specific subsets. In general, we may speculate that meta-information cannot convey the expertise from the domain-specific configuration to master model. The

master model becomes more proficient without meta-information by exchanging the knowledge between domains in the superset.

4.3 Experiments with Exhaustive dataset

We prepared a new dataset of microscopy images from the large annotated pool named Exhaustive dataset for the next set of experiments. The preprocessing consisted of selecting the records with better masks and contrasty sources, converting the source images from TIFF to bitmap format with values downscaled to the range [0, 255], a total of 7776 images which is 2.5 times bigger than the Primary dataset volume, extending the training time up to 40-50 hours. To expedite the process we downsampled this set to 3888 images, splitting it into the train/validation/test sets as 2577 : 642 : 669. Three sets from meta-annotations were selected representing microscope magnifications 10x, 20x, and 40x. The distributions of samples within this preprocessed dataset across magnification values and cell lines are provided in the Tables A.3 , A.4 of Appendix A. The encoding consisted of three "bits" for magnification.

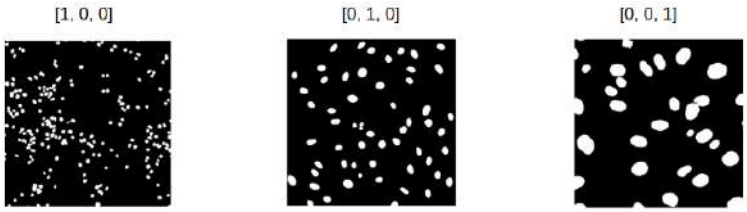

FIGURE 4.4: Examples of masks and metadata encoding in Exhaustive dataset experiments.

The backbone model topology was PPUnet, and the duration of experiments was 200 epochs each. From this point we started to use another reference model: dummy+SE, which accepted meaningless information (zeros for all samples) instead of encoded metadata. In later experiments, when the length of metadata encoded vector increased, it might be a potential factor of inequality between bare SE models and models with concatenated metadata because it implicitly affects the reduction ratio for a meaningful part of SE input.

The first experiments didn't show any advantage of metadata-driven models over the reference SE models. We additionally adjusted the pipeline fixing all possible seeds for PyTorch non-deterministic components. Eventually, the results didn't improve significantly, and meta-information didn't give an overall performance boost in the current experiment despite showing signs of being absorbed in label swap tests. The aggregated results for the experiments with encoded magnifications are below in Table 4.5. They also show that models individually trained on magnification subsets are surpassed by master models in some cases (for 40x subset, highlighted in maroon color). Also, the margin is small, which suggests that even this advantage may be attributed to randomness. We can also speculate that these subsets still share some features and may benefit from a diverse set of training images despite being different in scale and representation.

The intersection of common features is small especially for the domains with a magnification of 10x versus 40x. Such assumption is based on the drastic drop in scores for this subset when labels swap in the respective model. Below in Table 4.6

Subset \ Model	Model 10x	Model 20x	Model 40x	Naked	SE	Meta	Meta +SE	Dummy +SE
Combined				0.745	0.784	0.782	0.784	0.785
10x	0.754			0.705	0.744	0.746	0.737	0.746
20x		0.812		0.750	0.800	0.796	0.802	0.799
40x			0.807	0.780	0.807	0.803	0.809	0.808

TABLE 4.5: Results on subsets from the reference models and models encoded with magnification data in Exhaustive dataset experiments. The cases when master models exceed the scores of individual models are highlighted in maroon color

is an example for the model with metadata input concatenated with SE input. "Normal" mode implies provision of native meta-labels with images from the respective subset. In the "Zeros" mode only zeros are supplied throughout the full test (in such mode the outputs of CA units expose internal biases), for "Ones" the meta-input is populated with 1.0, in "Shuffle" mode (which is actually a shifted mode) images from 10x domain were paired with "20x" meta-label, images from 20x domain were paired with "40x" meta-label, images from 40x domain came with "10x" label. Also, there are modes when the single label "10x", or "20x" or "40x" is supplied for the test set, showing the highest score on the native subset expectedly.

For the metadata-driven model without concatenation, the picture is similar. And for the model with dummy meta-input the deviation in different modes is also present but to a small extent (no more than 0.005 of absolute F1 score value). That means some residual weights and biases have been still learned from the training though they are meaningless.

Subset \ Mode	Normal	Shuffle	Zeros	Ones	10x	20x	40x
Combined	0.784	0.553	0.701	0.626	0.564	0.661	0.611
10x	0.737	0.603	0.631	0.615	**0.737**	0.603	0.130
20x	0.802	0.695	0.783	0.758	0.633	**0.802**	0.695
40x	0.809	0.155	0.66	0.42	0.155	0.502	**0.809**

TABLE 4.6: Results of various test modes for the model with magnification meta-information concatenated with SE input. "Normal" mode implies provision of native meta-labels with images from the respective subset. In the "Zeros" mode only zeros are supplied as meta-labels throughout the full test, for "Ones" the meta-inpu is populated with 1.0, in "Shuffle" mode images were paired with non-native labels. In modes "10x", "20x" and "40x" only one respective label is supplied for the whole test set.

The meta-information for the Exhaustive dataset contains other parameters, including the cell line name matching Primary dataset domains except for MDCK, so the cell line parameter had six values. We conducted a set of experiments with a double parameter grid, adding 3-component magnification data and 6-component cell line data together into a 9-component vector. The total number of combinations is 18, and that is the number of subsets on the cross-sections of magnification and cell line domains. Below we present only a summary (Table 4.7) and a plot (Figure 4.5) for all models (including previous from magnification domains experiments) in the normal testing mode on the entire dataset. We also have scores for all 18 subsets, but they don't pose much interest and were used only for recording the results. Such

details were considered redundant for the format of this thesis. The idea of training 18 reference models on each subset also was rejected as superfluous. Yet this experiment demonstrated some relevance of meta-information compared to the dummy model and compared to the experiments where only magnification was used. But the margin is small and may be subject to randomness. Retraining the models several times would have given us more confidence in results, but investing the time and efforts in exploring such marginal effects was considered inappropriate.

Model	Naked	SE	Meta 3bit	Meta 3bit +SE	Dummy 3bit +SE	Meta 9bit	Meta 9bit +SE	Dummy 9bit +SE
F1 score combined	0.745	0.784	0.782	0.784	0.785	0.778	0.795	0.785

TABLE 4.7: Performance summary of models with one-parameter and two-parameter meta-information in Exhaustive dataset experiments.

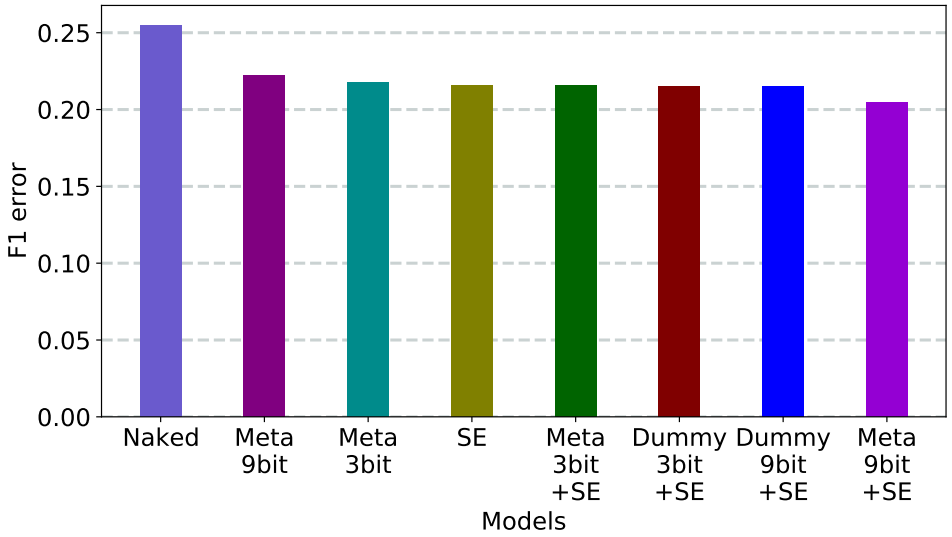

FIGURE 4.5: Performance summary of master models in Exhaustive dataset experiments.

The Exhaustive dataset served as a basis for one more lateral experiment investigating how the master model affects the performance on the specific subset when this subset is under-represented in the training data. First, the 40x subset was curtailed to 20% of the original size, overall comprising 10% of the entire training set. Then, in the second experiment, the 40x subset was decreased even more, down to 2.5% from the whole dataset. In this last experiment, the subset contained around 60 samples.

In both experiments, the master models (especially with SE/CA blocks) demonstrated a higher score on the 40x subset (Table 4.8), implying the enrichment of features compared to the individual model. The model with SE blocks was more accurate. However, the superiority of the models with meta-information is not convincing since the dummy model (fed with zeros instead of meta-labels) is not far from the meta-model.

Setup / Model	10% representation	2.5% representation
"Expert" model 40x	0.752	0.616
Naked master model	0.742	0.644
SE master model	0.757	0.722
Meta master model	0.758	0.735
SE+meta master model	0.763	0.732
SE + dummy master model	0.765	0.727

TABLE 4.8: Performance of models on under-represented 40x domain.

Though this study was less relevant in the general context of our project, the main takeaway from this experiment was that model performance for the under-represented subset could benefit from training with other subsets even if they have fewer features in common, as in the example with magnification domains.

4.4 Experiments with Extended dataset

The following experiment we designed with the inclusion of AstraZeneca (AZ) samples (628 training samples, as given in Table A.2 , nearly as an average number for each cell line in the Primary dataset). We appended a new subset to the initial Primary dataset and named it the Extended dataset.

The intention was that AZ dataset would provide more insights as perceptually it has greater domain distance from Primary dataset samples. So, first, the models were trained in a "binary" mode, when only 7LINES or AZ label was specified in the meta-label. Also, new metadata-driven models were trained with original labels for each cell line, and the AZ subset was denoted as the 8th line. PPUnet was a single topology choice for these experiments.

The summary Table 4.9 is presented below, where the "Combined" column denotes the F1 score for the whole dataset (and therefore weighed average score for all constituent subsets) and two other columns show scores for complementing subsets.

Subset / Model	Combined score	AZ score	7LINES score
AZ individual mode		0.852	
7LINES individual model			0.852
Naked master model	0.848	0.852	0.846
SE	0.850	0.851	0.849
"Binary" mode			
Meta	0.847	0.848	0.846
Meta+SE	0.851	0.854	0.850
Dummy+SE	0.851	0.855	0.849
"8 lines" mode			
Meta	0.850	0.850	0.850
Meta+SE	0.853	0.852	0.853
Dummy+SE	0.849	0.853	0.849

TABLE 4.9: Performance on Extended dataset.

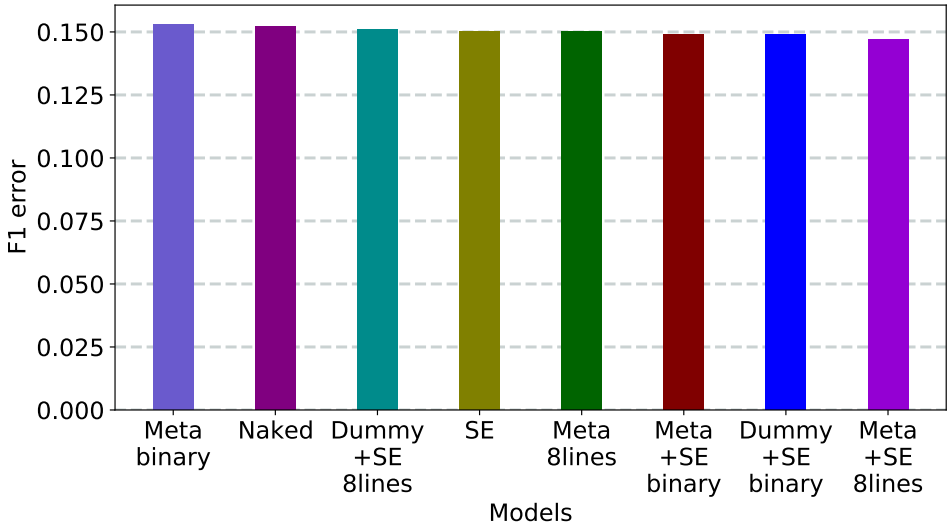

FIGURE 4.6: F1 error of models on Extended dataset experiments.

The resulting plot of F1 error on Figure 4.6 shows small unconvincing victory of the model with detailed meta-input concatenated to SE input.

4.5 Experiments with Heterogeneous dataset

In the next experiment, we decided to combine not only different datasets but different tasks as well. We added the external set with crevices in the concrete walls (Mendeley Data, 2019) to the two cell line sets (HeLa and HepG2), where the task was to segment these cracks. A couple of samples from this subset, which we marked with CRACK label, are shown below in Figure 4.7. We named this combined collection of samples a Heterogeneous dataset. The detailed data distribution within this dataset is given in Table A.5 of Appendix A. Meta-information vector had three components.

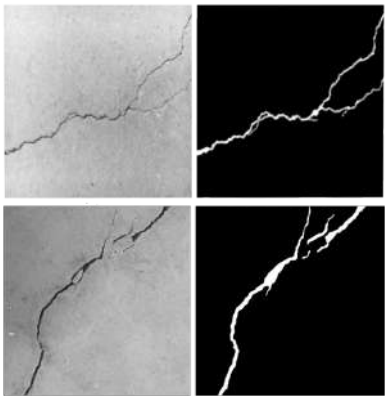

FIGURE 4.7: Examples of external subset with crevice segmentation from Heterogeneous dataset.

Surprisingly, all master models were able to recognize the tasks and performed well on a respective subset - almost on the same level as individual models (Table 4.10 shows F1 scores for subsets and for the whole dataset). Again there were no drastic performance boost for meta-models, but in general models with SE blocks worked somewhat better. However, the model with combined meta- and SE inputs showed

slightly better results on the weighted average F1 score for the Heterogeneous whole dataset (Figure 4.8).

Model \ Subset	Combined	CRACK	HeLa	HepG2
CRACK individual model		0.885		
HeLa individual model			0.901	
HepG2 individual model				0.807
Naked master model	0.835	0.884	0.888	0.794
SE master model	0.835	0.877	0.890	0.795
Meta master model	0.837	0.883	0.891	0.796
Meta+SE master model	0.842	0.890	0.893	0.803
Dummy+SE master model	0.837	0.886	0.891	0.797

TABLE 4.10: Performance on Heterogeneous dataset.

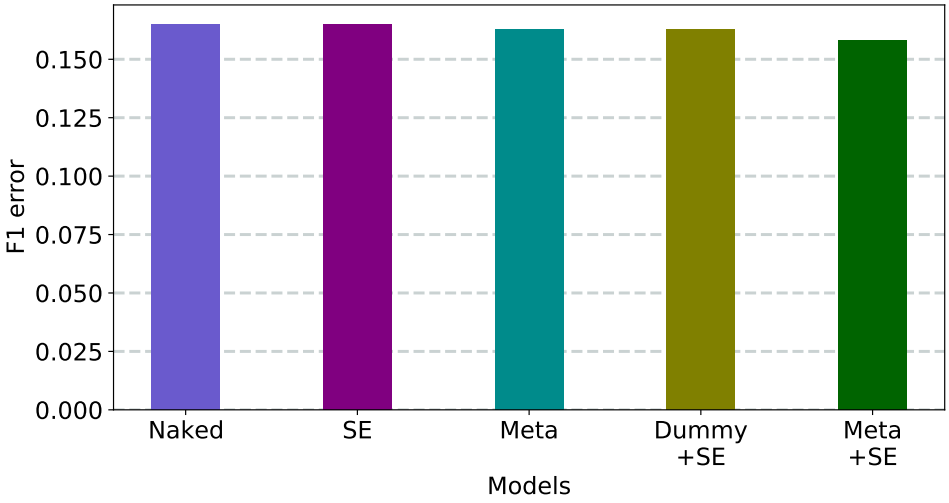

FIGURE 4.8: F1 error ranking of models from experiments on Heterogeneous dataset.

The models with meta-information demonstrate a solid response to meta-label change. There is a significant drop in the F1 score between CRACK and cell line subsets. It means the model has a strong separation in the prediction pathways for different tasks. Since the CRACK and cell line subsets have little features in common, the drop in scores is noticeable. At the same time, the fall between HeLa and HepG2 subsets is minor, implying these domains are alike. Thus test modes with different labels can serve for approximate determining how many features domains share. The example of test modes for the model Meta+SE is given in Table 4.11. "Normal" mode means the samples went to the model with native meta-labels. In “Shuffle” mode CRACK subset was paired with foreign label HeLa, HepG2 paired with CRACK, HeLa was paired with HepG2. In “Zeros” mode, the meta-label consisted of zeros permanently, in “Ones” mode consisted from ones, and during the modes “HeLa”, “HepG2”, “CRACK” these respective labels were permanent during the test run.

Subset \ Mode	Normal	Shuffle	Zeros	Ones	HeLa	HepG2	CRACK
Combined	0.842	0.458	0.746	0.755	0.752	0.772	0.098
HeLa	0.893	0.808	0.831	0.854	**0.893**	0.808	0.011
HepG2	0.803	0.000	0.671	0.674	0.681	**0.803**	0.000
CRACK	0.890	0.542	0.863	0.848	0.542	0.393	**0.890**

TABLE 4.11: F1 score response to various testing modes for the Meta+SE model from experiments on Heterogeneous dataset. "Normal"mode implies provision of native meta-labels with images from the respective subset. In "Zeros" mode only zeros are supplied as meta-labels throughout the full test, for "Ones" the meta-input is populated with 1.0, in "Shuffle" mode images were paired with non-native labels. In modes "HeLa", "HepG2", "CRACK" only one respective label is supplied for the whole test set.

Interesting observation: according to outcomes of the testing in different modes, the inference pathways of the CRACK domain use some features from cell line domains because prediction on CRACK subset with cell line label still produces some results on the CRACK subset. On the contrary, cell line pathways don't use CRACK features, obviously due to the fact CRACK features contain a lot of straight lines, which is uncommon in microscopy samples, while the concrete walls on the CRACK subset resemble the background of cell line images.

On this dataset, we also attempted to check a suggestion that good results of dummy models are caused by a lower reduction ratio in SE unit. In all experiments, we concatenate metadata using a constant reduction ratio of 2 (that determines the number of neurons in the hidden layer). In fact, this ratio appears to be slightly lower for meaningful inputs in the dummy model comparing to the pure meta-input in models without concatenation with SE. We retrained all models with ratios 1, 1.5, and 1.8 and eventually didn't observe a conceivable improvement with a lower ratio. There were cases of better scores, but the margin was small and inconclusive in statistical terms.

The main takeaway from the experiments with Heterogeneous dataset is that master models retain a capacity to perform well even for very different domains as long as they can implicitly infer the kind of task from the appearance of the sample. And they are capable of doing it without a significant drop in performance comparing to the individually trained models.

4.6 Experiments with Synthetic dataset

We came to the conclusion that the only application where the model can demonstrate its abilities is the case with similar domains but different tasks. An example from practice could be a multi-class segmentation (nuclei + cytoplasm) or multi-head topology. We had a small set of data for segmenting anomalies on microscopy images: some unusual areas, lesions, inclusions, debris, bubbles, etc. While preparing the large anomalies dataset, we conducted experiments with synthetic data. In this setup, meta-models were able to manifest their properties to the full extent.

We generated six classes: triangle, two types of a circle, two types of a square, and cross. These objects were scattered over the each image, and the model should predict only the required class based on the provided meta-label. We used a starter

code available in open access (Usuyama, 2020) . The data distribution in the Synthetic dataset is given in Table A.6 of Appendix A.

Model \ Subset	Combined	TRIANG	CIRCLE	FCIRCL	MSQUAR	SQUARE	CROSS
TRIANG model		0.986					
CIRCLE model			0.976				
FCIRCL model				0.985			
MSQUAR model					0.969		
SQUARE model						0.975	
CROSS model							0.956
Naked master model	0.001	0	0	0	0.002	0	0
SE master model	0	0	0	0	0	0	0
Meta master model	0.968	0.972	0.958	0.983	0.972	0.972	0.920
Meta+SE master model	0.965	0.978	0.955	0.974	0.973	0.965	0.914
Dummy+SE master model	0	0	0	0	0	0	0

TABLE 4.12: F1 scores of model from experiments on Synthetic dataset.

The tasks are done relatively well. While master models without meta-information are confused and fail to predict anything, the meta-models perform well in their respective domain/task with just little deterioration of domain-specific accuracy comparing to the individual models (Table 4.12). That means such metadata-driven model can be used as an alternative to multi-class or multi-head topology, effectively separating prediction pathways within layer channels. We used PPUnet as the backbone topology for this experiment.

The example of source images, predictions from metadata-driven models and ground truth masks are given in Appendix B (Figure B.1). We don't provide the F1 error plot for the models because it's evident that two metadata-driven models are absolute winners and perform on the same level on synthetic data.

The metadata-driven models also strongly respond on meta-label swap (Table 4.13), demonstrating the highest score only with native label and do not work with foreign labels (with partial engagement modes "zeros" , "ones" neither).

Subset \ Mode	Normal	Shuffle	Zeros	Ones	TRIANG	CIRCLE	FCIRCL	MSQUAR	SQUARE	CROSS
Combined	0.968	0.029	0.069	0.007	0.167	0.209	0.205	0.221	0.190	0.083
TRIANG	0.972	0.019	0.134	0.000	**0.972**	0.019	0.054	0.029	0.032	0.003
CIRCLE	0.958	0.039	0.020	0.020	0.038	**0.958**	0.039	0.038	0.060	0.009
FCIRCL	0.983	0.036	0.205	0.000	0.032	0.025	**0.983**	0.036	0.044	0.005
MSQUAR	0.972	0.039	0.007	0.003	0.029	0.043	0.029	**0.972**	0.039	0.013
SQUARE	0.972	0.007	0.041	0.000	0.028	0.036	0.052	0.047	**0.972**	0.007
CROSS	0.920	0.012	0.005	0.034	0.012	0.021	0.012	0.020	0.038	**0.920**

TABLE 4.13: F1 score response to various testing modes for the Meta model from experiments on Synthetic dataset. "Normal"mode implies provision of native meta-labels with images from the respective subset. In "Zeros" mode only zeros are supplied as meta-labels throughout the full test, for "Ones" the meta-input is populated with 1.0, in "Shuffle" mode images were paired with non-native labels. In modes "TRIANG", "CIRCLE", "FCIRCL", "MSQUAR", "SQUARE", "CROSS" only one respective label is supplied for the whole test set.

4.7 Experiments with Combo dataset

The experiments with the Synthetic dataset highlighted the cases where metadata-driven master models have advantages over the conventional ones. For example, the meta-information is helpful when we need to inform the model which output is required. This meta-input is even more beneficial when alternative outputs are not compatible. We designed an experiment where the model should predict different output from the same images based on the meta-label. The new Combo dataset consists of 2016 training samples from the Primary dataset with regular nuclei segmentation masks and 194 training samples from the same Primary pool but with different segmentation masks, outlining the anomalies on source images: blobs, debris, occlusions, non-cellular objects, optical defects, etc. In Appendix B there are few examples from this Combo dataset: source image with two different output masks (Figure B.2). The meta-information was encoded with two components, the larger part with nuclei segmented masks obtained the meta-label 7LINES, and the minor part with segmented anomalies was labeled as ANOM. The details of data distribution in the Combo dataset are in Table A.7 of Appendix A.

The backbone topology for these experiments was PPUnet. The results of running experiments with Combo dataset are the following: metadata-driven master models trained on both sets for different tasks can predict correct masks with high scores. And the good thing is that the master metadata-driven model gives a higher score on ANOM subset than the individual model trained exclusively on ANOM set. Master model without meta-input is confused and unable to infer ANOM tasks, it's totally suppressed by the larger 7LINES set. We also stopped using naked master model in this experiment as a redundant option. The Table 4.14 with results is given below. Models also strongly react to test modes, totally failing to predict task-specific masks during the meta-label swap.

Model \ Subset	Combined	ANOM	7LINES
ANOM individual model		0.736	
7LINES individual model			0.854
Master model (SE)	0.768	0.091	0.841
Master model (Meta)	0.844	**0.824**	0.845
Master model(Meta+SE)	0.849	**0.836**	0.850
Master model (SE+dummy)	0.767	0.104	0.835

TABLE 4.14: F1 score of the models from experiments on Combo set. The improvements are highlighted by maroon color.

In Appendix B in Figure B.3 there are few illustrated examples of how ANOM predictions improved on the metadata-driven model. The discovered phenomenon is similar to the effects observed during multi-task learning and is described in publications, with references in the chapter Related Work (Multi-task learning 2.7). Multi-task learning usually implies training a big master model with many output heads for different tasks. Such training usually has a synergic effect for each task comparing to the case when models are trained for each task separately. Our metadata-driven model has a single head and produces segmentation masks of the same format but for different segmenting tasks and sequentially in time.

To check if the effect is similar to those in real multi-task configuration, we constructed models with two heads having Unet3 as a backbone. There were two variants: high bifurcation when only the last level was duplicated and low bifurcation

when the full alternative decoder was recreated for a separate task (Figure 4.9). However, these models cannot operate both heads in parallel since the two sets are not equal - 7LINES is 10-fold larger, so the output for nuclei segmentation must occur ten times more often. So we still had to use a meta-label to redirect the source images to the respective head and applied a custom sampler that formed homogeneous batches consisting of samples from the same dataset. We called the model in this configuration "sequential" one.

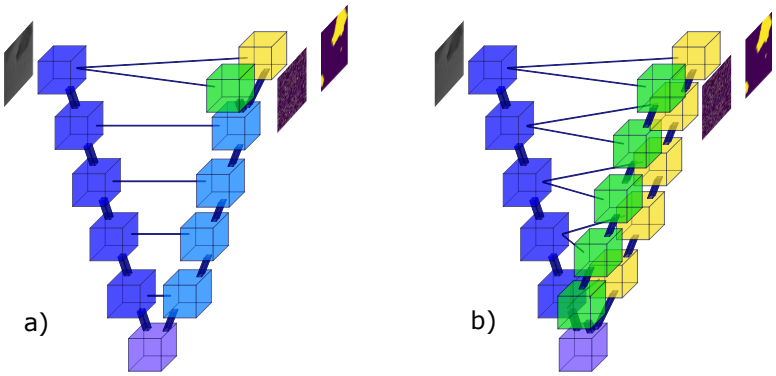

FIGURE 4.9: Multi-task learning model schema:
a) high bifurcation, b) low bifurcation.

The scores for dual-head models functioning in sequential mode showed the trend of improving ANOM scores from such kind of learning. We also included one dual-head model with low bifurcation trained on ANOM subset for reference to ensure the increased number of parameters is irrelevant to the observed phenomena of ANOM task improvement during the joint learning.

To make a genuine parallel model for multi-task learning, we curtailed the training dataset leaving only 194 source images that were common for both tasks (the validation set was also reduced in 7LINES part). The details of data distribution in such a modified dataset are given in Table A.8 of Appendix A. We kept the test set intact for the convenience of testing and compatibility. The models were altered for new processing, having two loss functions summed from each head. Again the effect of improving ANOM score is present. The scores for the 7LINES subset are lower because of a smaller number of samples from this domain. Led by curiosity, we launched another training on this curtailed dataset for our previous models in sequential mode with similar results. Also, the initial "mono" metadata-driven model was trained as well, giving the most remarkable boost for ANOM score among all experiments.

Finally, we separated the source images for the tasks. Since they had been common, there could be an effect of contrasting, so we replaced the images and masks for the nuclei segmentation task with the same number of images randomly picked from the Primary dataset. The "mono" metadata-driven model trained with such premise also demonstrated improvement for the ANOM subset. The results from all these experiments are summarized in Table 4.15 and models ranked by their contribution to ANOM score improvement are shown in Figure 4.10.

Model \ Subset(Task)	Combined	ANOM	7LINES
Full Combo set			
ANOM individual model (low bifurcation)		0.736	
7LINES (SOTA result for reference)			0.852
High-bifurcation sequential model	0.848	0.823	0.848
Low-bifurcation sequential model	0.851	0.836	0.851
Curtailed Combo set			
High-bifurcation parallel model	0.790	0.835	0.789
Low-bifurcation parallel model	0.793	0.833	0.792
High-bifurcation sequential model	0.793	0.829	0.792
Low-bifurcation sequential model	0.798	0.827	0.797
Mono meta-model	0.794	**0.854**	0.792
Mono meta-model (randomized 7LINES)	0.796	0.833	0.795

TABLE 4.15: F1 score for two-headed sequential, parallel and single headed model from experiments on Combo dataset (full and curtailed). The best improvement is highlighted by maroon color.

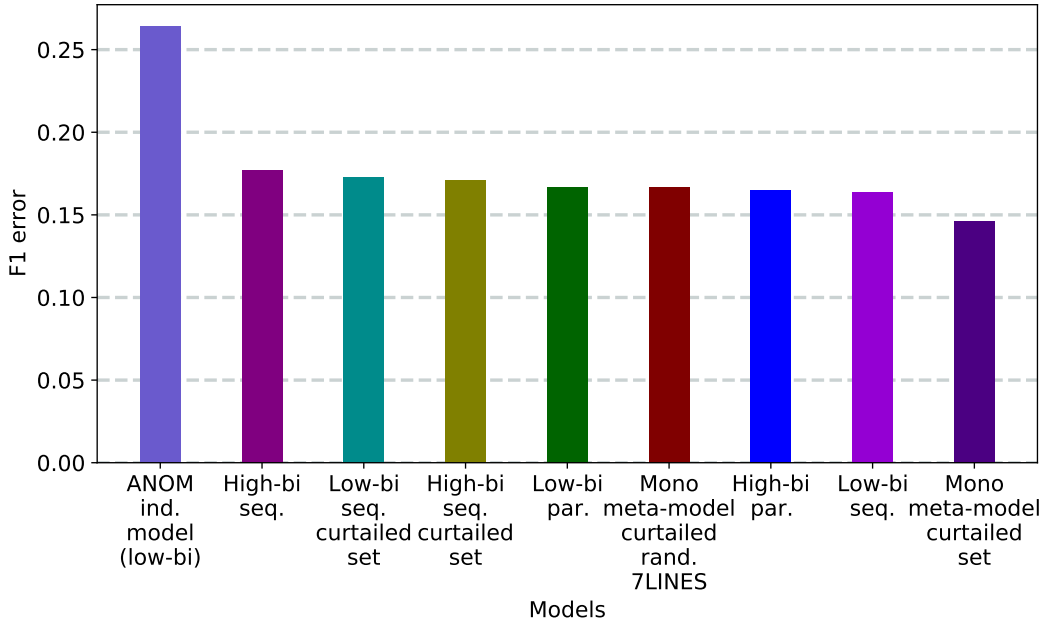

FIGURE 4.10: F1 error rate improvement on ANOM subset with multi-task models on Combo dataset. "Hi-bi" refers to double-headed high-bifurcated models, "Low-bi" to double-headed model with low bifurcation, "seq." means sequential mode and "par." - parallel mode, "rand." - randomized samples from 7LINES

4.8 Concluding experiments

The concluding experiments were dedicated to an ablation study on the positioning of Channel Attention blocks. We used the Synthetic dataset, Unet3 as a backbone model, and 50 epochs that suffice for quick convergence on this type of data. The tests on different modes reveal how models acquire and retain the ability to separate tacks by various channels.

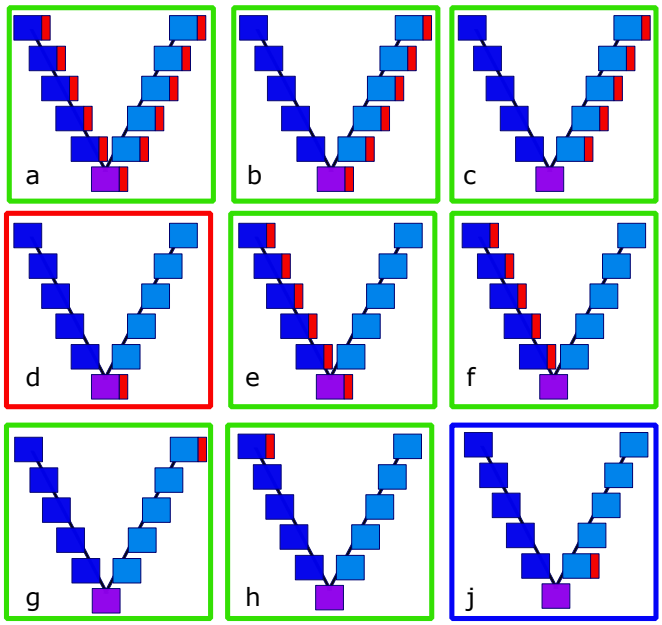

FIGURE 4.11: Variants of Unet3 with CA units positions

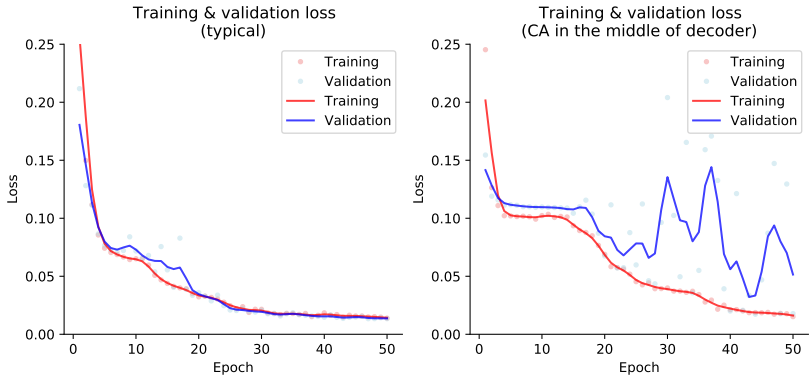

FIGURE 4.12: Training and validation loss curve typical for the successful model (left) and for the model with single CA block in the middle of decoder (right, variant *j* in ablation study).

The schema in Figure 4.11 contains the variants for placement of CA blocks, which are the SE legacy modules controlled exclusively by meta-information. The models retain task separation functionality even with a single block when placed at the beginning or near the model's output. The model with CA block in the middle (variant *d*) of Unet3 fails to perform inferences acting as master model without meta-input signifying ineffectiveness of CA unit on that position.

Single CA block placed in the middle of the decoder (variant *j*) still makes the model learn. However, the convergence is difficult, and the learning curve becomes unstable, but still possible. For example, in Figure 4.12, we exposed the learning curve for a successful training session typical for the rest of the models (left) and a somewhat wild validation curve for variant *j* with far digressions from the smooth descend.

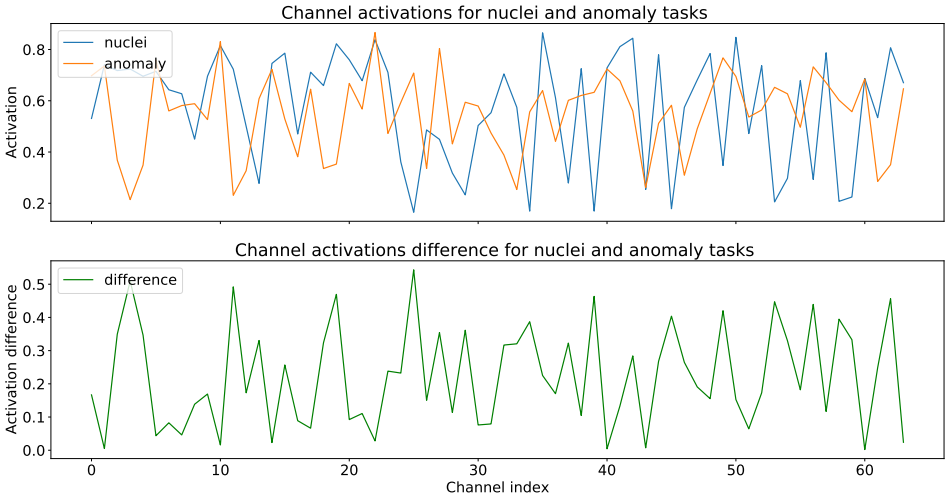

FIGURE 4.13: Channel activation (upper - absolute, lower - the difference modulus) for two tasks in the model with single CA unit.

The channel-wise activation vector from the CA unit provides the clue how tasks are specialized channel-wise. In Figure 4.13 there is a plot from the CA output on the model with a single CA block at the final stage of the decoder. This model is trained on Combo set with two tasks: 7LINES and ANOM. The upper plot shows the channel activation patterns for the same source image but for different tasks. The lower plot shows the absolute difference (modulus) between two activation vectors. We can infer that only several channels having a big difference in activation play a crucial role in delivering the task-specific output. The output convolution block reserves the last decision in channel selection, and it is plausible that only a few from 64 channels are used in the finalization of the output. This can be seen from the Figures B.4, B.5 of Appendix B where feature maps from different tasks still can be highlighted simultaneously in the CA unit output. Also, in the Figure B.6 of Appendix B we demonstrate the difference in channel activation vectors for all 11 CA units in the fully equipped Unet3 model (5 on the encoder + middle CA unit + 5 on the decoder).

Chapter 5

Conclusions

Summarising the results of our endeavors towards using metadata for cell nuclei segmentation by employing the Channel Attention Mechanism, we can group the conclusions into several items touching the various aspects of this work.

- *Metadata is digestible via Channel Attention Mechanism.*
  Adding metadata to Channel Attention units (which are the derivatives of Squeeze and Exatition blocks), we can effectively control the separation of features throughout the channels on each level on CNN. Even a single unit in a particular position suffices for such functionality.

- *Metadata-driven NN acts as an ensemble of individual models activated by respective labels.*
  Oftentimes ensembles may be trained on the same volume of data and differ in hyperparameter settings to express the various level of confidence in downstream meta-module for making the correct decision between predictions on the specific subset of data. Our models, on the contrary, have the fixed hyperparameters settings for all subsets but use different subsets to train individual expert submodels hidden in the internals of CNN channels, figuratively speaking. It's more accurate to describe it as a Mixture of Experts. The convenience of having a single master model instead of an ensemble of individual expert models should be obvious considering the reduction of training time and decreasing complexity of the system. It is unclear how many channel-dwelling submodels a single CNN can contain, but we may assume the number of domains can be pretty big for similar domains since many features are shared, and the number of NN parameters is high enough. Even in the hypothetical scenario, when the capacity of a single model might be depleted, the solution is scalable by increasing the number of channels.

- *The domain prediction pathways reside in separate channels.*
  Metadata literally highlights channels with features specific to an assigned domain and dims channels with foreign domain features (however, if domains are close in appearance, this separation is minimal, and many features are shared, expectedly). Thus the Channels Attention units functionally are a gating mechanism allowing the sharing of similar features and separation of incompatible features. Most useful this mechanism becomes for outputting the masks representing different tasks, which are impossible to retrieve from a single output in conventional models.

- *There is some benefit from using metadata for cell line and magnification domains, yet small.*
  The improvements around 1% in the F1 score do not look impressive, yet it is

statistically significant considering the amount of test data. Also, various cell lines show different responses to introducing metadata.

- *The performance gain could be more evident when individual expert models would outperform plain master models.*
  This claim is intuitive but yet remains speculation. We could not form the dataset from the available data where domain-specific models outperform plain master models with a significant margin on respective subsets. We hardly can expect a drastic gain in the performance of the metadata-driven model over the performance of the individual model when the plain master model also demonstrates high scores on subsets, not losing the accuracy at the expense of generalization. Instead, in many cases, we observed the converse effect of synergy in plain master models when their scores on subset exceeded the scores of domain-specific models. We assume that models can implicitly recognize tasks and domains, even when cross-domain distance is small, so supplementary metadata brings little information to add. In heterogeneous domains, master models could also recognize tasks and predict different masks with high accuracy comparable to individual models.

- *Metadata-driven NN can effectively separate similar segmentation tasks.*
  In the experiments with the Combo dataset, we demonstrated that a metadata-driven model could be trained to predict different masks that are otherwise incompatible in the plain master model with a single output due to confusion in the loss function. In a metadata-driven model, different outputs are possible because they reside in separate channels, and the final output layer only renders channels appropriate for the supplementary metadata. It is possibly the most obvious case where a metadata-driven model can outperform conventional master models. CNN is basically a kernel machine. On brightfield microscopy images, the nuclei outline is invisible in most cases, CNN infers segmentation masks from the cell boundary outline, suggesting the nuclei is located in the middle of the cell with more or less conventional shape. Suppose we have a cell line that looks similar to other cell lines, but internally, such cells have nuclei with different shapes or sizes. The plain master model would be confused and would segment the area common for nuclei from other domains. Only metadata can inform the NN model to output nuclei shape specific to the current cell line, and this specialization would result in a better total score. Unfortunately, we did not have such data samples in our repository, but we demonstrated the principle in the experiments with Synthetic and Combo datasets. In the medical industry, similar cases are likely as well. Suppose MRT scans or X-Ray shots of patients have similar organ shapes, but the interpretation of invisible parts may depend on external information, like the patient's age, treatment history, other metadata. In such cases, the segmentation of such images with metadata-driven models would be more accurate.

- *There are improvements for some tasks similar to effects in multi-task learning.*
  In the joint training of metadata-driven model with similar tasks, we could observe enhancement for the subset which was underrepresented in the dataset. The individual model on such subset performed worse than the metadata-driven master model. We hypothesized the beneficial influence of adjacent tasks could explain this effect. In the experiments with the Combo dataset, we proved this assumption by comparing the case with a genuine multi-task pipeline.

- *The metadata-driven NN can be a convenient alternative to multi-task(multi-head) models when tasks have the same format.*
  The benefit of it is a scalable pipeline, ready for accepting new datasets by simply denoting them with a new meta-label. In contrast, the whole architecture and pipeline should be refactored for the multi-headed models (or multi-class models with non-complementary classes) . The other convenience of our metadata-driven model is that the consecutive pipeline is agnostic to disproportions in the dataset and does not require the same source images for different tasks. In contrast, the multi-head model in classic implementation must have parallel output and shared source data. The metadata-driven model can find applications for transfer learning with the same convenience when fine-tuning is easily manageable by new meta-labels. One potential application for our model can be found in the artistic area. The style transfer techniques require a single model for each style. Using Channel Attention Mechanism, we can obtain as many models as possible using metadata to denote which style is required and even coherently mix styles in different proportions thanks to the floating-point format of metadata encoding. The restriction of our metadata-driven models is that tasks should be similar (like in this project, we had semantic segmentation with different output regions), because they share the same output, the same number of classes, and the same loss function.

We consider the project successful despite the modest performance boost in the experiments with Primary and Exhaustive datasets with conventional tasks. Nevertheless, the implications of how novel architecture works can be further investigated to find the appropriate niche and applications in the industrial and research fields.

Appendix A

Data Distribution

Cell line	Training set size	Validation set size	Test 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: Seven Cell Lines distribution

Source	Training set size	Validation set size	Test set size	Total
7LINES	2016	504	504	3024
AZ	628	78	78	784
Summary	2644	582	582	3808

TABLE A.2: Cell Lines augmented with AstraZeneca dataset

Magnification	Training set size	Validation set size	Test set size	Total
10x	843	218	230	1291
20x	862	215	233	1310
40x	872	209	206	1287
Summary	**2577**	**642**	**669**	**3888**

TABLE A.3: Exhaustive dataset, distribution by magnification

Cell line	Training set size	Validation set size	Test set size	Total
A549	435	97	114	646
NIH3T3	437	124	123	684
HELA	446	106	117	669
HepG2	409	116	107	632
HT1080	429	86	98	613
MCF7	421	113	110	644
Summary	**2577**	**642**	**669**	**3888**

TABLE A.4: Exhaustive dataset, distribution by cell lines

Source	Training set size	Validation set size	Test set size	Total
CRACK	333	56	56	445
HeLa	293	58	81	432
HepG2	283	82	67	432
Summary	**909**	**196**	**204**	**1309**

TABLE A.5: Data distribution in Heterogeneous dataset

Shape	Training set size	Validation set size	Test set size	Total
CIRCLE	100	20	20	140
CROSS	100	20	20	140
FCIRCL	100	20	20	140
MSQUAR	100	20	20	140
SQUARE	100	20	20	140
TRIANG	100	20	20	140
Summary	**600**	**120**	**120**	**840**

TABLE A.6: Data distribution in Synthetic dataset

Source	Training set size	Validation set size	Test set size	Total
7LINES	2016	504	504	3024
ANOM	194	70	101	365
Summary	**2210**	**574**	**605**	**3389**

TABLE A.7: Data distribution in Combo dataset

Source	Training set size	Validation set size	Test set size	Total
7LINES	194	70	504	768
ANOM	194	70	101	365
Summary	**388**	**140**	**605**	**1133**

TABLE A.8: Data distribution in curtailed Combo dataset

Appendix B

Trial results

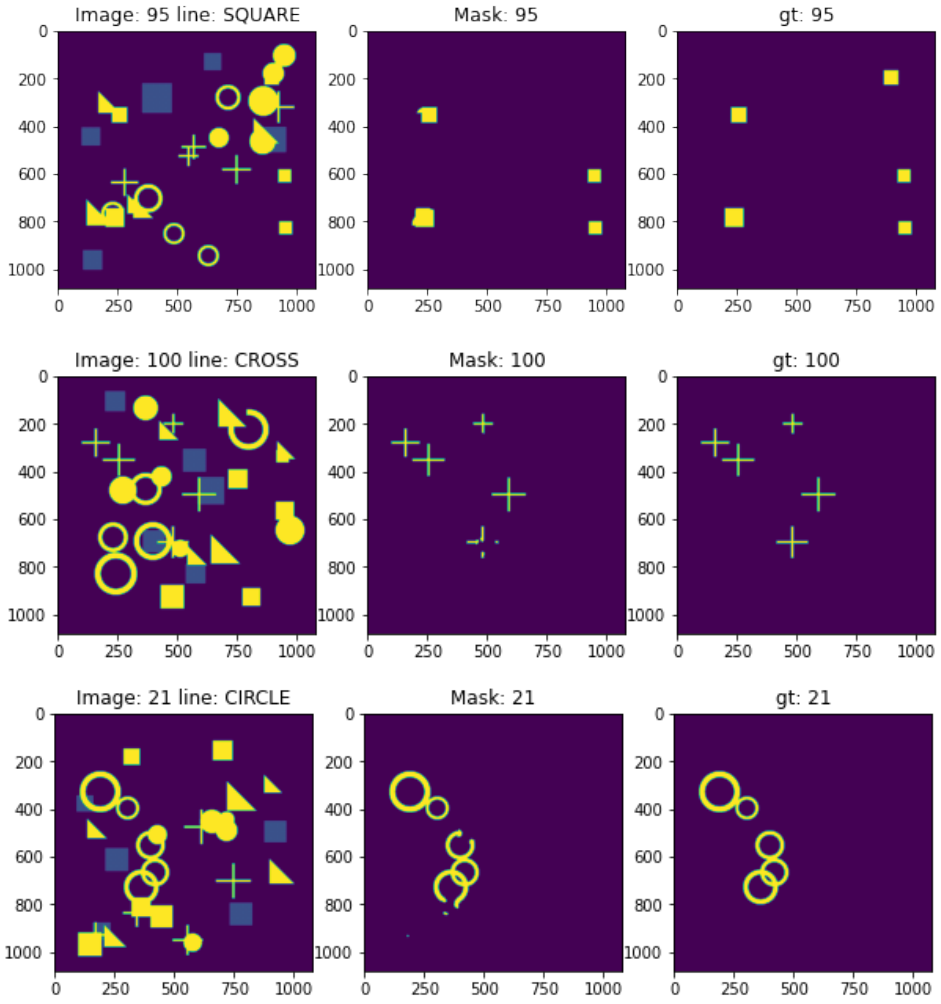

FIGURE B.1: Examples of Meta model predictions on Synthetic dataset: source, prediction, GT mask

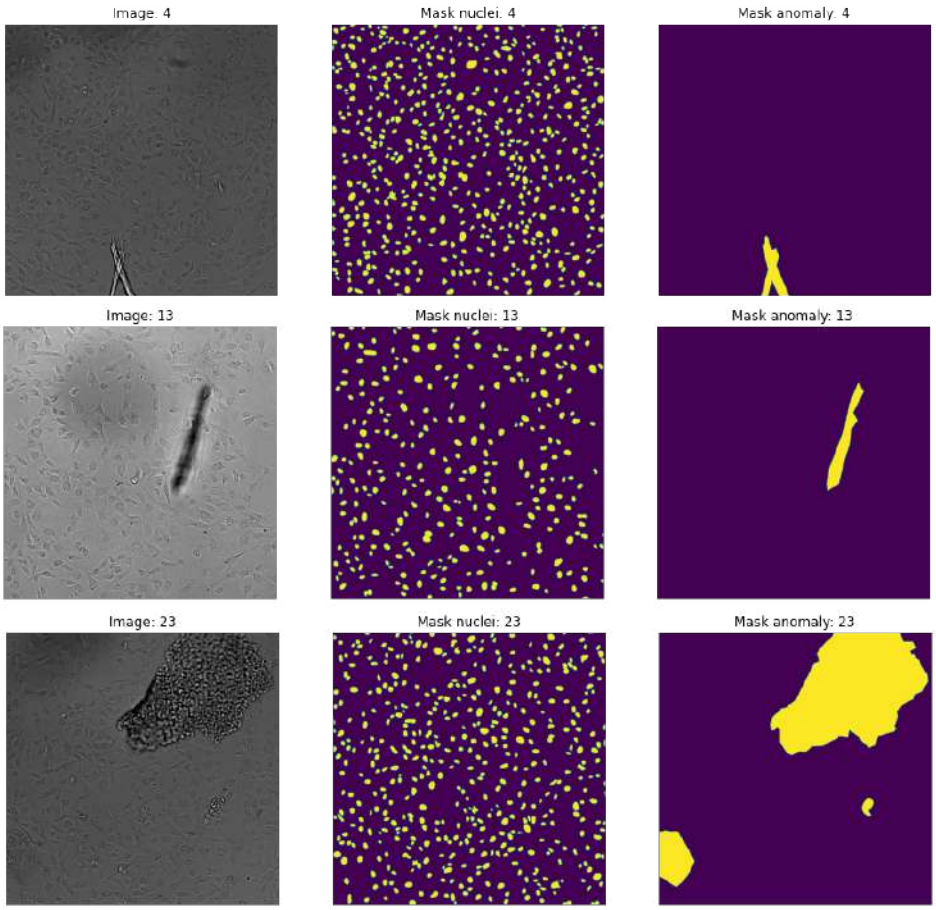

FIGURE B.2: Samples from Combo dataset with sorce images (left), nuclei segmented mask (center), anomaly segmented mask (right)

.

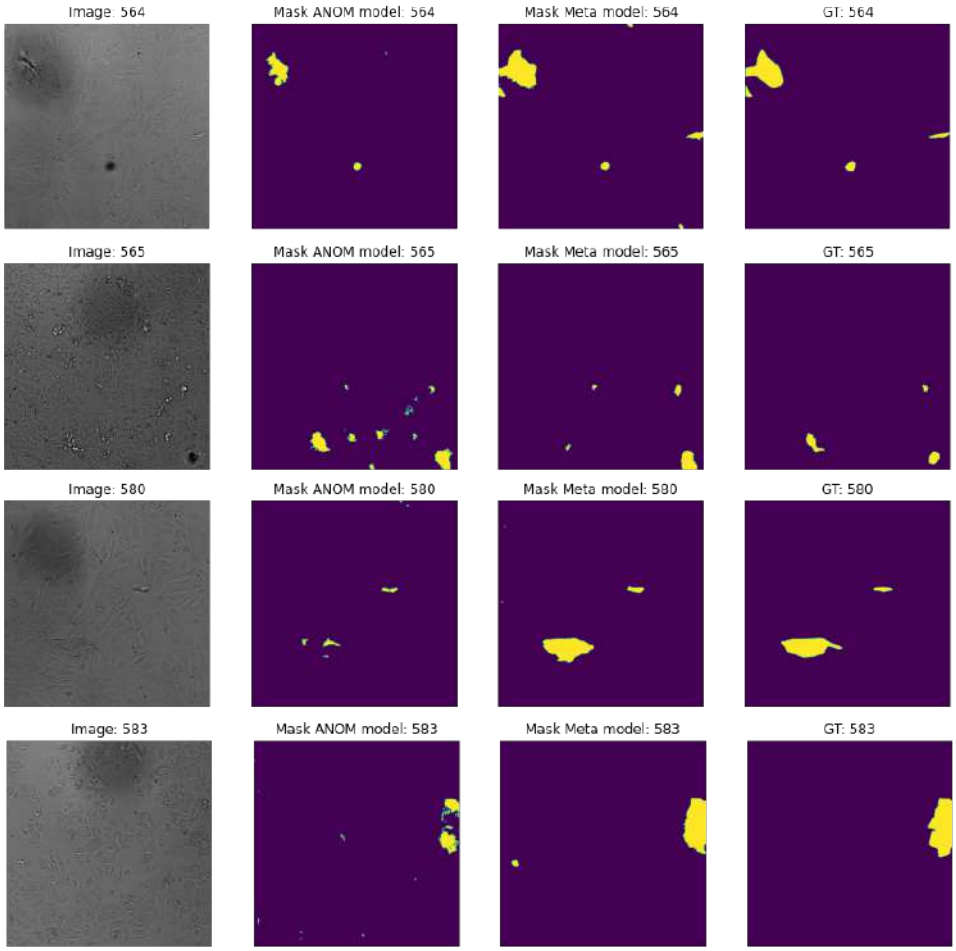

FIGURE B.3: Examples of prediction improvement with meta model. Leftmost - source image, second - prediction from individual ANOM model, third - prediction from Meta model, rightmost - ground truth.

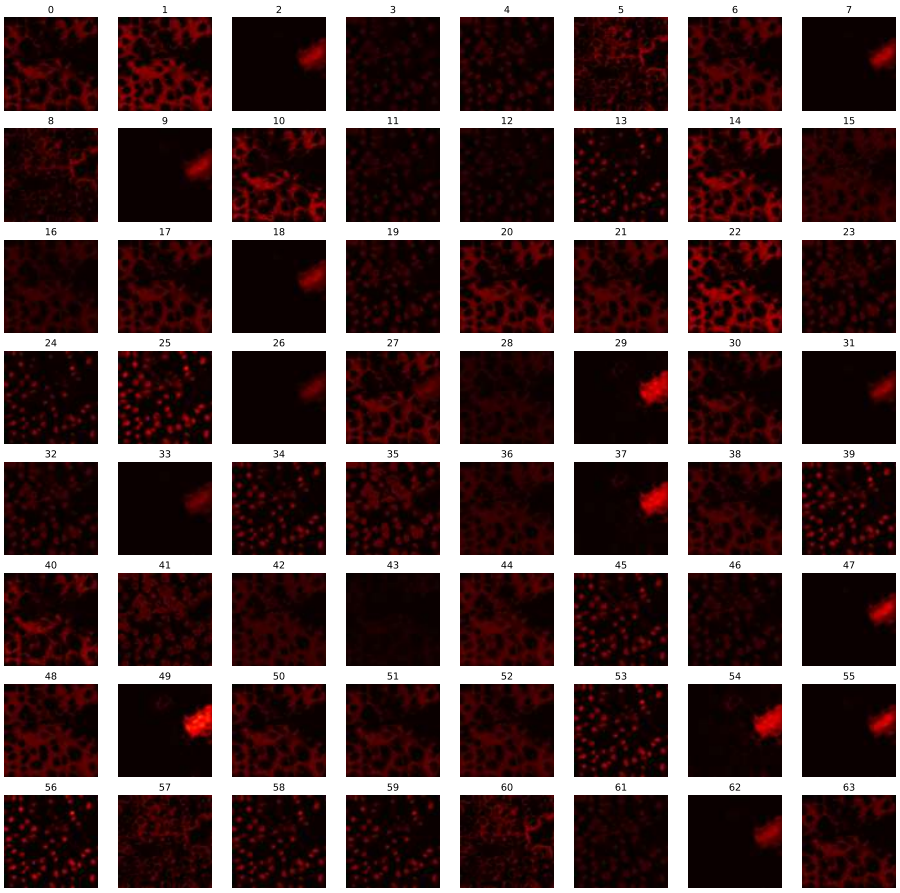

FIGURE B.4: Feature map at the output of a single CA unit for nuclei segmentation task.

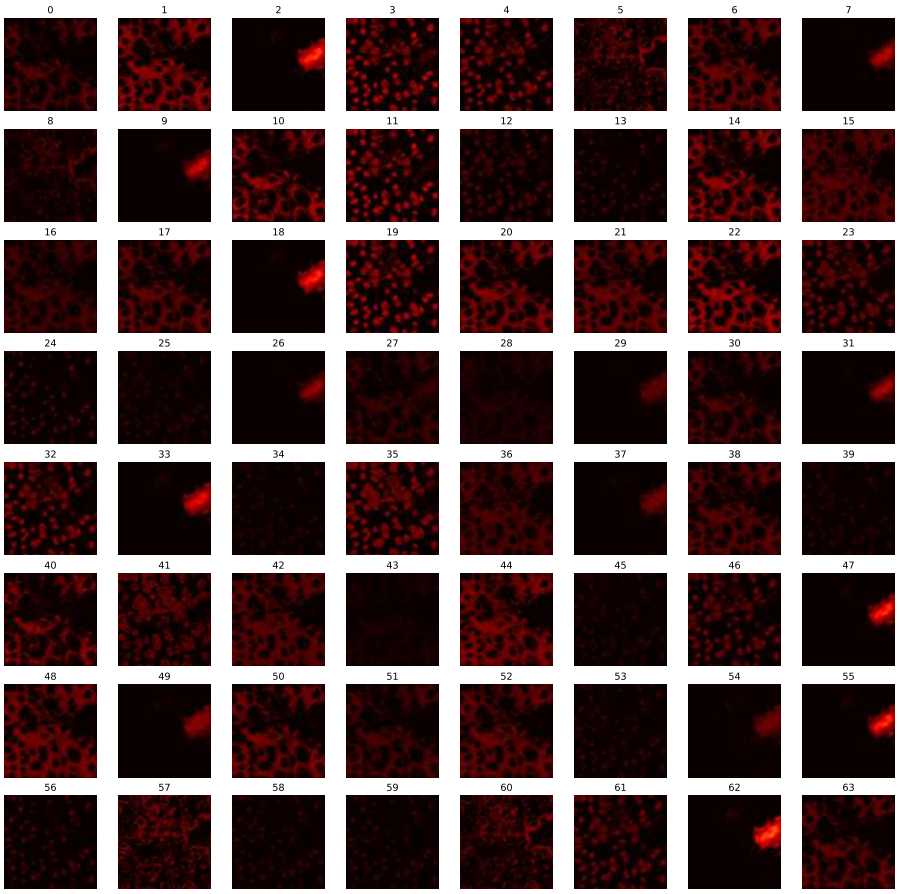

FIGURE B.5: Feature map at the output of a single CA unit for anomaly segmentation task.

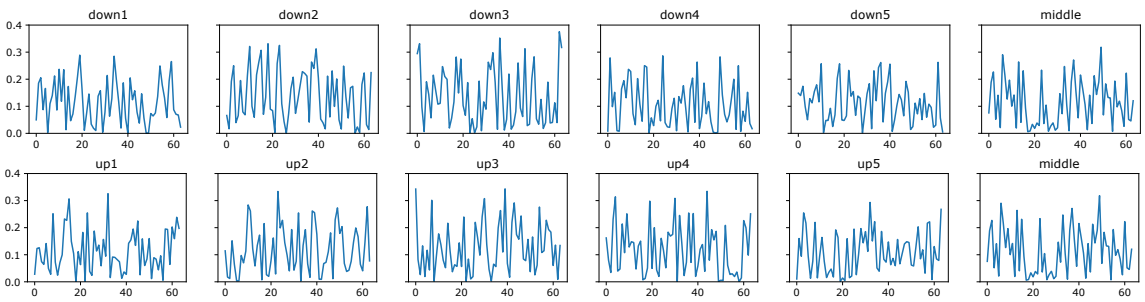

FIGURE B.6: Channel activation difference for the Combo tasks on Unet3 with 11 CA blocks (middle CA block is repeated on upper and lower plots.

Bibliography

Aljabar, P. et al. (2010). "Combining Morphological Information in a Manifold Learning Framework: Application to Neonatal MRI". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*. Ed. by Tianzi Jiang et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–8. ISBN: 978-3-642-15711-0.

AstraZeneca (2021). *AstraZeneca official site*. URL: https://www.astrazeneca.com/ (visited on 05/05/2021).

Belkin, Mikhail and Partha Niyogi (June 2003). "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation". In: *Neural Computation* 15.6, pp. 1373–1396. ISSN: 0899-7667. DOI: 10.1162/089976603321780317. eprint: https://direct.mit.edu/neco/article-pdf/15/6/1373/815527/089976603321780317.pdf. URL: https://doi.org/10.1162/089976603321780317.

Beucher, Serge (1979). "Use of watersheds in contour detection". In: *Proceedings of the International Workshop on Image Processing*. CCETT.

Bezdek, J. C., L. O. Hall, and L. P. Clarke (1993). "Review of MR image segmentation techniques using pattern recognition". In: *Medical Physics* 20.4, pp. 1033–1048. DOI: https://doi.org/10.1118/1.597000. eprint: https://aapm.onlinelibrary.wiley.com/doi/pdf/10.1118/1.597000. URL: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.597000.

Brosch, Tom and Roger Tam (2013). "Manifold Learning of Brain MRIs by Deep Learning". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*. Ed. by Kensaku Mori et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 633–640. ISBN: 978-3-642-40763-5.

Caruana, Rich (1997). "Multitask learning". In: *Machine learning* 28.1, pp. 41–75. DOI: 10.1023/A:1007379606734.

Chan, Philip K and Salvatore J Stolfo (1993). "Experiments on multistrategy learning by meta-learning". In: *Proceedings of the second international conference on information and knowledge management*, pp. 314–323.

Dietterich, Thomas G. (2000). "Ensemble Methods in Machine Learning". In: *Multiple Classifier Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–15. ISBN: 978-3-540-45014-6.

Fishman, Dmytro et al. (2019). "Segmenting nuclei in brightfield images with neural networks". In: *bioRxiv*. 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.

Gerber, Samuel et al. (2010). "Manifold modeling for brain population analysis". In: *Medical Image Analysis* 14.5. Special Issue on the 12th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) 2009, pp. 643–653. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2010.05.008. URL: https://www.sciencedirect.com/science/article/pii/S1361841510000617.

Gessert, Nils et al. (2020). "Skin lesion classification using ensembles of multi-resolution EfficientNets with meta data". In: *MethodsX* 7, p. 100864. ISSN: 2215-0161. DOI:

https://doi.org/10.1016/j.mex.2020.100864. URL: https://www.sciencedirect.com/science/article/pii/S2215016120300832.

Ghafoorian, Mohsen et al. (2017). "Transfer learning for domain adaptation in mri: Application in brain lesion segmentation". In: *International conference on medical image computing and computer-assisted intervention*. Springer, pp. 516–524.

Gray, Katherine R. et al. (2011). "Random Forest-Based Manifold Learning for Classification of Imaging Data in Dementia". In: *Machine Learning in Medical Imaging*. Ed. by Kenji Suzuki et al. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 159–166. ISBN: 978-3-642-24319-6.

He, Kaiming et al. (2015). "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.9, pp. 1904–1916. DOI: 10.1109/TPAMI.2015.2389824.

Hu, Jie, Li Shen, and Gang Sun (2017). "Squeeze-and-Excitation Networks". In: *CoRR* abs/1709.01507. arXiv: 1709.01507. URL: http://arxiv.org/abs/1709.01507.

Huang, Shih-Cheng et al. (2020). "Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines". In: *Digit. Med.* 3.136, pp. 1345–1359. DOI: https://doi.org/10.1038/s41746-020-00341-z.

Kawahara, Jeremy et al. (2019). "Seven-Point Checklist and Skin Lesion Classification Using Multitask Multimodal Neural Nets". In: *IEEE Journal of Biomedical and Health Informatics* 23.2, pp. 538–546. DOI: 10.1109/JBHI.2018.2824327.

Lemke, Christiane, Marcin Budka, and Bogdan Gabrys (2015). "Nonlinear dimensionality reduction combining MR imaging with non-imaging information". In: *Artif Intell Rev* 44, 117–130. DOI: https://doi.org/10.1007/s10462-013-9406-y.

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

Masoudnia, Saeed and Reza Ebrahimpour (2014). "Mixture of experts: a literature survey". In: *Artificial Intelligence Review* 42.2, pp. 275–293. DOI: 10.1007/s10462-012-9338-y.

Mendeley Data (2019). *Concrete Crack Segmentation Dataset*. URL: https://data.mendeley.com/datasets/jwsn7tfbrp/ (visited on 05/05/2021).

Misko, Oleh (2020). "Ensembling and transfer learning for multi-domain microscopy image segmentation". In:

Ngiam, Jiquan et al. (2011). "Multimodal Deep Learning". In: *ICML*, pp. 689–696. URL: https://icml.cc/2011/papers/399_icmlpaper.pdf.

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

Pan, Sinno Jialin and Qiang Yang (2010). "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.10, pp. 1345–1359. DOI: 10.1109/TKDE.2009.191.

Park, Hyunjin (2012). "ISOMAP induced manifold embedding and its application to Alzheimer's disease and mild cognitive impairment". In: *Neuroscience Letters* 513.2, pp. 141–145. ISSN: 0304-3940. DOI: https://doi.org/10.1016/j.neulet.2012.02.016. URL: https://www.sciencedirect.com/science/article/pii/S0304394012002030.

Pham, Dzung L., Chenyang Xu, and Jerry L. Prince (2000). "Current Methods in Medical Image Segmentation". In: *Annual Review of Biomedical Engineering* 2.1. PMID: 11701515, pp. 315–337. DOI: 10.1146/annurev.bioeng.2.1.315. eprint:

https://doi.org/10.1146/annurev.bioeng.2.1.315. URL: https://doi.org/10.1146/annurev.bioeng.2.1.315.

Raghu, Maithra et al. (2019). *Transfusion: Understanding Transfer Learning for Medical Imaging*. arXiv: 1902.07208 [cs.CV].

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241. DOI: https://doi.org/10.1007/978-3-319-24574-4_28.

Roy, Abhijit Guha, Nassir Navab, and Christian Wachinger (2018). "Recalibrating Fully Convolutional Networks with Spatial and Channel 'Squeeze & Excitation' Blocks". In: *CoRR* abs/1808.08127. arXiv: 1808.08127. URL: http://arxiv.org/abs/1808.08127.

Ruder, Sebastian (2017). *An Overview of Multi-Task Learning in Deep Neural Networks*. arXiv: 1706.05098 [cs.LG].

Tao, Lili and Bogdan J. Matuszewski (2013). "Non-rigid Structure from Motion with Diffusion Maps Prior". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: https://doi.org/10.1109/CVPR.2013.201.

Tenenbaum, Joshua B., Vin de Silva, and John C. Langford (2000). "A Global Geometric Framework for Nonlinear Dimensionality Reduction". In: *Science* 290.5500, pp. 2319–2323. ISSN: 0036-8075. DOI: 10.1126/science.290.5500.2319. eprint: https://science.sciencemag.org/content/290/5500/2319.full.pdf. URL: https://science.sciencemag.org/content/290/5500/2319.

Usuyama, Naoto (2020). *Synthetic images/masks for training*. URL: https://github.com/usuyama/pytorch-unet/ (visited on 04/15/2021).

Vaswani, Ashish et al. (2017). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL].

Wolz, Robin et al. (2011). "Manifold learning combining imaging with non-imaging information". In: *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 1637–1640. DOI: 10.1109/ISBI.2011.5872717.

— (2012). "Nonlinear dimensionality reduction combining MR imaging with non-imaging information". In: *Medical Image Analysis* 16.4, pp. 819–830. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2011.12.003. URL: https://www.sciencedirect.com/science/article/pii/S1361841511001733.

Zhang, Feihu and Benjamin W. Wah (Oct. 2017). "Supplementary Meta-Learning: Towards a Dynamic Model for Deep Neural Networks". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Zhang, Yu and Qiang Yang (Sept. 2017). "An overview of multi-task learning". In: *National Science Review* 5.1, pp. 30–43. ISSN: 2095-5138. DOI: 10.1093/nsr/nwx105. eprint: https://academic.oup.com/nsr/article-pdf/5/1/30/31567358/nwx105.pdf. URL: https://doi.org/10.1093/nsr/nwx105.

Zheng, Hao et al. (2019). "A New Ensemble Learning Framework for 3D Biomedical Image Segmentation". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 33.01, pp. 5909–5916. DOI: 10.1609/aaai.v33i01.33015909. URL: https://ojs.aaai.org/index.php/AAAI/article/view/4541.

Zhu, Bo et al. (2018). "Image reconstruction by domain-transform manifold learning". In: *Nature* 555, 487–492. DOI: https://doi.org/10.1038/nature25988.