Analyzing the Effectiveness of Image Augmentation Methods for Image Segmentation in Diabetic Foot Ulcer's

Christian Jordan*
*Faculty of Science and Engineering, Manchester Metropolitan University, Manchester, UK

***Abstract*—This research implements U-Net and DeepLabV3_MobileNetV3 deep learning image segmentation models to investigate the effectiveness of image augmentation techniques on the Diabetic Foot Ulcer dataset. Evaluation of the best models is performed using metrics dice loss and intersection-over-union. The model showed improvements in both metric finding that methods such as rotation, flipping and cropping can increase the latter from 0.6931 to 0.7340 by basic implementation.**

***Index Terms*—Image augmentation, medical imaging, deep learning, neural networks, segmentation**

I. Introduction

Diabetic Foot Ulcer's (DFU), a typically chronic disease that can lead to extremity amputation is a major cause for concern within the health industry. Overall, about 5% of patients with diabetes mellitus develop foot ulcers and 1% end up with an amputation [1].

With an estimated 50% mortality rate after 5 years [2], ongoing research for classification and segmentation of wounds using deep learning can contribute to decreasing diagnosis time and preventing wounds from progressing to latter stages of disease.

This project focuses on augmentation techniques leading to improved accuracy in wound segmentation models for purposes including accurate wound localisation, quantitative wound measurement, wound tissue classification and wound healing monitoring.

II. Project Prototype

A full project prototype for implementation of this research can be found on the GitHub link below (preferred).

https://github.com/christianmcb/Deep-Learning-Project/

An example of the main project notebook running in Google colab is also available on:

https://colab.research.google.com/drive/1eEYf_OWKfS1folXeuDpSoNepA87EhZG6?usp=sharing

III. Related Work

Researchers have been working on developing models that can be used to predict labels for image segmentation tasks, such as U-Net [3], SegNet [4], V-Net [5] and more. These models are examples of convolutional neural networks that are similar in structure and strategy with alterations to account for differences in feature depth within the images. This makes every image segmentation task unique and the research potential for finding the best models and techniques a mammoth task.

Prior works on this specific dataset focus on enhancing the model configurations such as replacing the decoder in HarDNet-MSEG for increased accuracy and decreased speed [6]; or studying loss function impact on the models potential [7].

Perhaps the most interesting of research for Diabetic Foot Ulcer's on the analysis of augmentation methods is the proposal of a refined mix-up augmentation technique proposed by Hresko et al. [8], who studied the effects of mixing images within the training set and improved the Dice Score from 0.6530 to 0.6713.

Many of these methods use additional image augmentation techniques, but little evaluation and statistical analysis has been provided on the effects of them individually and collectively to improve model scores. This potential gap forms the basis of this project.

IV. Methodology

A. Dataset and Groud Truth

The DFUC 2022 dataset has approved for use by the UK National Health Service (NHS) and courtesy of images from Lancashire Teaching Hospital for the purposes of research; and provided in coalition with Manchester Metropolitan University [9].

Consisting of 2000 images for training and validation, of size 640x480 (WxH) and aspect ratio 1.333, Table I.

	Channels	Height	Width	Label Size (% of Total)
count	2000.0	2000.0	2000.0	2000.000000
mean	3.0	480.0	640.0	0.023663
std	0.0	0.0	0.0	0.036806
min	3.0	480.0	640.0	0.000298
25%	3.0	480.0	640.0	0.004366
50%	3.0	480.0	640.0	0.010367
75%	3.0	480.0	640.0	0.026339
max	3.0	480.0	640.0	0.347518

TABLE I: Descriptive statistics of training data.

Examples from the dataset choosing images from the smallest 1% label sizes (row 1), middle 5% (row 2) and largest % (row 3) are shown in Figure 1. Notice that examples from the smallest label size are difficult to predict with the naked eye,

and may require specialist review to be diagnosed. This may make it difficult for a model to predict.

Fig. 1: Sample images from the dataset.

Ground Truth Labels

From Table I, mask annotations have a large range in the training data, suggesting models need adaptability to be able to make accurate predictions. Figure 2 shows the distribution of label sizes within the training data, with data highly positively skewed,

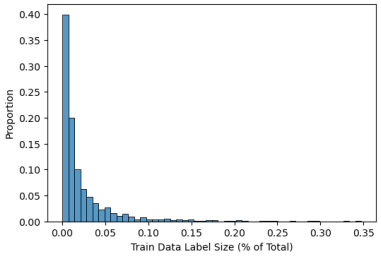

Fig. 2: Distribution of label sizes.

B. Segmentation Models

U-Net Architecture

U-Net is a collection of convolutional neural networks designed for biomedical image segmentation. Consisting of a contracting path to capture context and a symmetric expanding path to enable precise localisation [3].

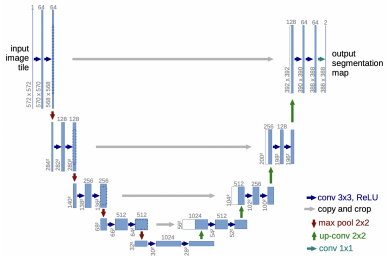

Fig. 3: Visual representation of U-Net model architecture [3].

The up-sampling strategy for implementing the U-Net model in this project includes an encoding block with two 2D convolutions (kernel size 3), two x batch normalisation, ReLU activation and max pooling, whereas the down-sampling strategy reverses up-sampling by applying the same transformation backwards. U-Net uses skip connections to copy some of the data from the encoding block and retain some of the spatial information lost in up-sampling, increasing the accuracy in segmentation tasks.

U-Net can be adapted to capture more or less significant information by including more or less encoding/decoding blocks, depending on the specific task at hand, however deeper models take a longer time to train.

U-Net is a large deep learning model with many parameters due to the skip connections and additional layers in the decoding path [10], making it computationally expensive and prone to over-fitting.

DeepLabV3 MobileNetV3 Architecture

MobileNetV3 is the backbone architecture for this model, and uses a inverted residual block consisting of 2D convolutional layers to increase the number of channels, a batch normalisation layer to normalise and rescale the output and a ReLU activation function to limit computational cost increasing exponentially [11].

The backbone uses 2x 2D convolutional layers and 15 inverted residual blocks as part of the up-sampling process, increasing the number of channels from 3 to 960 iteratively.

DeepLabV3 introduces the idea of atrous convolutions [12], which build on the convolutions that might be seen in models such as U-Net by establishing a new parameter; rate **r**, that, to put simply, spaces out the kernels and allow them to look at larger areas of the images, without increasing the calculation size [13]. It is used to classify the output from MobileNetV3 to provide a prediction of the mask. Consisting of 2x 2D convolutional layers, 3x ASPP (atrous) convolutional blocks (increased stride and dilation to convolutional layers) and 1x Atrous Spatial Pyramid Pooling (ASPP) block to capture a broader range of features.

The DeepLabV3 architecture reduces the number of channels from the output of the MobileNetV3 encoder from 960 to a single output channel, corresponding to the label prediction.

C. Loss Function

Every deep learning model requires a loss function to monitor how well the model is performing on the training

data, with the goal of updating model weights and minimising loss.

Sigmoid Activation

The Sigmoid Activation Function is important for the prediction of binary segmentation masks such as in this project. Binary masks contain values in the range $(0, 1)$, however predictions can easily fall outside of this range, hindering the calculation and effectiveness of loss functions and exaggerating predictions of outliers.

Model predictions can be transformed using the formula in Equation 1.

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{1}$$

where, x =predicted pixel value.

Dice Loss

Dice Loss as a loss function in deep learning models for medical imaging was first introduced by Milletari et al. [14] and has recently been utilised in many variations to increase accuracy in image segmentation. Other loss functions such as cross entropy loss calculate the per pixel loss and fail to acknowledge whether the correctly predicted pixels are boundaries or not [15].

It can be calculated for binary instances; predicting a single class, such as this using Equation 2. With the values of the label between 0 and 1, dice loss requires a sigmoid function which transforms the predictions to this range.

$$D = 1 - \frac{2\sum_i^N p_{i,j} g_{i,j}}{\sum_i^N p_{i,j} + g_{i,j}} \tag{2}$$

where, $p_{i,j}$ = model output and $g_{i,j}$ = label.

BCE With Logits Loss

Binary cross entropy loss (BCE Loss) uses the predicted values as probabilities that the pixel is either 1 or 0 (wound or not) and calculates the total log loss by comparison.

The formula for calculating log loss is shown in Equation 3 [16].

$$l_n = -w_n[y_n \cdot \log \sigma(x_n) + (1 - y_n) \cdot \log(1 - \sigma(x_n))] \tag{3}$$

$$l(x, y) = mean\left(L\{l_1, l_2, ..., l_N\}\right)$$

where, N = batch size, y_n = label and x_n =, predicted probability of the pixel belonging to the wound.

Working with binary masks of 0 and 1 requires the prediction to form a probability between this range, therefore BCE **With Logits Loss** combines a Sigmoid function to transform the predictions accordingly.

D. Optimiser

In neural networks, an optimisers aim is to update the weights of the model such that the loss function is minimised efficiently.

Given a vector of input variables (image) $x = \{x_1, ..., x_N\}$ and the actual outputs (label) $y = \{y_1, ..., y_N\}$, the goal is to find a model that maps x to a function $f(x)$ such that the residuals (loss) between x and y is at it's lowest [17].

Stochastic Gradient Descent

Gradient descent initially provides a function of independent variables $V = \{v_1, ..., v_r\}$, that are the variables to be updated by the algorithm shown below. The weights are updated by a fraction of the derivative, called the learning rate γ, to ensure the gradient descent isn't excessive and the model converges to the minimum point. It is important to tune the learning rate parameter so that the model converges quickly to the minimum point.

$$\nabla V = \left(\frac{\partial V}{\partial v_1}, ..., \frac{\partial V}{\partial v_r}\right)$$
$$V_n \leftarrow V_{n-1} - \gamma \cdot \nabla V_{n-1} \tag{4}$$

Stochastic Gradient Descent (SGD) is a variation of the gradient descent algorithm in Equation 4, where the gradient is only calculated for a small number of the observations. This will reduce the computation time per epoch, however the model may then require a higher number of epochs to converge. Therefore, this will reduce the total computation time in some cases.

Adam

Adam is another optimiser based on gradient descent, however introduces new variables into Equation 4 to help with faster convergence. The first moment estimation m_n (initially 0), tracking the exponential moving averages of the gradient, allowing for more accurate variable updating and faster convergence. The second moment estimation s_n (initially 0), tracks the moving average of the squared gradients (uncentered variance) and helps the optimiser adapt the learning rates of individual parameters during the learning process [18].

$$\begin{aligned}
m_n &= \beta_1 \cdot m_{n-1} + (1 - \beta_1) \cdot \nabla V_{n-1} \\
s_n &= \beta_2 \cdot s_{n-1} + (1 - \beta_2) \cdot (\nabla V_{n-1})^2 \\
\hat{m_n} &= \frac{m_n}{1 - \beta_1^n} \\
\hat{s_n} &= \frac{s_n}{1 - \beta_2^n} \\
V_n &\leftarrow V_{n-1} - \frac{\gamma \cdot \hat{m_n}}{\sqrt{\hat{s_n}} + \epsilon}
\end{aligned} \tag{5}$$

where ϵ is an small adjustment (default $1e{-}8$) to avoid division by 0 when $\hat{s_n}$ is 0.

E. Image Augmentation Techniques

This section introduces the image augmentation techniques that have been used throughout the experiments in this project.

Image augmentation methods artificially create new data within existing data by transforming the images in order to improve generalisation of deep learning models to new data. For example, shifting or rotating the wound may help the

model increase accuracy of similar wounds that exist in a different location than the existing wound.

A downside to image augmentation can be that it takes longer for the model to train, which can be undesirable if the data is very large.

Albumentations is an image augmentation/transformation library that contains the algorithms for implementing the augmentations outlined in this section. The implementation for each of the following takes a parameter p which is the probability that the image is transformed.

Examples of the image augmentation techniques in practice can be found in Figure 4.

Flipping

Flipping is of the most basic image augmentation methods; requires flipping the image around the x-axis or the y-axis. Equation 6 is for flipping horizontally, and 7 vertically.

$$dst_{i,j} = src_{i,src.cols-j-1} \tag{6}$$
$$dst_{i,j} = src_{src.rows-i-1,j} \tag{7}$$

Rotation

The formula for rotating an image by given angle is shown by the matrix multiplication in Equation 8.

$$dst_{i,j} = src_{i,j} \cdot \begin{bmatrix} \alpha & \beta & (1-\alpha) \cdot center.x - \beta \cdot center.y \\ -\beta & \alpha & \beta \cdot center.x + (1-\alpha) \cdot center.y \end{bmatrix} \tag{8}$$

where, $\alpha = \cos(angle)$ and $\beta = \sin(angle)$.

Cropping

Cropping the image is simple once the bounds have been initialised, that is, minimum and maximum values of i and j to be included in the crop, then either remove the outer values and resize the image, or pad the outer values with a discrete value, such as 1.

if $(x_{min} < i < x_{max})\&(y_{min} < j < y_{max})$]:

$x_{i,j} = x_{i,j}$

else:

$x_{i,j} = 1$

For this project, the image then needs to be resized so that it can be understood by the model correctly.

Gaussian Noise

Applying random Gaussian noise to the images in the training set can again help with the generalisation of the model.

$$dst_{i,j} = src_{i,j} + \nu_{i,j} \tag{9}$$

where $\nu =$ normally distributed random variable with mean μ and variance σ^2.

Coarse Dropout

With similar intuition to that of using Gaussian noise, coarse dropout also introduces noise to the data, by adding a number of "holes" to the image, again helping the model to generalise to unseen data.

It works by using a random number generator to choose the number of holes to apply (up to parameter "max_holes"), and then the position of the holes on the image, and fills the selected pixels in the range of the holes with a discrete value such as 1.

Brightness Contrast

The Random Brightness Contract augmentation changes the brightness of the image by a certain value m.

$$dst_I = lut(src_I) \tag{10}$$

where $dst =$ output, $src =$ input and lut is a look-up table transform based on the input array provided by OpenCV.

Elastic Transformation

In affine transformations such as shifting and rotating, pixels $x_{i,j}$, i and j are transformed in the same direction so the resulting image is visually similar in a changed position.

Elastic transformations take a different approach and transforms the pixels in different directions based on a set of parameters [19], using a Gaussian filter (smoothing) to keep the characteristics of the image intact.

This transformation increases epoch iteration time by 5-6x in comparison to other transformations included, which is to be considered when choosing an augmentation strategy to apply.

F. Experimental Setup

Training and Validation Sets

Given the full training data of 2,000 images and mask labels, to validate the results of the segmentation models, the data has been split into training and validation sets of 1,600 and 400 images respectively. This can be achieved using Sk-Learn's feature "train_test_split", which will randomly split the data into train and validation sets for training models effectively. The dataset used for this project has already been split into training and validation sets as part of the Grand Challenge, and in line with this challenge, it is important to keep the same validation sets so results can be compared.

Evaluation Metrics

In order to compare models, experimental setup must include an evaluation metric. In this project, working with the intersection-over-union (IOU), which can be calculated using Equation 11.

$$IOU = \frac{x \cap y}{x \cup y} \tag{11}$$

where x is the predicted label values and y the actual label values.

Device Specification

The model has been built using Python (version 3.9.16), PyTorch (version 1.12.1) and CUDA version 11.6. Full environment setup and project prototype can be found on **enter github**. The GPU used for training and testing the models in this project is the NVIDIA Quadro P5000 with 16GB RAM.

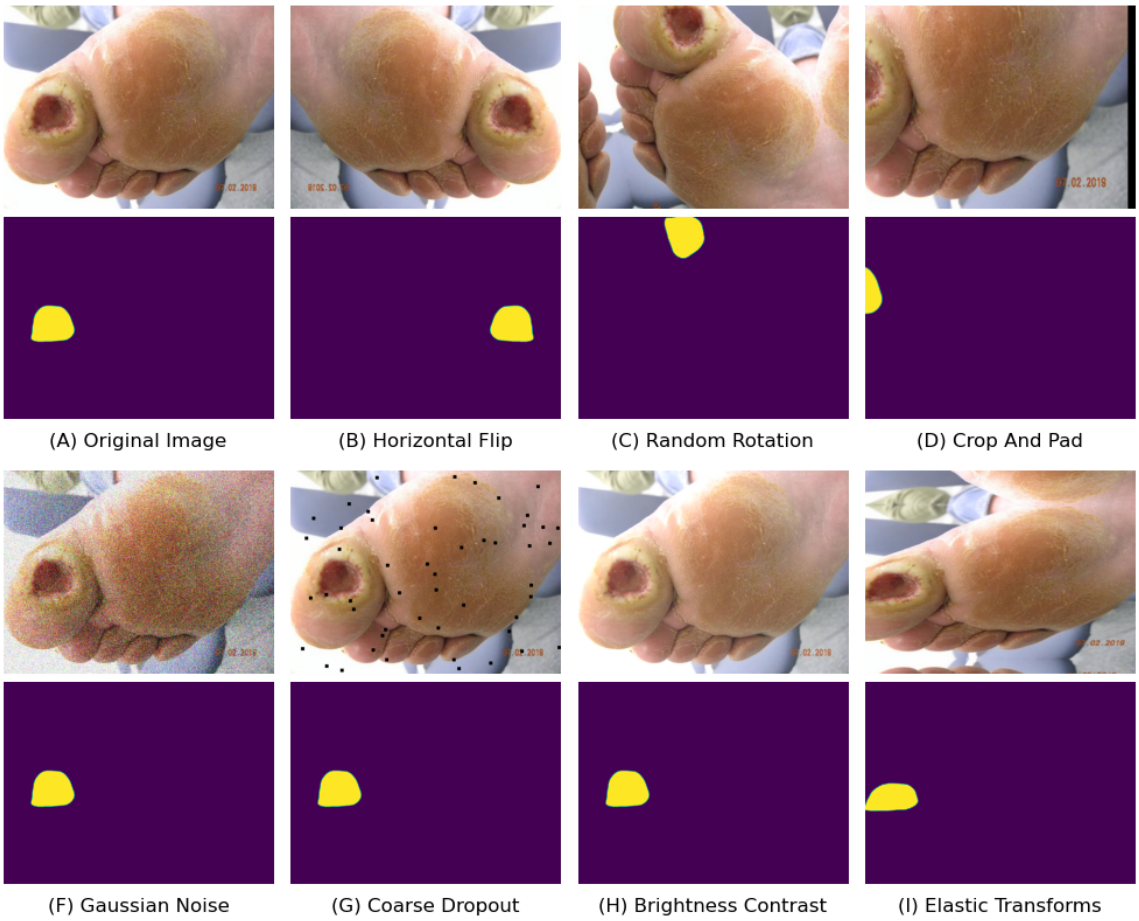

Fig. 4: Examples of image augmentation techniques.

V. Experiments

A. Initial Experiments and Model Tuning

In order to carry out the image augmentation experiments to find most effective techniques for the Diabetic Foot Ulcer dataset, first establish some of the best performing models based on loss function, optimiser, batch size and learning rate.

Dice Loss vs BCE With Logits Loss

The Dice Loss function outperforms BCE with Logits Loss marginally and should be selected based on the results in Table II.

Model	Optimiser	Batch Size	Loss Function	Epochs	Train IOU	Val IOU
UNet	Adam	16	**Dice Loss**	48	0.7495	**0.6099**
UNet	Adam	16	BCEWithLogits	48	0.8318	0.5980

TABLE II: Results from loss function comparison.

SGD vs ADAM

Following the results of the optimiser experiments, Adam has been chosen as the best optimiser with results shown in Table III, converging at a faster rate and with better generalisation to the unseen data.

Model	Optimiser	Batch Size	Epoch	Train IOU	Val IOU
Deep Lab	Adam	16	21	0.911862016	0.693143725
Deep Lab	SGD	16	23	0.723619699	0.608266592

TABLE III: Results from SGD and Adam optimisers.

Best Batch Size

Using a batch size of 16 provides the best performing model whilst still converging fairly quickly, showing that the model can learn from a larger batch size and retain the information. Larger batch sizes are preferred where there is no drop in metric, and 16 is the largest batch size that the GPU can deal with due to memory constraints.

Model	Optimiser	Batch Size	Loss Function	Epoch	Train Loss	Train IOU	Val Loss	Val IOU
Deep Lab	Adam	**16**	Dice Loss	21	0.04668	0.9119	0.1844	**0.6931**
Deep Lab	Adam	4	Dice Loss	18	0.0868	0.8471	0.2136	0.6629
Deep Lab	Adam	8	Dice Loss	38	0.0476	0.9114	0.2108	0.6613
Deep Lab	Adam	2	Dice Loss	17	0.1078	0.8149	0.2451	0.6375

TABLE IV: Results from batch size optimisation.

Learning Rate Tuning

The learning rate (LR) is a variable weight that controls the rate at which the model parameters are updated each iteration (or epoch). Optimising the learning rate for the optimiser is

not a task involving the best model based on the evaluation metric IOU. Instead, look to find the LR that descends to the lowest point fastest, without converging or diverging.

It is optimal to use a larger batch size with a higher LR if model accuracy is not sacrificed by doing so.

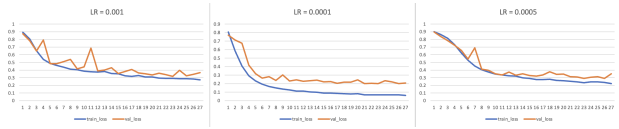

Fig. 5: Training and validation loss by learning rate.

Figure 5 shows the trajectory of the training and validation losses as the model iterates through a number of epochs. Large fluctuations in validation losses such as in plot "LR=0.001" suggest LR is large, whereas the validation losses following the training losses closely, "LR=0.0001", suggest a low LR. The true optimal LR is in the middle "LR=0.0005".

U-Net vs DeepLab

DeepLabV3_MobileNetV3 clearly outperforms this implementation of the U-Net model as shown in Table V, and also converges at a much faster rate, making it ideal for this problem.

Model	Optimiser	Batch Size	Epoch	Train IOU	Val IOU
Deep Lab	Adam	16	21	0.911862016	0.693143725
UNet	Adam	16	48	0.74946988	0.60994148

TABLE V: Results of U-Net vs Deep Lab

B. Selected Model and Augmentation Experiments

Selected Model

After initial experiments to find optimal model and parameters, the selected variables for image augmentation experiments are shown in Table VI.

Parameter	
Model	DeepLabV3_MobileNetV3
Batch Size	16
Max Epochs	1000
Early Stopping	15 Epochs
Initial LR	0.0005
Scheduler	Reduce on Plateau
Scheduler Factor	0.5
Scheduler Patience	5
Loss Function	Dice Loss
Evaluation	Batch IOU

TABLE VI: Selected model specification.

Augmentation Experiments

Using the model specification in Table VI, experiments were run by altering the image augmentation techniques outlined in Section IV-E on the training set.

The models are evaluated by the batch validation IOU and evaluated individually in section VI.

VI. Results and Discussion

Results of Image Augmentation

Table VII shows a clear improvement on the average validation loss and IOU metrics for a range of image augmentation methods, proving that techniques such as rotation, flipping, cropping and elastic transformation have a positive impact on the generalisation of the models on unseen data.

Interestingly, the training loss on the baseline model reaches as low as 0.0467, showing that the model has started to over-fit with its best performance on the validation data. In contrast, when applying transformations, the model has a higher training loss yet has higher validation IOU and lower validation loss, confirming the improved generalisation.

Building the Best Model

Once the most effective image augmentation techniques have been discovered, Table VIII shows how these methods work together in improving the overall model compared with the baseline. With IOU being the most robust evaluation metric for medical imaging segmentation, the increase from 0.6931 to 0.7340 should be considered large.

Augmentation	Epoch	Train Loss	Train IOU	Val Loss	Val IOU	P-Value
Best Model	61	0.100312	0.816992	0.1587327	0.733960	0.013428
None	21	0.046680	0.911862	0.184362	0.693144	Baseline

TABLE VIII: Results of the best performing model.

In order to compare the means of the two model evaluations on the validation set, use a non-parametric test as the results are not normally distributed, such as the Mann-Whitney U Test for difference in means.

Using the baseline and best models to predict individual images in the validation set, run the statistical test at the 5% significance level where:

- H_0: The distribution of values predicted by the baseline are not less than the best predictions.
- H_1: The distribution of values predicted by the baseline are less than the best predictions.

Since the P-Value of 0.0134 is less than 0.05, there is sufficient evidence to reject the null hypothesis and conclude that the best model is statistically better than the baseline model for the validation data.

Predictions Evaluation

It is clear from visualising some of the predictions such as in Figure 6 that the best model using image augmentation can better localise the wound in both small and large ulcerations, creating better versatility and generalisation to new data, confirming the results in Table VIII.

Validation of Best Model

In order to validate the results of the best model, build it again using the same augmentation techniques, however split an extra 200 images into a testing set. Therefore, this time there are 1,200 images for training, 400 for validation and 200 for testing.

Augmentation	Parameters	Probability	Epoch	Train Loss	Train IOU	Val Loss	Val IOU
None - Baseline	None	1.0	21	0.046679888	0.911862016	0.184362233	**0.693143725**
Rotate	Limit = 90	0.5	24	0.097143345	0.826351762	0.165319741	0.723573983
Shift Scale Rotate		0.5	25	0.077971622	0.857169867	0.168767393	0.715779126
Vertical Flip		0.5	24	0.072463147	0.867014885	0.168853432	0.715564787
Crop And Pad	Percent=(-0.3x0.05)	0.5	31	0.078433134	0.853652835	0.173725367	0.709363103
Elastic Transform		0.5	25	0.074385449	0.862742901	0.174507424	0.707136452
Horizontal Flip		0.5	22	0.063207187	0.88263607	0.176180914	0.70599252
Crop And Pad	Percent=0.5	0.5	32	0.069142766	0.871650398	0.17706342	0.701804638
Coarse Dropout	Max Holes=50, H8W8	0.5	20	0.054564618	0.897100747	0.182362184	0.698696673
Gaussian Noise	Var=(10x50)	0.5	16	0.078295268	0.857126772	0.185060501	0.696708739
Gaussian Noise	Var = (100x5000)	0.5	25	0.109427497	0.805231154	0.183787584	0.694526911
Coarse Dropout	MaxHoles=8, H(8)W(8)	0.5	21	0.046302523	0.911655426	0.184352353	0.691724002
Random Brightness Contrast		0.2	20	0.050479174	0.904540718	0.188359007	0.690256774

TABLE VII: Results of individual image augmentation techniques on training and validation metrics.

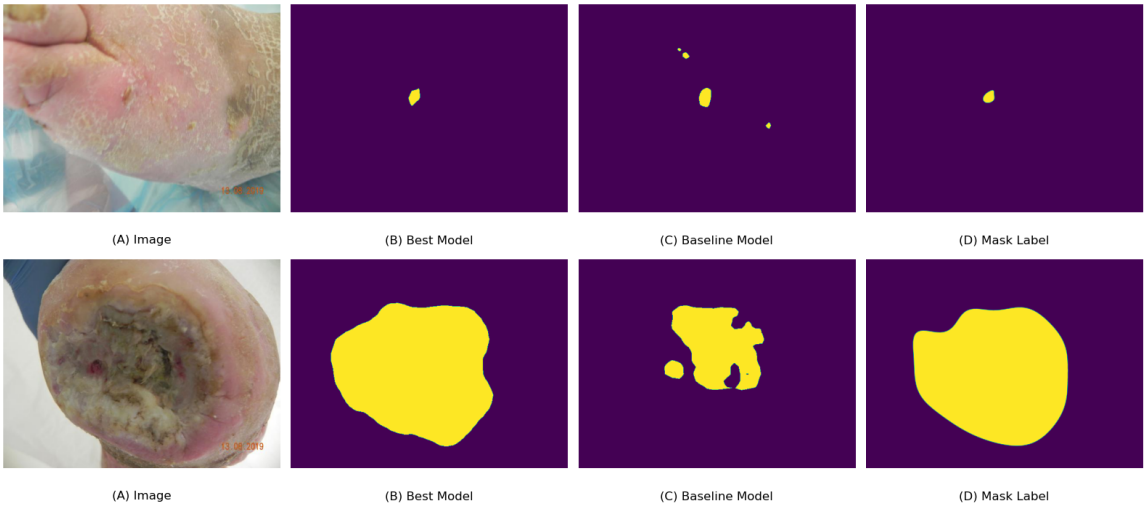

Fig. 6: Predicted masks and label.

Model	Epoch	Train IOU	Val IOU	Test Loss	Test IOU
Best Model	31	0.79024446	0.71893096	0.21769898	0.65977675
Baseline	21	0.91682452	0.699780583	0.21832837	0.6495559

TABLE IX: Results of validation with 200 test images.

The results of the validation test are not as clear, showing smaller differences in the test loss and IOU as shown in Table IX, therefore more analysis would be needed to confirm that this experiment is statistically significant.

VII. Conclusion

The results of this research support the use of image augmentation methods such as rotation, shifting, scaling, flipping, cropping and elastic transforms in the segmentation of medical images, with improved model results. Further works would need to be carried out to properly validate these results if time constraints were not an issue. In order to further validate the results, more data would help greatly, including the use of the testing set available on the Grand Challenge website. Training the best and baseline models on the full training data and consequently validating with this test data could produce improved models.

There are also many more augmentation methods available to be experimented with, and also the opportunity to explore new creations of augmentation techniques.

Acknowledgment

I am personally grateful to Professor Moi Hoon Yap and Senior Researcher Mr Bill Cassidy for guidance and teaching in the foundations of this research.

References

[1] M. M. Oliver TI, "Diabetic foot ulcer," 2023. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK537328/

[2] L. Chen, S. Sun, Y. Gao, and X. Ran, "Global mortality of diabetic foot ulcer: A systematic review and meta-analysis of observational studies," *Diabetes, Obesity and Metabolism*, vol. 25, no. 1, pp. 36–45, 2023. [Online]. Available: https://dom-pubs.pericles-prod.literatumonline.com/doi/abs/10.1111/dom.14840

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," 2016.

[5] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," 2016.

[6] T.-Y. Liao, C.-H. Yang, Y.-W. Lo, K.-Y. Lai, P.-H. Shen, and Y.-L. Lin, "Hardnet-dfus: An enhanced harmonically-connected network for diabetic foot ulcer image segmentation and colonoscopy polyp segmentation," 2022.

[7] A. Galdran, G. Carneiro, and M. Ángel González Ballester, "On the optimal combination of cross-entropy and soft dice losses for lesion segmentation with out-of-distribution robustness," 2022.

[8] D. J. Hresko, J. Vereb, V. Krigovsky, M. Gayova, and P. Drotar, "Refined mixup augmentation for diabetic foot ulcer segmentation," in *Diabetic Foot Ulcers Grand Challenge*, M. H. Yap, C. Kendrick, and B. Cassidy, Eds. Cham: Springer International Publishing, 2023, pp. 92–100.

[9] C. Kendrick, B. Cassidy, J. M. Pappachan, C. O'Shea, C. J. Fernandez, E. Chacko, K. Jacob, N. D. Reeves, and M. H. Yap, "Translating clinical delineation of diabetic foot ulcers into machine interpretable segmentation," 2022.

[10] S. Premanand, "Top 8 interview questions on unet architecture," Jan 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2023/01/top-8-interview-questions-on-unet-architecture/

[11] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," 2019.

[12] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017. [Online]. Available: http://arxiv.org/abs/1706.05587

[13] A. Sankar, "A primer on atrous convolutions and depth-wise separable convolutions," Sep 2021. [Online]. Available: https://towardsdatascience.com/a-primer-on-atrous-convolutions-and-depth-wise-separable-convolutions-443b106919f5

[14] F. Milletari, N. Navab, and S. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," *CoRR*, vol. abs/1606.04797, 2016. [Online]. Available: http://arxiv.org/abs/1606.04797

[15] S. Du, "Understanding dice loss for crisp boundary detection," Jul 2022. [Online]. Available: https://medium.com/ai-salon/understanding-dice-loss-for-crisp-boundary-detection-bb30c2e5f62b

[16] PyTorch, "Bcewithlogitsloss¶." [Online]. Available: https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html

[17] M. Stojiljković, "Stochastic gradient descent algorithm with python and numpy," Apr 2023. [Online]. Available: https://realpython.com/gradient-descent-algorithm-python/

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[19] P. Y. Simard, D. Steinkraus, J. C. Platt *et al.*, "Best practices for convolutional neural networks applied to visual document analysis." in *Icdar*, vol. 3, no. 2003. Edinburgh, 2003.