An Exploration in Eosinophil Segmentation with Computer Vision

Kevin Lin
School of Data Science
University of Virginia
Charlottesville, VA
pex7ps@virginia.edu

Jason Wang
School of Data Science
University of Virginia
Charlottesville, VA
jyw5hw@virginia.edu

Judy Fox
School of Data Science
University of Virginia
Charlottesville, VA
ckw9mp@virginia.edu

Donald Brown
School of Data Science
University of Virginia
Charlottesville, VA
deb@virginia.edu

***Abstract*—Eosinophilic esophagitis (EoE) severely impacts the lives of patients while being costly to detect through traditional methods. Our approach focuses on This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—image segmentation, medical imaging, computer vision**

I. Introduction

Eosinophilic esophagitis (EoE) is a chronic disease characterized by the prevalence of a type of white blood cell (eosinophil) in the esophagus. EoE affects approximately 0.5-1.0 in 1,000 people and can be seen in 2-7% of patients that undergo endoscopy for any reason [1]. EoE is believed to be triggered by dietary components in patients and is increasing in prevalence [2]. Clinical symptoms include swallowing difficulties, food impaction, and chest pain [3]. For diagnosis, patients presenting symptoms related to EoE undergo an endoscopy where the collected biopsy tissue samples are then evaluated for concentration of eosinophils. The accepted criterion for pathologists to diagnose EoE involves identifying at least one High-Power Field (HPF; 400× magnification adjustment) within a patient's tissue biopsy slide that contains 15 or more eosinophils [4]. The dataset is obtained from the Gastroenterology Data Science Lab from UVA Hospital patient data. A sample image is given in Figure 1. Each image is 512x512x3 large and there are 514 images/masks in the dataset spanning 30 UVA Medical Center patients. We will keep the three channels [r,g,b] for the image but will import the masks as grayscale. All data is obtained from subjects under conditions of academic use only. No personal health information (PHI) is present in the data. On the subject of eosinophil detection, we can use the cross-domain adaptation method proposed in [5] by subsetting the dataset and introducing it incrementally to the model (e.g. Slides from patients that had 0-4 eosinophils, 5-9 eosinophils, 10-14 eosinophils, 15+ eosinophils). With this division we would also get not only a binary answer on whether or not the patient has EoE but also if the patient could be a potential borderline case (10-14 eosinophils). Additionally, there would be further information about the distribution of the number of eosinophils per slide and give the model a chance to learn what

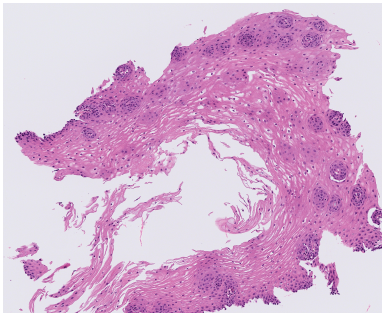

Fig. 1. Example Image Data from Gastroenterology Data Science Lab

the data presents at the boundary conditions. This results can provide a measure of uncertainty through respective confusion matrices for each subset of the data in addition to one for the full dataset. If we notice severe class imbalances, we can use the F1 score as our evaluation metric since it combines both precision and recall. To show this directly, let true positives be TP, false positives be FP, and false negatives be FN. Then we have,

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

Equation 1 illustrates that precision is a measure of how many true positives there are out of all samples that were classified as positive.

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{2}$$

Equation 2 illustrates that recall is a measure of how many true positives were found (recalled) out of all the samples that were actually positives.

F1 is defined to be the harmonic mean of Precision and Recall. Simplifying yields,

$$\text{F1} = \frac{2(\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}. \tag{3}$$

In the case of a large class imbalance, the "averaging" in Equation 3 of the Precision and Recall means that only a high Precision and Recall will produce a high F1 score. If either are low, the F1 score will drop significantly. Furthermore, the few

shot learning approach would directly address the relatively small amounts of labelled data. Although there is no guarantee of obtaining as good results as Hassan, et al., promise has been shown that the method at least improves on conventional domain adaptation techniques. However, the main approach for EoE is that of a segmentation problem and not a classification problem. This work with the cross-domain adaptation method would provide insight into whether there is or is not EoEs in a given image versus a segmentation algorithm that would be able to label pixels as eosinophils or other cells or just empty space. We could approach a segmentation model with a similar methodology by trying to subset the data and feeding it periodically to the model.

For measures of uncertainty, we can use a modified Monte Carlo Dropout approach using this cross-domain adaptation. We will treat the data as discrete since the image is made up of a three-dimensional tuple [r, g, b]. Since we have a very large dataset, we will use variational approximation, specifically minimizing KL divergence, in order to approximate the distribution of the images.

Since maximizing the evidence lower bound (ELBO) is mainly impractical, we will be using a Monte-Carlo approximation. We can use Monte Carlo (MC) Dropout as a Bayesian approximation to identify which image segments correspond to different cells. In Bayesian neural networks, each weight is represented by a probability distribution, in which we will assume Gaussian instead of a number. The learning aspect corresponds to Bayesian inference which we will use MC Sampling. We will measure uncertainty for every pixel using cross-entropy over two classes of "background" and "foreground".

Work from Gal and Ghahramani in 2016 [6] suggests that a Monte Carlo Dropout U-Net is equivalent to the deep Gaussian process used in Bayesian Neural Networks. Essentially, we can minimize the KL divergence using approximation through Monte Carlo integration to get an unbiased estimate. For the Monte Carlo Dropout UNet, we apply dropout before every weight where dropout is defined as switching off neurons at each training step

II. Methodology

A. Prior Work

The U-Net is convolutional network architecture for fast and precise segmentation of images [7]. It has been shown to outperform what was previously considered the best method (a sliding-window convolutional network) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks. For the UNet model, we used 23 convolutional layers with batch normalization. In both the encoder and decoder steps, we used a ReLU activation function, and for the final layer we used a sigmoid activation function. Loss was computed using binary cross entropy and ADAM was used as the optimizer. To prevent overfitting, we implemented early stopping and data augmentation.

The underlying architecture of the U-Net model is shown in Figure 2. At first, the encoder is used to obtain and normalize

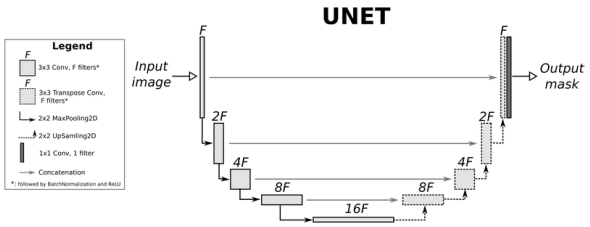

Fig. 2. UNet Architecture

Test Set Dice Coefficient Statistics				
Model	***Size***	***Median***	***Min***	***Max***
U-Net	4.9M	0.628	0.594	0.685
U-Net	8.6M	0.660	**0.632**	0.698
U-Net	10.9M	**0.665**	0.600	**0.701**
Res. U-Net	2.7M	0.632	0.588	0.697
Res. U-Net	4.7M	0.656	0.609	0.696
Res. U-Net	7.4M	0.557	0.541	0.645
R2U-Net	3.4M	0.634	0.606	0.686
R2U-Net	6.0M	0.614	0.530	0.647
R2U-Net	9.4M	0.631	0.572	0.666
Attn. U-Net	3.1M	0.517	0.439	0.627
Attn. U-Net	4.5M	0.529	0.465	0.586

Fig. 3. Adorno et.al. Dice Values [8]

the transformation of the input volume, using a Leaky ReLU activation function at each layer. At the bottleneck of this architecture, the volume will be in the size of 2×2×2 which represents the reduction of dimensionality prior to using a sigmoid activation function for segmentation. The decoder then up-samples this transformed 2×2×2 volume to reconstruct the image with this segmentation.

Once again, our dataset is 514 images from the GI Data Science Lab and sized 5112x512x3. To improve upon Adorno, et. al. [8], we augment the data through flipping and rotation for each image. This means that for each input image, we have created a flipped and rotated image as well, effectively tripling our input dataset. Data augmentation here makes the model generalize better due to the larger amount of training data. For reference, the results from Adorno, et. al. [8] are shown in Figure 3.

The size field in Figure 3 refers to the total number of parameters of each model. For comparison, our UNet results are shown in Table I.

Model	***Size***	***Median***	***Min***	***Max***
U-Net	494K	0.689	0.622	0.794

TABLE I
Test Results Dice Score

Most significantly, we can see that our approach has at least one order of magnitude less in size which means our model is less complex. Given that the same U-Net approach, we can assume, holding all other factors such as GPU availability constant, our approach runs faster and more efficiently. Even at their most complicated model, our model performance

easily outperforms Adorno et. al.'s. As their model outperformed other approaches such as Residual U-Net, R2U-Net, and Attention U-Net, our approach also outperforms these approaches.

However, it is important to note that we devoted a significant amount of time to tune parameters. Additionally, we used binary cross entropy as our loss during training which is not the ideal metric when dealing with medical segmentation data. These results further the statement that "suboptimal approaches with good parameter estimates are better than optimal approaches with bad parameter estimates" [9]. Further work here would be to try to show that these results generalize to other medical imaging datasets and to test to see if using dice loss as our training evaluation metric would improve performance.

B. Computer Vision Approaches

When preforming image segmentation tasks in medical imaging, one of the biggest challenges is not only to achieve high accuracy but also obtaining faster inference. This is often achieved by lightening the model by the number of layers or parameters. The previously discussed U-Net architecture is comprised of many 2D convolutional layers, where a 2D convolution is applied over an input signal that is composed of many input planes. Current research has identified many other convolution mechanisms that can improve performance as well as decrease training time in U-Nets.

1) Dilated Convolutional Layers: Dilated convolution is a technique that adapts the traditional convolution kernel by inserting spaces between consecutive elements. The dilation factor (l) determines how much space is put between the consecutive elements. The purpose is to expand the receptive field of the U-Net without having to increase the computational cost. In fact, with dilated convolutions the computational cost should stay the exact same as the number of parameters stays the same as well. Wang et al. was able to propose a novel U-Net variant containing dilated convolutions where instead of two standard 2d convolutions, the authors used on standard convolution followed by multiple dilated convolutions [10]. The results showed the modification outperformed the vanilla U-Net while using fewer parameters and therefore increased training efficiency.

2) Separable Convolutional layers: Separable convolution layers are similar to dilated in that they try and reduce the number of parameters and computational cost. Separable layers instead try and divide a single convolution into two or more convolutions that produce the same output. This reduces the number of individual multiplications and should improve training efficiency. There are two types of separable convolutions: spatially separable convolutions and depth-wise separable convolutions. Spatially separable convolutions operates on the image width and height while depth-wise separable convolutions operates on the depth dimension which is common in images as they contain an RGB channel. Gadosey et al. was able to demonstrate the advantages of using separable convolution in the U-Net architecture [11]. By using depth-wise convolutions in the entire network, they were able to make a lightweight CNN that was a smaller model, had fewer parameters, and therefore had faster inference time. This proposed SD-UNet was able to achieve comparable or even sometimes better results than vanilla U-Nets and current state of the art.

III. Results

A. Dilated Convolution Layers

To investigate the effect of dilated convolution layers on the performance and efficiency of U-Nets we took the previous architecture discussed above and modified each convolution 2d layer into a dilated convolution layer. This just impacts the encoder part of the U-Net as the decoder still needs to have transposed convolution layers. Next we trained U-Nets at various different levels of dilation factors. Previous research suggested that we keep the dilation factor between 2-16, which led us to select four factors [2,4,8,16]. Zhang et al. explored the concept of pyramid dilated convolution where the dilation factor is changed inside an encoder block [12]. As previously mentioned, the dilated layers in the table above were kept constant throughout, but we also wanted to explore how variable dilation factors affected performance. Given we have two convolution layers in each block, we set the first dilated convolution to a dilation factor of 3, and the second dilated convolution to a dilation factor of 6. All training was done on a single A100 GPU with 80GB of ram in PyTorch 1.10.

Dilation Factor	***Median***	***Min***	***Max***
2	0.492	0.237	0.658
4	0.529	0.189	0.645
8	0.372	0.177	0.521
16	0.126	0.023	0.228
Pyramid	0.525	0.317	0.659

TABLE II
DILATION FACTOR DICE SCORE

The main purpose of dilated layers is to improve efficiency, so we also noted the time it took to train the entire U-Net with each level of the dilation factor. The times are listed in table 3 with the total time and time per epoch.

Dilation Factor	***Time per Epoch (seconds)***
Original Model	145.05
2	145.64
4	149.53
8	150.33
16	151.07
Pyramid	151.13

TABLE III
TIME TO TRAIN DILATED CONVOLUTIONS

B. Separable Convolution Layers

Similarly to the dilated convolution layers, we wanted to investigate the effects of separable convolution layers on the performance and training efficiency of U-Nets. Again we replaced each convolution layer in the U-Net encoder with

a spatially separable convolution as well as a depth-wise convolutional layer. The results are presented in table 4 below.

Type of Convolution	*Median*	*Min*	*Max*
Spatially	0.533	0.364	0.669
Depth-Wise	0.558	0.363	0.675

TABLE IV
SEPARABLE DICE SCORE

Training was again done on an A100 GPU with 80GB of ram to compare training times to the previous dilated results. Time to train is presented in table 5 below.

Dilation Factor	*Time per Epoch (seconds)*
Original Model	145.05
Spatially	383.41
Depth-wise	376.16

TABLE V
TIME TO TRAIN SEPARABLE CONVOLUTIONS

IV. DISCUSSION

The results of our experiments are interesting and many conclusions can be derived from them. Firstly we can discuss the performance of both dilated and separable convolutions. Both variations as well as all levels of dilation factors contributed to lower performance compared to the original vanilla U-Net.

A. Dilated Layers

As mentioned, all levels of dilation as well as the pyramid dilated convolution architecture performed worse than the original U-Net. The dilation factor that preformed best was at a level of 4, followed closely by the pyramid implementation. This could be due to a variety of reasons, including the range of the receptive field. As the dilation factor increased from 4, the performance dramatically worsened, dropping all the way to a median dice score of 0.126. This is due to the fact that the receptive field is too large in comparison to the original 512x512 image, and therefore all local pixel relationships are lost. The pyramid implementation worked fairly well as it was in the dilation factor range of 3 and 6, which explains why its performance was very similar to a dilation factor of 4. All dilation factors increased the training time of the vanilla U-Net by a couple seconds, with higher dilation factors and pyramid implementation increasing the number of seconds. This is explained by the fact that dilated layers increase the receptive field without increasing computational requirements. Next steps would include changing the architecture of the U-Net to adapt to the increase in receptive field and decrease the training time.

B. Separable Layers

Both versions of separable convolutions led to similar but worse performance when compared to the vanilla U-Net. Spatially separated convolutions led to a median Dice score of 0.533 while depth-wise led to a median Dice score of 0.558. This is in line with the premise of separable convolutions where a single convolution is divided into two or more convolutions that lead to similar results. In this case performance was degraded but not significantly. The area which performed much worse than the original U-Net was in training time, where spatially separated convolutions and depth-wise separated convolutions were trained in 383.41 and 376.16 seconds respectively. This is over 2x the amount of time required to train the original U-Net. This could be due to many factors including the TensorFlow implementation of these layers. However more research needs to be done to understand why the compute time increased so dramatically.

C. Future Work

Overall all adaptations of U-Net hindered the performance and training efficiency when compared to the original U-Net. All of these adaptations included changes to the convolutional layers, however the architecture was kept the same. Future work could include changing the architecture of the U-Net in regards to the type of convolutional layers used. For example, with the larger receptive field of dilated layers, not as many encoder blocks would be required as it would account for more blocks. This would also significantly improve training efficiency as well as improve performance as well. Similar conclusions can be said for separable convolutional layers, where different implementations can improve training efficiency as well as performance.

ACKNOWLEDGMENT

The authors would like to thank Dr. Sana Syed and Adam Greene of the UVA Gastroenterology (GI) Data Science Laboratory for obtaining the EoE dataset through biopsies of UVA Medical Center patients. Additionally, the staff of the GI Data Science Laboratory provided direct medical feedback on our results.

REFERENCES

[1] E. S. Dellon, "Epidemiology of eosinophilic esophagitis," *Gastroenterology Clinics of North America*, vol. 43, no. 2, pp. 201–218, 2014.

[2] S. Carr, E. S. Chan, and W. Watson, "Eosinophilic esophagitis," *Allergy, Asthma and Clinical Immunology*, vol. 14, no. s2, pp. 1–11, 2018.

[3] T. M. Runge, S. Eluri, C. C. Cotton, C. M. Burk, J. T. Woosley, N. J. Shaheen, and E. S. Dellon, "Causes and Outcomes of Esophageal Perforation in Eosinophilic Esophagitis," *Journal of Clinical Gastroenterology*, vol. 51, no. 9, pp. 805–813, 2017.

[4] G. T. Furuta, C. A. Liacouras, M. H. Collins, S. K. Gupta, C. Justinich, P. E. Putnam, P. Bonis, E. Hassall, A. Straumann, and M. E. Rothenberg, "Eosinophilic Esophagitis in Children and Adults: A Systematic Review and Consensus Recommendations for Diagnosis and Treatment. Sponsored by the American Gastroenterological Association (AGA) Institute and North American Society of Pediatric Gastroenterol," *Gastroenterology*, vol. 133, no. 4, pp. 1342–1363, 2007.

[5] T. Hassan, B. Hassan, M. U. Akram, S. Hashmi, A. H. Taguri, and N. Werghi, "Incremental Cross-Domain Adaptation for Robust Retinopathy Screening via Bayesian Deep Learning," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, no. October, pp. 1–14, 2021.

[6] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," *33rd International Conference on Machine Learning, ICML 2016*, vol. 3, pp. 1651–1660, 2016.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, 5 2015.

[8] W. Adorno, A. Catalano, L. Ehsan, H. V. von Eckstaedt, B. Barnes, E. McGowan, S. Syed, and D. E. Brown, "Advancing eosinophilic esophagitis diagnosis and phenotype assessment with deep learning computer vision," *BIOIMAGING 2021 - 8th International Conference on Bioimaging; Part of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2021*, pp. 44–55, 2021.

[9] D. E. Brown, "Naïve Bayes, DS 6014: Bayesian Machine Learning," 2021.

[10] S. Wang, S.-Y. Hu, E. Cheah, X. Wang, J. Wang, L. Chen, M. Baikpour, A. Ozturk, Q. Li, S.-H. Chou, C. D. Lehman, V. Kumar, and A. Samir, "U-Net Using Stacked Dilated Convolutions for Medical Image Segmentation," *arxiv*, 2020.

[11] P. K. Gadosey, Y. Li, E. A. Agyekum, T. Zhang, Z. Liu, P. T. Yamak, and F. Essaf, "SD-UNET: Stripping down U-net for segmentation of biomedical images on platforms with low computational budgets," *Diagnostics*, vol. 10, no. 2, 2020.

[12] W. Zhang, X. Lu, Y. Gu, Y. Liu, X. Meng, and J. Li, "A robust iris segmentation scheme based on improved U-Net," *IEEE Access*, vol. 7, pp. 85082–85089, 2019.