Deep convolutional neural networks have proved effective in segmenting lesions and anatomies in various medical imaging modalities. However, in the presence of small sample size and domain shift problems, these models often produce masks with non-intuitive segmentation mistakes. In this paper, we propose a segmentation framework called ErrorNet, which learns to correct these segmentation mistakes through the repeated process of injecting systematic segmentation errors to a segmentation mask based on a learned shape prior, followed by attempting to predict the injected error. During inference, ErrorNet corrects the segmentation mistakes by adding the predicted error map to the initial segmentation mask. ErrorNet has advantages over alternatives based on domain adaptation or CRF-based post processing, because it requires neither domain-specific parameter tuning nor any data from the target domains. We have evaluated ErrorNet using five public datasets for the task of retinal vessel segmentation. The selected datasets differ in size and patient population, allowing us to evaluate the effectiveness of ErrorNet in handling small sample size and domain shift problems. Our experiments demonstrate that ErrorNet outperforms a base segmentation model, a CRF-based post processing scheme, and a domain adaptation method, with a greater performance gain in the presence of dataset limitations above.

Index Terms—retinal vessel segmentation, limited data, domain shift, error prediction, error correction

1. INTRODUCTION

Medical imaging datasets are often unrepresentative of the patient population, lacking adequate annotated images or otherwise being limited to a particular clinical site or vendor. The former leads to the small sample size problem whereas the latter causes the domain shift problem. In the presence of unrepresentative training datasets, even the most sophisticated architectures may generate non-intuitive segmentation errors such as holes in the segmentations of connected organs or breaks along the segmented vessels. Although acquiring additional annotations strikes as a natural workaround to reduce systematic segmentation errors caused by unrepresentative datasets, it incurs substantial annotation cost. Recent active learning and interactive segmentation methodologies [1, 2, 3] provide cost-effective solutions to expand medical datasets, but they still require highly-trained medical experts in the loop. Unsupervised domain adaptation is an expert-free solution that aims to expand medical datasets by bridging the domain shift between the current training set and the target test sets. However, this approach requires unlabeled data from the target domains, which not only is scarcely available, but also does not scale well to many target domains (e.g., widespread clinical deployment). Post-processing methods based on conditional random fields (CRFs) is another approach to reducing systematic segmentation errors caused by limited datasets. While effective in natural images, application of CRFs in medical images have shown inconclusive results [4]. Furthermore, post-processing with CRFs often requires extensive application-specific parameter tuning. There is a need to develop a methodology that can reduce the systematic errors caused by dataset limitations without requiring experts, additional datasets, or extensive parameter tuning.

In this paper, we propose ErrorNet, a segmentation framework with the capability of reducing segmentation errors caused by domain shifts or limited datasets. ErrorNet consists of a base segmentation network, an error injection network, and a domain adaptation method, with a greater performance gain in the presence of dataset limitations above.

1. INTRODUCTION

Medical imaging datasets are often unrepresentative of the patient population, lacking adequate annotated images or otherwise being limited to a particular clinical site or vendor. The former leads to the small sample size problem whereas the latter causes the domain shift problem. In the presence of unrepresentative training datasets, even the most sophisticated architectures may generate non-intuitive segmentation errors such as holes in the segmentations of connected organs or breaks along the segmented vessels. Although acquiring additional annotations strikes as a natural workaround to reduce systematic segmentation errors caused by unrepresentative datasets, it incurs substantial annotation cost. Recent active learning and interactive segmentation methodologies [1, 2, 3] provide cost-effective solutions to expand medical datasets, but they still require highly-trained medical experts in the loop. Unsupervised domain adaptation is an expert-free solution that aims to expand medical datasets by bridging the domain shift between the current training set and the target test sets. However, this approach requires unlabeled data from the target domains, which not only is scarcely available, but also does not scale well to many target domains (e.g., widespread clinical deployment). Post-processing methods based on conditional random fields (CRFs) is another approach to reducing systematic segmentation errors caused by limited datasets. While effective in natural images, application of CRFs in medical images have shown inconclusive results [4]. Furthermore, post-processing with CRFs often requires extensive application-specific parameter tuning. There is a need to develop a methodology that can reduce the systematic errors caused by dataset limitations without requiring experts, additional datasets, or extensive parameter tuning.

In this paper, we propose ErrorNet, a segmentation framework with the capability of reducing segmentation errors caused by domain shifts or limited datasets. ErrorNet consists of a base segmentation network, an error injection network, and a domain adaptation method, with a greater performance gain in the presence of dataset limitations above.

1. INTRODUCTION

Medical imaging datasets are often unrepresentative of the patient population, lacking adequate annotated images or otherwise being limited to a particular clinical site or vendor. The former leads to the small sample size problem whereas the latter causes the domain shift problem. In the presence of unrepresentative training datasets, even the most sophisticated architectures may generate non-intuitive segmentation errors such as holes in the segmentations of connected organs or breaks along the segmented vessels. Although acquiring additional annotations strikes as a natural workaround to reduce systematic segmentation errors caused by unrepresentative datasets, it incurs substantial annotation cost. Recent active learning and interactive segmentation methodologies [1, 2, 3] provide cost-effective solutions to expand medical datasets, but they still require highly-trained medical experts in the loop. Unsupervised domain adaptation is an expert-free solution that aims to expand medical datasets by bridging the domain shift between the current training set and the target test sets. However, this approach requires unlabeled data from the target domains, which not only is scarcely available, but also does not scale well to many target domains (e.g., widespread clinical deployment). Post-processing methods based on conditional random fields (CRFs) is another approach to reducing systematic segmentation errors caused by limited datasets. While effective in natural images, application of CRFs in medical images have shown inconclusive results [4]. Furthermore, post-processing with CRFs often requires extensive application-specific parameter tuning. There is a need to develop a methodology that can reduce the systematic errors caused by dataset limitations without requiring experts, additional datasets, or extensive parameter tuning.

In this paper, we propose ErrorNet, a segmentation framework with the capability of reducing segmentation errors caused by domain shifts or limited datasets. ErrorNet consists of a base segmentation network, an error injection network, and a domain adaptation method, with a greater performance gain in the presence of dataset limitations above.
training datasets that face larger domain shifts.

**Contributions.** Our contributions include: 1) a novel segmentation framework with an error correction mechanism based on shape prior, 2) extensive evaluation using five public datasets in both same- and cross-dataset evaluation, 3) demonstrated gain over several baselines including CRF-based post processing and a domain adaptation method.

2. RELATED WORK

Due to limited space, we contrast our approach against domain adaptation, post-segmentation refinement, and conditional prior networks, and refer the readers to [4] for a comprehensive survey on methods for medical image segmentation with limited datasets.

**Domain adaptation:** These methods typically require unlabeled [5, 6] or often labeled data [7, 8] from the target domains. Thus, they are hardly applicable when the target domain is unknown a priori or too diverse to have data from (e.g., widespread clinical use). In contrast, ErrorNet does not require data from the target domains as it corrects segmentation errors based on a shape prior learned during the training stage.

**Conditional Prior Networks (CPNs):** CPNs can faithfully learn a shape prior for flows between images [9]. We adopt a similar structure but instead of learning a prior distribution of flows, we use the CPN architecture to learn the prior distribution of segmentation masks. We additionally extend CPNs such that they can generate examples that lie within the learned segmentation prior [10]. We learn the shape prior of segmentation masks so that ErrorNet can leverage the learned segmentation shape prior and train itself to correct imperfect segmentation masks when only limited data is available.

**Post-processing schemes:** Methods based on different variants of CRFs can be used to force connectivity constraints in segmentation results [11, 12]. However, these methods often require extensive parameter tuning and have shown only mixed results for medical image segmentation [4]. In contrast, ErrorNet is end-to-end trainable and application agnostic, eliminating the need for heuristic designs. Denoising autoencoders have also been used as post-processing recently [13], but handcrafted, domain-specific error patterns are required for training. This limitation is overcome in ErrorNet as the error patterns are learned systematically.

3. METHOD

Figure 1 shows the overview of our proposed segmentation framework, ErrorNet, which consists of three consecutive networks: a base segmentation network, an error injection network, and an error prediction network. Given a training image \( I \), the segmentation network generates an initial segmentation mask \( S \), which is then degraded by the error injection network based on a learned shape prior of the vessels, resulting in the degraded mask \( \hat{S} \). The prediction network takes as input the original image stacked with the degraded segmentation mask, and outputs an error map \( \hat{E} \), which attempts to predict the true error, \( E \), between the ground truth and the degraded segmentation map at the pixel-level. During testing, we bypass the error injection network, directly passing the initial segmentation result to the error prediction model. The final segmentation mask is obtained by adding the predicted error map to the initial segmentation, \( S^* = S + \hat{E} \). We explain each individual network as follows:

**Base segmentation network:** We choose the widely used U-Net as the base segmentation network, which is trained by minimizing the cross entropy loss, \( L_{seg} = -\sum_i \log(p_i) \).

**Error injection network:** The task for the error injection network is to generate a degraded segmentation mask, by inject-
ing error patterns to the initial segmentation result. However, it is critical for the error patterns to be representative; otherwise, the subsequent error prediction network would learn an unrelated and perhaps trivial vision task, leading to an ineffective error correction mechanism. Also, the error patterns must be diverse; otherwise, the subsequent error prediction network will overfit to a limited set of error patterns in the training set, particularly when the training set is small. The importance of diverse error patterns is even more pronounced for cross-domain model evaluation where the base segmentation model may produce segmentation maps with error patterns that are partially or largely different from that of the source training dataset. The choice of error injection network is thus critical to the success of the suggested framework.

For this purpose, we use a variational autoencoder (VAE), which is trained by minimizing \( L_{\text{vae}} = \sum_i (\hat{S}^{(i)} - S^{(i)})^2 + KL(P_{\theta}(z^{(i)}|S^{(i)}))|N(0, 1)) \) where the first term constrains the degraded mask to be similar to the initial segmentation result and the second term constrains the latent space of the VAE to follow a standard normal distribution. A VAE generates representative and diverse error patterns, because, during training it learns a distribution over segmentation maps in a high dimensional feature space. By sampling from this distribution, VAE can generate diverse variants of a given segmentation map, which can represent the manifold around the input segmentation map. This degradation mechanism introduces stochasticity in the model, increasing the input space of imperfect segmentations, enabling us to train a more robust error prediction network.

**Error prediction network:** We use a shallow U-Net for the task of error prediction. Specifically, the error prediction network takes as input the degraded segmentation map stacked with the original image, and returns a k-channel error prediction map at the original image resolution, \( \hat{E}^{(i)} \), where \( k \) is the number of classes. We use a \( \tanh \) activation function in the output layer of the error prediction model. A negative value in the predicted error map corresponds to decreasing the probability of the class being predicted whereas a positive value corresponds to increasing the probability. We train the error prediction network by minimizing \( L_{\text{pred}} = \sum_i (E^{(i)} - \hat{E}^{(i)}) \) where \( E^{(i)} = (\hat{S}^{(i)} - G^{(i)})^2 \) with \( G^{(i)} \) being the ground truth mask for the \( i^{th} \) image in the batch.

### 4. EXPERIMENTS

**Architecture details:** We use a U-Net as the base segmentation model. The U-Net consists of 4 downsampling blocks in the encoder and 4 upsampling blocks in the decoder. All convolutions layers use 3x3 kernels. We have followed the best practices as suggested in [14] for configuring and training our U-Net architecture. Specifically, we have used instance normalization in downsampling blocks and leaky Relu as activation functions, and further excluded large black background regions while normalizing the training images, to name a few.

For the error injection module, we use a VAE with 3 downsampling blocks, a 6400 dimensional latent feature space, and 3 upsampling blocks. To ensure that the injected error patterns do not transform the segmentation mask completely, we sample from the latent space with a variance of 0.0001. The error prediction module follows a U-Net architecture with 3 downsampling blocks followed by 3 upsampling blocks with skip connections. Both VAE and error prediction network use batch normalization and relu activation functions. The number of kernels in the VAE and error prediction network were optimized so that a GPU with 12 GB RAM can hold the entire ErrorNet in memory. We refer the readers to the supplementary material for architecture details.

**Training Details:** While ErrorNet can be trained end-to-end, we have found that stage-wise training facilitates convergence. Specifically, we first train the base segmentation by minimizing the segmentation loss function, \( L = L_{\text{seg}} \). We then freeze the weights of the base segmentation network and train error injection network by minimizing the VAE loss, \( L = L_{\text{vae}} \). Once VAE is trained, we train the error prediction network by minimizing \( L = L_{\text{pred}} \), while freezing the weights of the base segmentation and the error injection networks. We refer to this training scheme as stage-wise training henceforth. The ErrorNet can now be trained jointly in an end-to-end fashion, while freezing the weights of the error injection module, effectively minimizing \( L = L_{\text{pred}} + L_{\text{seg}} \). We refer to this training scheme as Joint Tr in Table 2.

**Datasets:** Table 1 summarizes the 5 datasets used to evaluate ErrorNet for the task of retinal vessel segmentation. The selected datasets vary in terms of size, population, and acquisition machine, allowing us to evaluate the effectiveness of ErrorNet under different sample sizes and domain shift.

**Performance baselines:** We compare ErrorNet against 1) a U-Net carefully-optimized according to the best practices suggested in [14], the same U-Net with CRF-based post-processing, a recent unsupervised domain adaptation method [16], and V-GAN [15], which is a modern retinal vessel segmentation network trained in an adversarial manner.

| Dataset | # images | Data splits |
|---------|----------|-------------|
|         | Total | Diseased | Healthy | Train | Val | Test |
| DRIVE   | 40    | 7        | 33      | 18    | 2   | 20   |
| STARE   | 20    | 10       | 10      | 10    | 2   | 8    |
| CHASE   | 20    | 0        | 20      | 17    | 5   | 6    |
| ARIA    | 143   | 82       | 61      | 121   | 5   | 17   |
| HRF     | 45    | 30       | 15      | 26    | 5   | 14   |

Table 1: Datasets used in our experiments
also compare ErrorNet with and without joint training.

**Evaluation scenarios:** We evaluate ErrorNet in the presence of small sample size and domain shift problems. To study the small size, we train ErrorNet using Chase, which is a small dataset, and Aria, which is the largest dataset under study. To study the domain shift problem, we evaluate the models above on the domains other than the one they are trained on.

**Results:** We have summarized the results of both evaluation scenarios in Table 2. When Chase dataset is used for training and testing, ErrorNet achieves a Dice of 81.5 outperforming all performance baselines. The ablation study also shows that ErrorNet with joint training achieves a 1-point increase in Dice. Inclusion of VAE, on the other hand, shows no significant performance gain. This is because the training and test domains are the same (Chase). In the cross-domain evaluation, ErrorNet and in particular the VAE module achieve greater performance gains over the baselines. Specifically, ErrorNet achieves an average Dice score of 71.0, outperforming the second best [15] and third best method [16] by 2.4 and 2.5 points, respectively. The VAE module also enables a 1.6-point increase in Dice. The widened performance gap is due to the domain shift caused by different patient population and pathologies present in the datasets. While Chase contains only healthy fundus images of children’s eyes with a central vessel reflex, all the other datasets used for testing contain pathological cases from adult populations. We hypothesize that ErrorNet can effectively learn the general structure of eye vessels; and thus, it can help correct mis-segmentations introduced by dataset limitations.

ErrorNet trained with Aria continues to improve the segmentation performance over the baseline models, with both VAE and joint training features showing consistent performance gains. However, the superiority over baselines is not as drastic as in the case where ErrorNet was trained with Chase. Intuitively, these results make sense. Aria is a larger, more varied dataset with images from both healthy and diseased patients; and thus, the models trained on Aria generalize better to other datasets. As a result, the improvements made by the error correction module are smaller.

**5. CONCLUSION**

We presented ErrorNet, a framework for systematic handling of segmentation errors caused by limited datasets. We evaluated ErrorNet using 5 public datasets for the task of retinal vessel segmentation. Our results demonstrated the effectiveness of ErrorNet in both same-dataset and cross-dataset evaluations, particularly when the size of training set was small and domain shift was large. Our future work would focus on evaluating ErrorNet on other medical image segmentation tasks as well as evaluating the effectiveness of ErrorNet for the task of active learning.

| Architecture          | Error Pred | VAE | Train on | Test on | CHASE | DRIVE | ARIA | STARE | HRF | Avg. | CHASE | DRIVE | ARIA | STARE | HRF | Avg. |
|-----------------------|------------|-----|----------|---------|-------|-------|------|-------|-----|------|-------|-------|------|-------|-----|------|
| ErrorNet w/ ablation  | ✔          | ✔   |          |         |       |       |      |       |     | 80.2 | 68.6  | 60.7  | 60.2  | 62.2  | 66.4 | 76.8 | 72.1 |
| ErrorNet w/ ablation  | ✔   ✔      | ✔   |          |         |       |       |      |       |     | 80.1 | 71.8  | 61.1  | 59.8  | 67.2  | 68.0 | 76.2 | 71.3 |
| ErrorNet w/ ablation  | ✔   ✔      | ✔   |          |         |       |       |      |       |     | 81.5 | 73.2  | 66.5  | 65.2  | 68.6  | 71.0 | 76.7 | 78.9 |
| DA-ADV [16]          |            |     |          |         |       |       |      |       |     | 72.3 | 69.3  | 68.2  | 64.7  | 67.4  | 68.4 | 71.5 | 72.9 |
| V-GAN[15]            |            |     |          |         |       |       |      |       |     | 79.7 | 71.5  | 64.2  | 61.0  | 66.6  | 68.5 | 68.7 | 73.3 |
| U-Net [14] + CRF     |            |     |          |         |       |       |      |       |     | 81.2 | 65.4  | 62.6  | 56.4  | 63.6  | 65.8 | 78.4 | 69.5 |
| U-Net [14]           |            |     |          |         |       |       |      |       |     | 79.3 | 67.6  | 60.3  | 59.5  | 61.3  | 65.6 | 76.7 | 77.3 |
| Avg.                 |            |     |          |         |       |       |      |       |     | 76.8  | 72.0  | 72.8  | 72.3  | 73.9  |      |      |      |

Table 2: Comparison between ErrorNet and other performance baselines. Dice is used for comparison. Grey columns indicate same-domain evaluation whereas the other columns contain the results for cross-domain evaluation. ErrorNet outperforms the competing baselines on average, but the performance gap is wider in the presence of domain shift (cross-domain evaluation) and small sample size (the small Chase dataset used for training). Ablation studies show that VAE and joint-training are effective in improving the performance of ErrorNet.

Fig. 2: Effectiveness of ErrorNet in bridging breaks along segmented vessels. Each row compares the segmentation results before (left) and after (right) error correction. The yellow boxes indicate regions where the ErrorNet has connected fragmented vessels or sharpened vessel structures. To respect the space limit, only cropped images are shown. Full-image results are available in the appendix.

Qualitative comparison: Figure 2 compares the segmentation results before and after error correction by the error prediction network. Recall that the error injection network is not used during inference—segmentation results are directly sent to the error prediction network for error correction. As highlighted by the yellow boxes, ErrorNet has connected fragmented vessels or sharpened vessel structures.

5. CONCLUSION

We presented ErrorNet, a framework for systematic handling of segmentation errors caused by limited datasets. We evaluated ErrorNet using 5 public datasets for the task of retinal vessel segmentation. Our results demonstrated the effectiveness of ErrorNet in both same-dataset and cross-dataset evaluations, particularly when the size of training set was small and domain shift was large. Our future work would focus on evaluating ErrorNet on other medical image segmentation tasks as well as evaluating the effectiveness of ErrorNet for the task of active learning.
6. REFERENCES

[1] Jamshid Sourati, Ali Gholipour, Jennifer G Dy, Xavier Tomas-Fernandez, Sila Kurugol, and Simon K Warfield, “Intelligent labeling based on fisher information for medical image segmentation using deep learning,” IEEE transactions on medical imaging, 2019.

[2] Weicheng Kuo, Christian Häne, Esther Yuh, Pratik Mukherjee, and Jitendra Malik, “Cost-sensitive active learning for intracranial hemorrhage detection,” in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2018, pp. 715–723.

[3] Tomas Sakinis, Fausto Milletari, Holger Roth, Panagiotis Koriatis, Petro Kostandy, Kenneth Philbrick, Zeynettin Akkus, Ziyue Xu, Daguang Xu, and Bradley J Erickson, “Interactive segmentation of medical images through fully convolutional neural networks,” arXiv preprint arXiv:1903.08205, 2019.

[4] Nima Tajbakhsh, Laura Jeyaseelan, Qian Li, Jeffrey Chiang, Zhihao Wu, and Xiaowei Ding, “Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation,” arXiv preprint arXiv:1908.10454, 2019.

[5] Yuankai Huo, Zhoubing Xu, Hyeonsoo Moon, Shuming Bao, Albert Assad, Tamara K Moyo, Michael R Savona, Richard G Abramson, and Bennett A Landman, “Synseg-net: Synthetic segmentation without target modality ground truth,” IEEE transactions on medical imaging, vol. 38, no. 4, pp. 1016–1025, 2018.

[6] Cheng Chen, Qi Dou, Hao Chen, Jing Qin, and Pheng-Ann Heng, “Synergistic image and feature adaptation: Towards cross-modality domain adaptation for medical image segmentation,” arXiv preprint arXiv:1901.08211, 2019.

[7] Qi Dou, Cheng Ouyang, Cheng Chen, Hao Chen, Ben Glocker, Xiaihai Zhuang, and Pheng-Ann Heng, “Pnp-adanet: Plug-and-play adversarial domain adaptation network with a benchmark at cross-modality cardiac segmentation,” arXiv preprint arXiv:1812.07907, 2018.

[8] Zizhao Zhang, Lin Yang, and Yefeng Zheng, “Translating and segmenting multimodal medical volumes with cycle-and shape-consistency generative adversarial network,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9242–9251.

[9] Yanchao Yang, Alex Wong, and Stefano Soatto, “Dense depth posterior (dpd) from single image and sparse range,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3353–3362.

[10] Diederik P Kingma and Max Welling, “Auto-encoding variational bayes,” arXiv preprint arXiv:1312.6114, 2013.

[11] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker, “Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation,” Medical image analysis, vol. 36, pp. 61–78, 2017.

[12] Christian Wachinger, Martin Reuter, and Tassilo Klein, “Deepnat: Deep convolutional neural network for segmenting neuroanatomy,” NeuroImage, vol. 170, pp. 434–445, 2018.

[13] Agostina J Larrazabal, Cesar Martinez, and Enzo Ferrante, “Anatomical priors for image segmentation via post-processing with denoising autoencoders,” To appear in MICCAI 2019.

[14] Fabian Isensee, Philipp Kickingereder, Wolfgang Wick, Martin Bendszus, and Klaus H Maier-Hein, “No new-net,” in International MICCAI Brainlesion Workshop. Springer, 2018, pp. 234–244.

[15] Jaemin Son, Sang Jun Park, and Kyu-Hwan Jung, “Retinal vessel segmentation in fundoscopic images with generative adversarial networks,” arXiv preprint arXiv:1706.09318, 2017.

[16] Nanqing Dong, Michael Kampffmeyer, Xiaodan Liang, Zeya Wang, Wei Dai, and Eric Xing, “Unsupervised domain adaptation for automatic estimation of cardiothoracic ratio,” in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2018, pp. 544–552.
Appendix

This appendix consists of 6 figures and 3 tables. The figures serve to illustrate the error correction mechanisms of ErrorNet in high resolution uncropped images. Due to limited space, we showed only low resolution cropped results in the main text. The tables include architecture details for the base segmentation network, error injection network, and error prediction network. The readers are welcome to contact Nima Tajbakhsh at ntajbakhsh@voxelcloud.io for further clarification on our method, results, or architecture details.
Table A.1: Architecture details for the Segmentation Network. All convolution layers use 3x3 kernels. As suggested by [14], instance normalization and leaky-relu activation functions are used.

| Name              | Feature maps (input) | Feature maps (output) |
|-------------------|----------------------|-----------------------|
| **Encoder pathway** |                      |                       |
| Conv layer - 1a   | 640 x 640 x 1        | 640 x 640 x 32        |
| Conv layer - 1b   | 640 x 640 x 32       | 640 x 640 x 32        |
| Max pool - 1      | 640 x 640 x 32       | 320 x 320 x 32        |
| Conv layer - 2a   | 320 x 320 x 32       | 320 x 320 x 64        |
| Conv layer - 2b   | 320 x 320 x 64       | 320 x 320 x 64        |
| Max pool - 2      | 320 x 320 x 64       | 160 x 160 x 64        |
| Conv layer - 3a   | 160 x 160 x 64       | 160 x 160 x 128       |
| Conv layer - 3b   | 160 x 160 x 128      | 160 x 160 x 128       |
| Max pool - 3      | 160 x 160 x 128      | 80 x 80 x 128         |
| Conv layer - 4a   | 80 x 80 x 128        | 80 x 80 x 256         |
| Conv layer - 4b   | 80 x 80 x 256        | 80 x 80 x 256         |
| Max pool - 4      | 80 x 80 x 256        | 40 x 40 x 256         |
| Conv layer - 5a   | 40 x 40 x 256        | 40 x 40 x 512         |
| Conv layer - 5b   | 40 x 40 x 512        | 40 x 40 x 512         |
| **Decoder Pathway** |                    |                       |
| Upsample - 1      | 40 x 40 x 512        | 80 x 80 x 512         |
| Concat - 1        | 80 x 80 x 512 (up sample - 1) | 80 x 80 x 768 |
|                    | 80 x 80 x 256 (conv - 4b) |                       |
| Conv layer - 6a   | 80 x 80 x 768        | 80 x 80 x 256         |
| Conv layer - 6b   | 80 x 80 x 256        | 80 x 80 x 256         |
| Upsample - 2      | 80 x 80 x 256        | 160 x 160 x 256       |
| Concat - 2        | 160 x 160 x 256 (upsample - 2 ) | 160 x 160 x 384 |
|                    | 160 x 160 x 128 (conv - 3b) |                       |
| Conv layer - 7a   | 160 x 160 x 384      | 160 x 160 x 128       |
| Conv layer - 7b   | 160 x 160 x 128      | 160 x 160 x 128       |
| Upsample - 3      | 160 x 160 x 128      | 320 x 320 x 128       |
| Concat - 3        | 320 x 320 x 128 (upsample - 3) | 320 x 320 x 192 |
|                    | 320 x 320 x 64 (conv - 2b) |                       |
| Conv layer - 8a   | 320 x 320 x 192      | 320 x 320 x 64        |
| Conv layer - 8b   | 320 x 320 x 64       | 320 x 320 x 64        |
| Upsample - 4      | 320 x 320 x 64       | 640 x 640 x 64        |
| Concat - 4        | 640 x 640 x 64 (upsample - 4) | 640 x 640 x 96 |
|                    | 640 x 640 x 32 (conv - 1b) |                       |
| Conv layer - 9a   | 640 x 640 x 96       | 640 x 640 x 32        |
| Conv layer - 9b   | 640 x 640 x 32       | 640 x 640 x 32        |
| Output layer      | 640 x 640 x 32       | 640 x 640 x 1         |
Table A.2: Architecture details for the Error Injection Network. All convolution layers use 3x3 kernels. Batch normalization and relu activation functions are used throughout the network.

| Name                      | Feature maps (input) | Feature maps (output) |
|---------------------------|----------------------|-----------------------|
| **Encoder Pathway**       |                      |                       |
| Conv layer - 1a           | 640 x 640 x 1        | 640 x 640 x 32        |
| Conv layer - 1b           | 640 x 640 x 32       | 640 x 640 x 32        |
| Max pool - 1              | 640 x 640 x 32       | 320 x 320 x 32        |
| Conv layer - 2a           | 320 x 320 x 32       | 320 x 320 x 64        |
| Conv layer - 2b           | 320 x 320 x 64       | 320 x 320 x 64        |
| Max pool - 2              | 320 x 320 x 64       | 160 x 160 x 64        |
| Conv layer - 3a           | 160 x 160 x 64       | 160 x 160 x 128       |
| Conv layer - 3b           | 160 x 160 x 128      | 160 x 160 x 128       |
| Max pool - 3              | 160 x 160 x 128      | 80 x 80 x 128         |
| encoder conv - 4a         | 80 x 80 x 128        | 80 x 80 x 512         |
| encoder conv - 4b         | 80 x 80 x 512        | 80 x 80 x 1           |
| encoder dense - mu        | 80 x 80 x 1          | 6400                  |
| encoder dense - sigma     | 80 x 80 x 1          | 6400                  |
| **VAE latent space sampling** |                      |                       |
| sampling - 1              | 6400 (encoder dense - mu) | 6400                  |
| reshape - 1               | 6400 (encoder dense - sigma) | 6400                  |
|                          |                      |                       |
| **Decoder Pathway**       |                      |                       |
| Conv transpose - 1        | 80 x 80 x 1          | 160 x 160 x 1        |
| Conv layer - 5a           | 160 x 160 x 64       | 160 x 160 x 64       |
| Conv layer - 5b           | 160 x 160 x 64       | 160 x 160 x 64       |
| Conv transpose - 2        | 160 x 160 x 64       | 320 x 320 x 32       |
| Conv layer - 6a           | 320 x 320 x 32       | 320 x 320 x 32       |
| Conv layer - 6b           | 320 x 320 x 32       | 320 x 320 x 32       |
| Upsample - 3              | 320 x 320 x 32       | 640 x 640 x 32       |
| Conv layer - 7a           | 640 x 640 x 32       | 640 x 640 x 32       |
| Conv layer - 7b           | 640 x 640 x 32       | 640 x 640 x 32       |
| Output layer              | 640 x 640 x 32       | 640 x 640 x 2        |
| Sigmoid layer             | 640 x 640 x 2        | 640 x 640 x 2        |
Table A.3: Architecture details for the Error Prediction Network. All convolution layers use 3x3 kernels. Batch normalization and relu activation functions are used throughout the network.

| Name                | Feature maps (input)                                      | Feature maps (output) |
|---------------------|----------------------------------------------------------|-----------------------|
| Encoder Pathway     |                                                          |                       |
| Concat - input      | 640 x 640 x 1 (input image)                             | 640 x 640 x 2         |
|                     | 640 x 640 x 1 (degraded segmentation)                   |                       |
| Conv layer - 1a     | 640 x 640 x 2                                          | 640 x 640 x 32        |
| Conv layer - 1b     | 640 x 640 x 32                                          | 640 x 640 x 32        |
| Max pool - 1        | 640 x 640 x 32                                          | 320 x 320 x 32        |
| Conv layer - 2a     | 320 x 320 x 32                                          | 320 x 320 x 64        |
| Conv layer - 2b     | 320 x 320 x 64                                          | 320 x 320 x 64        |
| Max pool - 2        | 320 x 320 x 64                                          | 160 x 160 x 64        |
| Conv layer - 3a     | 160 x 160 x 64                                          | 160 x 160 x 128       |
| Conv layer - 3b     | 160 x 160 x 128                                         | 160 x 160 x 128       |
| Max pool - 3        | 160 x 160 x 128                                         | 80 x 80 x 128         |
| Conv layer - 4a     | 80 x 80 x 128                                           | 80 x 80 x 256         |
| Conv layer - 4b     | 80 x 80 x 256                                           | 80 x 80 x 256         |
| Decoder Pathway     |                                                          |                       |
| Upsample - 2        | 80 x 80 x 256                                           | 160 x 160 x 256       |
| Concat - 2          | 160 x 160 x 256 (upsample - 2)                          | 160 x 160 x 384       |
|                     | 160 x 160 x 128 (conv - 3b)                            |                       |
| Conv layer - 7a     | 160 x 160 x 384                                         | 160 x 160 x 128       |
| Conv layer - 7b     | 160 x 160 x 128                                         | 160 x 160 x 128       |
| Upsample - 3        | 160 x 160 x 128                                         | 320 x 320 x 128       |
|                     | 320 x 320 x 128 (upsample - 3)                          | 320 x 320 x 192       |
|                     | 320 x 320 x 64 (conv - 2b)                             |                       |
| Concat - 3          | 320 x 320 x 128 (upsample - 3)                          | 320 x 320 x 192       |
| Conv layer - 8a     | 320 x 320 x 192                                         | 320 x 320 x 64        |
| Conv layer - 8b     | 320 x 320 x 64                                          | 320 x 320 x 64        |
| Upsample - 4        | 320 x 320 x 64                                          | 640 x 640 x 64        |
|                     | 640 x 640 x 64 (upsample - 4)                           | 640 x 640 x 96        |
|                     | 640 x 640 x 32 (conv - 1b)                             |                       |
| Concat - 4          | 640 x 640 x 64 (upsample - 4)                           | 640 x 640 x 96        |
| Conv layer - 9a     | 640 x 640 x 96                                          | 640 x 640 x 32        |
| Conv layer - 9b     | 640 x 640 x 32                                          | 640 x 640 x 32        |
| Output layer        | 640 x 640 x 32                                          | 640 x 640 x 1         |
Fig. A.1: [Chase $\rightarrow$ HRF] Effectiveness of ErrorNet for cross-dataset evaluation, where the training set comes from the Chase dataset but the test set comes from the HRF dataset. Top: Fundus image. Bottom-Left: Segmentation result for an HRF dataset image from the segmentation network (before error correction). Bottom-Right: corresponding segmentation result generated by ErrorNet after error correction. The yellow boxes indicate example regions where the ErrorNet model has connected fragmented vessels or sharpened vessel structures.
Fig. A.2: [Chase $\rightarrow$ Drive] Effectiveness of ErrorNet for cross-dataset evaluation, where the training set comes from the Chase dataset but the test set comes from the Drive dataset. Top: Fundus image. Bottom-Left: Segmentation result for a Drive dataset image from the segmentation network (before error correction). Bottom-Right: corresponding segmentation result generated by ErrorNet after error correction. The yellow boxes indicate example regions where the ErrorNet model has connected fragmented vessels or sharpened vessel structures.
Fig. A.3: \([\text{Chase} \rightarrow \text{Chase}]\) Effectiveness of ErrorNet for same-dataset evaluation, where the training and test sets both come from the \textit{Chase} dataset. Top: Fundus image. Bottom-Left: Segmentation result for a \textit{Chase} dataset image from the segmentation network (before error correction). Bottom-Right: corresponding segmentation result generated by ErrorNet after error correction. The yellow boxes indicate example regions where the ErrorNet model has connected fragmented vessels or sharpened vessel structures. As expected, improvement is not as drastic as that of cross-dataset evaluation.
Fig. A.4: [Aria → HRF] Effectiveness of ErrorNet for cross-dataset evaluation, where the training set comes from the Aria dataset but the test set comes from the HRF dataset. Top: Fundus image. Bottom-Left: Segmentation result for an HRF dataset image from the segmentation network (before error correction). Bottom-Right: corresponding segmentation result generated by ErrorNet after error correction. The yellow boxes indicate example regions where the ErrorNet model has connected fragmented vessels or sharpened vessel structures.
Fig. A.5: [Aria → Stare] Effectiveness of ErrorNet for cross-dataset evaluation, where the training set comes from the *Aria* dataset but the test set comes from the *Stare* dataset. Top: Fundus image. Bottom-Left: Segmentation result for a *Stare* dataset image from the segmentation network (before error correction). Bottom-Right: corresponding segmentation result generated by ErrorNet after error correction. The yellow boxes indicate example regions where the ErrorNet model has connected fragmented vessels or sharpened vessel structures.
Fig. A.6: [Aria \rightarrow Drive] Effectiveness of ErrorNet for cross-dataset evaluation, where the training set comes from the Aria dataset but the test set comes from the Drive dataset. Top: Fundus image. Bottom-Left: Segmentation result for a Drive dataset image from the segmentation network (before error correction). Bottom-Right: corresponding segmentation result generated by ErrorNet after error correction. The yellow boxes indicate example regions where the ErrorNet model has connected fragmented vessels or sharpened vessel structures.