Mitotic or Non-Mitotic?: Resolving Covariate Shifts in Multi-Domain Mitotic Figure Detection

Jiwon Jeong
jj835

Abstract

Incremental investigation of the MIDOG++ dataset provided insights about covariate shifts in the classification of mitotic figures of cancer histology images. For single-domain classification, AlexNet based models performed well. As the task grew to require cross domain generalizability, basic training methods were still performant when the target domain label data is accessible, but single domain trained models failed under covariate shifts. When target labels are removed or target distributions are entirely unavailable, empirical weighted risk minimization and meta-learning domain generalization algorithms allow models to generalize across covariate shifts. A final AlexNet-based model was trained with various covariate shift corrections to achieve a 0.6854 F1 score while naive algorithms achieved 0.6093.

1. Introduction

Histology images of tumor tissue is useful for identifying cancer types in tissues through identifying mitotic figures. Particularly, a large concern is if the cancer is mitotic or not mitotic, because it is a large indicator for the aggressiveness of the cancer [6]. For example, different histological patterns, like the density of mitotic figures, can help manual subclassification of soft-tissue sarcoma [9]. Classification by deep learning algorithms show promise to automate the identification of different tissue types, but ”domain shifts” like from different scanner sources limits these models from being generalized across all histology images [8]. Naive computer vision models trained on multi-domain histology data do not generalize well across covariate shifts to unseen scanner, tumor, or facility sources. The MIDOG++ dataset provides histology images of 10 different sources [1]. This investigation will partition the images by sources to test the generalization of mitotic vs non-mitotic figure classification models to unseen distributional shifts. Through insights from experiments on controlled subsets of the MIDOG++ dataset, a shift-resistant model is built to classify between mitotic or non-mitotic figures in unseen domains.

1.1. Related Work

Prior work on the MIDOG++ dataset combine the data by tumor type. When training on a single tumor type or domain, models classify well on images from the same tumor (0.73 average F1 score). Multi-domain models also perform well. However, models trained on single tumor types perform very poorly on unseen tumor type images (with 0.4-0.5 F1 scores). Performing a leave-one-out training strategy tends to show better generalization on unseen domains (0.67 average F1 score). Overall the work shows high covariate shifts between tumor domains and notable improvement through leave-one-out training methods [1].

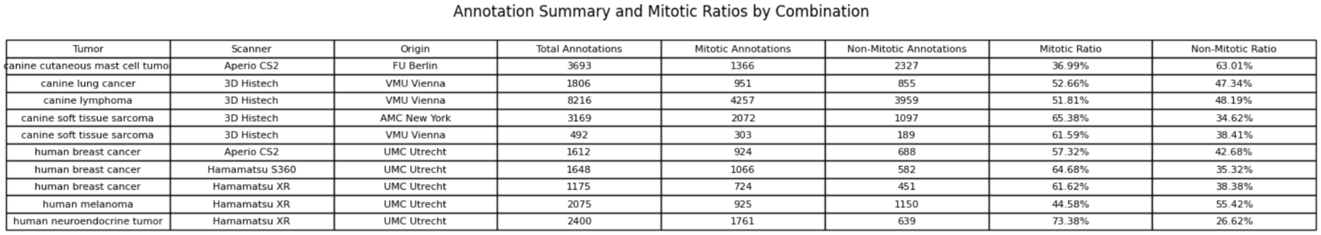
1.2. Dataset description

MIDOG++ dataset contains 26286 annotations of structures on 553 images. The images come from a variety of different domains with varying tumor types, scanners, and facilities. Most sources contribute about 50 images, but VMU Vienna facility only contributes 15 images. There is some class imbalance seen in Figure 1, but most annotation sources have around a 60/40 split. Each annotation is a 50 by 50 pixel box with the structure of interest in the center. The dataset provides the original images and dimensions of the annotations. Quick visualization of annotations from the same tumor type and facility (human breast cancer from UMC Utrecht) but with different scanners Hamamatsu S360 (Figure 2a) and Hamamatsu XR (Figure 2b) show visual differences in color.

Averaging annotation images within distinct sources also show differences beyond a uniform color filter. For example, the average mitotic structures of 3D Histech|VMU Vienna data are darker purple, but only the canine lymphoma tumor type has a lighter ring around the main structure.

2. Methods

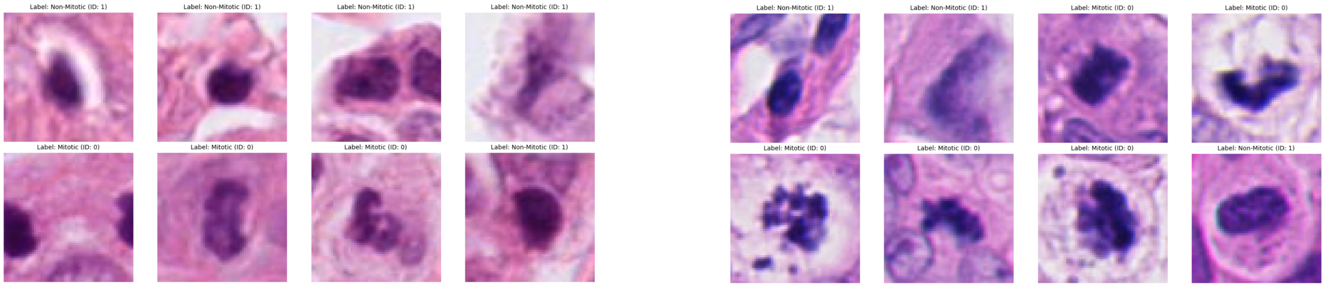
This investigation takes an incremental approach to the MIDOG++ dataset, creating controlled distributional shifts to generate incremental insights. The subset of the dataset changes with experiments. Throughout AI code assistance was used [2].



Annotation Summary and Mitotic Ratios by Combination

Tumor	Scanner	Origin	Total Annotations	Mitotic Annotations	Non-Mitotic Annotations	Mitotic Ratio	Non-Mitotic Ratio
canine cutaneous mast cell tumo	Aperio CS2	FU Berlin	3693	1366	2327	36.99%	63.01%
canine lung cancer	3D Histech	VMU Vienna	1806	951	855	52.66%	47.34%
canine lymphoma	3D Histech	VMU Vienna	8216	4257	3959	51.81%	48.19%
canine soft tissue sarcoma	3D Histech	AMC New York	3169	2072	1097	65.38%	34.62%
canine soft tissue sarcoma	3D Histech	VMU Vienna	492	303	189	61.59%	38.41%
human breast cancer	Aperio CS2	UMC Utrecht	1612	924	688	57.32%	42.68%
human breast cancer	Hamamatsu S360	UMC Utrecht	1648	1066	582	64.68%	35.32%
human breast cancer	Hamamatsu XR	UMC Utrecht	1175	724	451	61.62%	38.38%
human melanoma	Hamamatsu XR	UMC Utrecht	2075	925	1150	44.58%	55.42%
human neuroendocrine tumor	Hamamatsu XR	UMC Utrecht	2400	1761	639	73.38%	26.62%

Figure 1. Distribution of label types across annotation sources.


Label: Non-Mitotic (ID: 1)
Label: Non-Mitotic (ID: 1)
Label: Non-Mitotic (ID: 1)
Label: Non-Mitotic (ID: 1)
Label: Mitotic (ID: 0)
Label: Mitotic (ID: 0)
Label: Mitotic (ID: 0)
Label: Non-Mitotic (ID: 1)
Label: Non-Mitotic (ID: 1)
Label: Non-Mitotic (ID: 1)
Label: Mitotic (ID: 0)
Label: Mitotic (ID: 0)
Label: Mitotic (ID: 0)
Label: Mitotic (ID: 0)
Label: Mitotic (ID: 0)
Label: Non-Mitotic (ID: 1)

(a) Scanned with Hamamatsu S360

(b) Scanned with Hamamatsu XR

Figure 2. Annotations of mitotic figures from two different scanners for the same cancer type and facility origin. Images come from human breast cancer from UMC Utrecht.

2.1. Experiment 1: Single-domain classification

To avoid issues of distributional shift, a single domain subset of the MIDOG++ dataset was selected. **Dataset.** 8216 annotations of the canine lymphoma images|3D Histech|VMU Vienna images were split.

2.1.1. Baseline CNNs

Two baseline CNNs were trained. One based on LeNet and the other based on the AlexNet architecture. **Preprocessing.** All annotations were cropped from the images in 50 by 50 pixel inputs. **Network 1.** A variant of LeNet-5 was used with minimal modernizations to adapt to PyTorch libraries[4]. Alterations include replacing Gaussian activation layers with softmax layers [10]. **Network 2.** A variant of AlexNet was used with minimal modernizations to adapt to the smaller pixel size of the MIDOG++ annotations[3]. Only the first convolutional kernel size and padding was reduced to 7 and 5, respectively. **Training.** The network was initialized with Kaiming Uniform and trained with PyTorch SGD for 50 epochs with a learning rate of 0.5. **Evaluation.** Accuracy of validation set batches across training and accuracy of entire validation set after training was measured.

2.1.2. Factorizing improvements to AlexNet

The base LeNet architecture was updated to get closer to AlexNet through factorized changes and evaluated on performance. **Preprocessing.** All annotations were cropped from the images in 50 by 50 pixel inputs. **Networks.** 4 additional networks were built, all based on changing one choice in LeNet towards AlexNet: dropout, activation, pooling, and depth. Then 2 additional networks were built that combines changes in activation and dropout and changes in activation and pooling. And a final network combines three changes in activation function, pooling, and dropout. **Training.** The network was initialized with Kaiming Uniform and trained with PyTorch SGD for 50 epochs with a learning rate of 0.5. **Evaluation.** Accuracy of validation set batches across training and accuracy of entire validation set after training was measured.

2.1.3. Data augmentations

Different data augmentations were individually tested. **Preprocessing.** All annotations were cropped from the images in 50 by 50 pixel inputs. Different augmentation techniques were individually tested: 50% random horizontal flipping, vertical flipping, brightness, saturation, hue, and contrast jitter. A baseline dataset was also tested. **Network.** All augmentation techniques trained on LeNet-5 with ReLU, max pooling, and dropout modernizations. **Training.** The network was initialized with Kaiming Uniform and trained with PyTorch SGD for 10 epochs with a learning rate of 0.5. **Evaluation.** Accuracy of validation set and training set after training was recorded.

2.1.4. Padding methods

Different padding methods were individually tested on the first convolutional layer. **Preprocessing.** Most annotations were cropped from the images in 50 by 50 pixel inputs.

One exception is the ”image padding” method that replaces padding by using the existing surrounding image data. All datasets augmented with random brightness jitters, horizontal flips, and vertical flips. **Networks.** All networks are based on the same modified AlexNet, where the first convolutional kernel is reduced to 7 and padding size is reduced to 5. From this, four different padding methods were tested only on this first layer. Zero padding, reflect padding, replicate padding, and image padding were used. **Training.** The network was initialized with Kaiming Uniform and trained with PyTorch SGD for 50 epochs with a learning rate of 0.5. **Evaluation.** Accuracy of validation set and training set after training was recorded.

2.1.5. RegNets

More modern CNN networks were used. RegNet includes modern improvements like batch norm, network in network, and residual networks [10]. **Preprocessing.** All annotations were cropped from the images in 50 by 50 pixel inputs. All datasets augmented with random brightness jitters, horizontal flips, and vertical flips. **Baseline RegNetX32.** RegNetX32 was used [7]. **Baseline RegNetX32 Training.** The network was initialized with Kaiming Uniform and trained with PyTorch SGD for 50 epochs with a learning rate of 0.5. **RegNet-based Networks.** Then variations of RegNetX32 following network design space suggestions [7] led to variation experiments where network hyperparameters are independently varied. Group size was tied across blocks and fixed at 16. No bottleneck ratios were performed on the ResNeXt block layer channels. **Stem Channels**. Stem channels of 8, 16, 32, 64 were tested on RegNetX32. **Last Block Depth.** Last block depth of 8, 16, 32, 64 were tested on RegNetX32 with stem channel size of 8. **First Block Depth**. First block depth of 8, 16, 32, 64 were tested on RegNetX32 with stem channel of 8 and last block depth of 8. **RegNet-based Network Training.** The network was initialized with Kaiming Uniform and trained with PyTorch SGD for 10 epochs with a learning rate of 0.5. **Evaluation.** Accuracy of validation set and training set after training was recorded. These accuracies were considered in relation to the parameter size of the networks.

2.2. Designing channel magnitude and ratio network space

The channel sizes within blocks of stages were tied according to suggested design spaces [7]. However, the channel size choice between the first and second stage were varied to generate insights about the network distribution. **Preprocessing.** All annotations were cropped from the images in 50 by 50 pixel inputs. All datasets augmented with random brightness jitters, horizontal flips, and vertical flips.**Reference RegNet.** All RegNet variants had depth of 8 and 32 for the first and second stages. Group size and stem channel size was 16 and 8 respectively. No bottleneck ratios were performed on the ResNeXt block layer channels. **Channel design space.** The ratio of channel size between the first and second stages were varied between 1:1, 1:2, 1:4, 1:8, 1:16 where the larger channel is always the second stage. For each ratio, the first stage channel was varied between 16, 32, 64, 128, and 256. Training. **Training.** The network was initialized with Kaiming Uniform and trained with PyTorch SGD for 5 epochs with a learning rate of 0.5. **Evaluation.** Accuracy of validation set and training set after training was recorded. These accuracies were considered in relation to the parameter size of the networks. **Cumulative network distribution.** The cumulative accuracy distribution over networks following either ratio or first stage channel size design restrictions were plotted according to the following empirical approximation.

$$F(e, Z) = \frac{1}{n} \sum_{i=1}^{n} 1(e_i \leq e)$$

Where F is the sample cumulative distribution of accuracies e over sample of networks Z [10].

2.3. Experiment 2: Data-abundant covariate shifts

Next a scenario was constructed where the model has some access to different domain data. The model is allowed to access target domain distributions during training. **Dataset.** This experiment uses human breast cancer |* |UMC Utrecht data, thus experiencing scanner covariate shifts. **Preprocessing.** All annotations were cropped from the images in 50 by 50 pixel inputs. All datasets augmented with random brightness jitters, horizontal flips, and vertical flips. **Training.** Unless stated otherwise, the network was initialized with Kaiming Uniform and trained with PyTorch SGD for 50 epochs with a learning rate of 0.5.

2.3.1. Cross domain generalization on single-domain and naive combined models

Data access. Models can only be trained on single domains and have no access to other domains beyond validation. For reference, a combined model was also trained. **Network 1.** The modernized LeNet-5 network with modernized ReLU, max pooling, and dropout was used. **Network 2.** In addition, models were trained with batch norm layers before every nonlinearity. **Evaluation** Validation accuracy of models on their own domain and across the other two domains was computed after training.

2.3.2. Empirical weighted risk minimization

Data access. Models have access to training data across all domains. The domain source and target domain is also known. However, the target domain labels are unknown, so it is different from a simple combined domain model. **Network.** The modernized LeNet-5 network with modernized ReLU, max pooling, batch norm, and dropout was used. **Weighted empirical risk minimization.** A classifier to identify if data points belong to the target domain was trained. Then this was used to weight data points during training. Specifically, the loss of data points was weighted by $e^{h(x_i)}$, where x_i is a data point and h(x) is the classifier. **Evaluation** Validation accuracy of models on their own domain and across the other two domains was computed after training.

2.4. Experiment 3: Data-deficient covariate shifts with single domain training

This experiment attempts to resolve covariate shifts when there is no data of the target distribution during training.

2.4.1. Augmentations and generalization

Due to the initial visualization of different colors and hues in scanners, it was hypothesized that aggressive augmentation methods can assist generalization. **Dataset.** This experiment uses canine soft tissue sarcoma |3D Histech |AMC New York or VMU Vienna data, thus experiencing facility origin covariate shifts. The VMU Vienna data is very small, so it is appropriate as a data-deficient target distribution. **Data access.** The models only have access to the AMC New York data. VMU Vienna data can only be access for evaluation purposes. **Preprocessing.** All images are cropped to 50 by 50 pixel of the annotation. Different augmentation techniques were independently applied including aggressive brightness, contrast, saturation, and hue jitters. **Networks and Training.** Models were trained on modernized LeNet with ReLU, max pooling, dropout, and batch norm. They were initialized with Kaiming Uniform and trained for 50 epochs on 0.5 learning rate. **Evaluation.** Batch validation accuracy was tracked on AMC New York data and the target domain (VMU Vienna) was only accessed for final evaluation of the trained models.

2.5. Experiment 4: Data-deficient covariate shifts with multi domain training

2.5.1. Meta-learning domain generalization

With only a single domain source and the target (inaccessible) target domain, the previous methods were limited. With the addition of more domain sources with at least some similarities to the canine soft tissue sarcoma|3D Histech |VMU Vienna data, new algorithms could be applied. **Dataset.** This experiment uses * |3D Histech |* data, thus experiencing multiple forms of covariate shift in tumor and facility origin. The VMU Vienna data is very small, so it is appropriate as a data-deficient target distribution. **Data access.** Models do not have access to VMU Vienna data. Other 3D Histech data is accessed. **Preprocessing.** All images are cropped to 50 by 50 inputs with brightness jitter and horizontal and vertical flips. **Networks and Training.** Models were trained on modernized LeNet with ReLU, max pooling, dropout, and batch norm. They were initialized with Kaiming Uniform and trained for 30 epochs on 0.5 learning rate. **Meta-learning domain generalization (MLDG) algorithm.** The MLDG splits the seen training domains into meta-test domains and meta-train domains. These allocations of the seen training data is continually resplit and the model attempts to simultaneously optimize both the meta-test and meta-train sets [5]. The MLDG algorithm is applied to one model with a normal training performed as benchmark **Evaluation.** The models are evaluated on accuracy of seen sources and the unseen VMU Vienna data.

2.5.2. Scaling to entire dataset

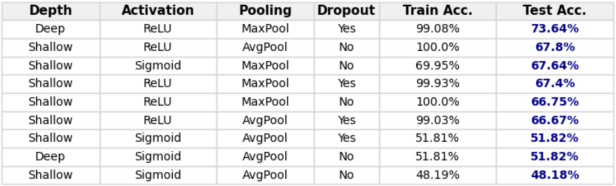
In the final experiment, the MLDG algorithm is applied to the entire dataset where all of the domains are possible training sources except for canine cutaneous mast cell tumor |Aperio CS2 |FU Berlin, which acts as an unseen target domain. **Data access.** Models can train on all domains, but do not have any access to the FU Berlin data. **Preprocessing.** All images are cropped to 50 by 50 inputs with brightness jitter and horizontal and vertical flips. **Networks.** The adjusted AlexNet is used with only the first convolution reduced to kernel size of 7 and padding of 5. **Training.** Models are initialized with Kaiming Uniform. Learning rate is adjusted between 0.005, 0.001, and 0.0001. Meta learning rate is adjusted between 0.01, 0.05, 0.001, and 0.005. **Evaluation.** Models are evaluated on accuracy and F1 score to the unseen FU Berlin data.

3. Results

3.1. Experiment 1: Single-domain classification

Baseline CNN. The baseline LeNet-5 model performed very poorly, reaching a 0.4818 validation accuracy. The accuracy throughout the training did not change much and kept jittering between 0.4818 and 0.5182. The AlexNet-based model reaches a 0.7364 validation accuracy. Throughout training, validation accuracy was observed to increase minimally beyond 20 epochs while training accuracy kept increasing up to 40 epochs. **Factorizing improvements to AlexNet.** Figure 3 summarizes results of the various LeNet-5 modernizations. Notably, the base LeNet-5 starts at the worst accuracy and the AlexNet ends up with the best accuracy. Independent variations between LeNet and AlexNet lie on a spectrum between.

Data augmentations. Comparing the validation accuracies, brightness jitter (0.658), vertical flips (0.656), and horizontal flips (0.656) result in marginally better accuracies than the baseline with no augmentation (0.655). Saturation and hue perform marginally worse (0.643) while contrast jitters performs the worst (0.620). Interestingly the train accuracy follows a different trend where saturation and hue jitters have higher accuracies (0.775, 0.748) than the baseline (0.729), and all others have lower accu-



Depth	Activation	Pooling	Dropout	Train Acc.	Test Acc.
Deep	ReLU	MaxPool	Yes	99.08%	**73.64%**
Shallow	ReLU	AvgPool	No	100.0%	**67.8%**
Shallow	Sigmoid	MaxPool	No	69.95%	**67.64%**
Shallow	ReLU	MaxPool	Yes	99.93%	**67.4%**
Shallow	ReLU	MaxPool	No	100.0%	**66.75%**
Shallow	ReLU	AvgPool	Yes	99.03%	**66.67%**
Shallow	Sigmoid	AvgPool	Yes	51.81%	**51.82%**
Deep	Sigmoid	AvgPool	No	51.81%	**51.82%**
Shallow	Sigmoid	AvgPool	No	48.19%	**48.18%**

Figure 3. Model performance of modernizations on LeNet-5

Table 1. Evaluation of padding methods

Padding	Train Accuracy	Val Accuracy
Zero	0.845	0.783
Image	0.857	0.779
Reflect	0.863	0.777
Replicate	0.846	0.772

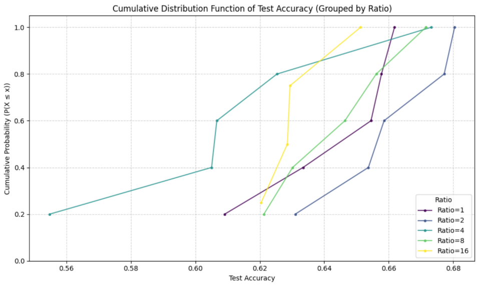


Figure 4. Cumulative distribution of networks over depth ratios

racy than the baseline. **Padding methods.** From Table 1, zero padding performs marginally better than others on the validation accuracy.

RegNets. Limiting stem size to 8 resulted in the best validation performance at 0.757. Increasing the last block depth increases performance from 0.706 to 0.738 from 8 depth to 64, but at the cost of 617.81% parameter count increase. Interestingly train accuracy drops slightly with increasing last block depth. Changing first block depth has negligble improvements on train or test accuracy from 0.718. Actually increasing the block depth to 16 significantly hurt performance.

Designing channel magnitude and ratio network space. In Figure 4 , the empirical cumulative network distribution over ratios show that networks with depth ratio of 1:2 contains all of the other distributions. Also, the empirical cumulative network distribution over magnitudes weakly shows that the 256 and 128 first depth size is contained by the other distributions.

3.2. Experiment 2: Data-abundant covariate shifts

Cross domain generalization on single-domain and naive combined models. Single-domain models perform well on their own domain. Generally, they perform poorly when used to evaluate on other domains. The combined model performs better on all domains than the single-domain trained models when early stopping is applied. **Empirical weighted risk minimization.** As seen in Figure 5, the naive combined model performs worse on validation accuracy. When the model is weighted, the performance increases for the domain it is targeted towards. Interestingly, the weighted models also perform better than the baseline combined model on domains that it was not targeted for.

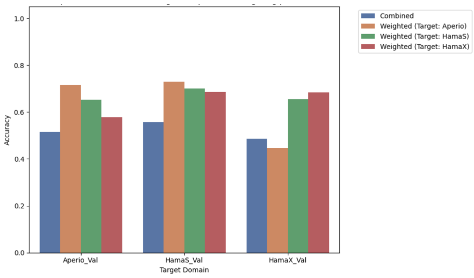


Figure 5. Cross-domain performance of empirical weighted risk minimization

Table 2. Evaluation of MLDG-based model

Model	Seen Val Accuracy	Unseen Val Accuracy
Naive	0.7131	0.7114
MLDG-based	0.6480	0.7195

3.3. Experiment 3: Data-deficient covariate shifts with single domain training

Augmentations and generalization. Validation accuracies with addition of strong contrast and saturation jitter is higher with seen domain evaluation. Overall accuracies are similar. However, all augmentation strategies of strong brightness, contrast, saturation, and hue result in decreased performance on unseen domains compared to the baseline.

3.4. Experiment 4: Data-deficient covariate shifts with multi domain training

Meta-learning domain generalization. As seen in Table 2, the MLDG-based model performs worse on seen validation accuracy, but ends up performing better than the naively trained model that very slightly decrease performance between seen and unseen domains.

Scaling to entire dataset. As seen in Table 3, the MLDG-based model performs better on unseen generalization tasks than the naively trained model in both metrics of F1 score and accuracy. The final model has an F1 score of 0.6899.

Table 3. Final model evaluations on unseen FU Berlin data

Model	Unseen F1 Score	Unseen Accuracy
Naive	0.5973	0.6093
MLDG-based	0.6899	0.6854

4. Discussion

4.1. Experiment 1: Single-domain classification

The basic LeNet-5 trains very poorly. The AlexNet performs better on stability. Notably, when the modernized upgrades between AlexNet and LeNet is factorized, the difference from changing avg pooling to max pooling has the greatest effect on training stability where most models with average pooling remain stuck with no training. Next, the change in activation function is important. Importantly, depth of the network alone does not support the improvements seen between AlexNet and LeNet. This preliminary experiment also shows that improvements to models are factorizable, a large assumption taken in future experiments. Future studies can challenge this assumption.

For augmentation, both vertical and horizontal flips worked well which makes sense with the flip-invariant nature of histology. Interestingly, augmentation of saturation and hue help training accuracy at the cost of validation accuracy. This may reveal that saturation and hue are important to the classification.

Zero padding resulted in best performing models than other padding methods, including image padding that uses the original image data. This may result from the curated dataset that always placed mitotic figures in the center of images. In addition, the task is classification, not the identification or segmentation of figures. Thus, the edge data is not relevant to the model learning, and zero padding is preferred.

RegNet hyperparameter search led to marginal improvements from RegNetX32. Often, parameter increase costs were not worth marginal improvements. Particularly no improvement from increasing the first block depth is significant, because the experiment intentionally violated a network design guideline from prior works [7]. The results of no improvement support the findings.

Estimating the cumulative network distribution over ratios showed that the ratio of 1:2 for stage depths is a good network design constraint for the specific RegNet restrictions used. Less clear results were seen with the first stage depth size, but sizes of 32 and 64 seem to be good network design spaces. A large limitation in both studies is the lack of computational power, limiting the sampling to only 5 networks per constraint. In addition, only the depths could be varied, so the full distribution may not have been captured. Future studies will expand the empirical sampling to more networks that vary more than just the depths of stages while testing the same design constraints of depth ratio and first stage depth size.

4.2. Experiment 2: Data-abundant covariate shifts

Initial cross domain generalization tests show that covariate shifts exist within the MIDOG++ dataset. When models have full access to target distributions and their labels, a combined model can be trained to achieve good performance in the target task. Naively combining different domains can still result in better performance than single-domain training, likely due to dataset size increases. In addition, the observation on early stopping of the combined model demonstrated the usefulness of early stopping as a regularization technique.

However when label access to target distribution is removed, empirical weighted risk minimization became a successful alternative to support model generalization to target distributions. As expected, the models targeting certain distributions performed best when evaluated on that distribution. Unexpectedly, most models that were targeted for a certain distribution still performed better than the baseline combined model when evaluated on distributions that were not the target of the algorithm. This is an unexpected finding that is an interesting future direction to investigate.

4.3. Experiment 3-4: Data-deficient shifts

In experiment 3, when given a single training domain and asked to generalize to unseen target distributions, the options were limited. Investigations on augmentation were promising due to the visualized color differences between scanners and tumors of the dataset, but these augmentations did not improve generalization.

When given more training domains in experiment 4, alternate algorithms like meta-learning domain generalization (MLDG) improved model generalization to unseen domains well. This algorithm corrected covariate shifts when training both LeNet-based and AlexNet-based models. MLDG-based models were observed to perform worse on seen domains, but generalized better to unseen domains than naively trained models. This demonstrates how covariate shift correction can come at the cost of in-domain performance.

A major limitation is lack of a robust benchmark. The final comparison can only show the effect of applying the MLDG algorithm. It is difficult to know if MLDG-based models perform better due to reduced covariate shift, or from serendipitous effects. Another major limitation is that the label shift was not addressed in the FU Berlin target distribution (only 37% positive labels). Simultaneous covariate and label shift correction is a future investigation given more compute and time.

5. References

References

[1] Marc Aubreville, Frauke Wilm, Nikolas Stathonikos, Katharina Breininger, Taryn A. Donovan, Samir Jabari, Mitko Veta, Jonathan Ganz, Jonas Ammeling, Paul J. van Diest, Robert Klopfleisch, and Christof A. Bertram. A comprehensive multi-domain dataset for mitotic figure detection. *Scientific Data*, 10:484, 2023. 1

[2] Google DeepMind. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 1

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 2

[4] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324. IEEE, 1998. 2

[5] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M. Hospedales. Learning to generalize: Meta-learning for domain generalization. 2017. 4

[6] R. Patuzzo et al. The prognostic role of mitotic rate in cutaneous malignant melanoma: Evidence from a multicenter study on behalf of the italian melanoma intergroup. *Cancer*, 129:2331–2340, 2023. 1

[7] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10428–10436, 2020. 3, 6

[8] Kristoffer Stacke, Gabriel Eilertsen, Jonas Unger, and Claes Lundström. Measuring domain shift for deep learning in histopathology. *IEEE Journal of Biomedical and Health Informatics*, 25:325–336, 2020. 1

[9] M. Trojani et al. Soft-tissue sarcomas of adults; study of pathological prognostic variables and definition of a histopathological grading system. *International Journal of Cancer*, 33:37–42, 1984. 1

[10] Aston Zhang, Zachary Lipton, Mu Li, and Alexander Smola. *Dive into Deep Learning*. Cambridge University Press with Amazon Web Services, 2020. Chapter 8.8: Designing Convolution Network Architectures, accessed 2025-12-15. 2, 3