Influence of background preprocessing on the performance of deep learning retinal vessel detection

James Owler[a], **Peter Rockett**[b]
[a]Bioengineering—Interdisciplinary Programmes Engineering, University of Sheffield, S1 3JD, UK
[b]Dept. of Electronic and Electrical Engineering, University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK

Abstract.

Purpose: Segmentation of the vessel tree from retinal fundus images can be used to track changes in the retina, and be an important first step in a diagnosis. Manual segmentation is a time consuming process that is prone to error; effective and reliable automation can alleviate these problems but one of the difficulties is uneven image background which may affect segmentation performance.

Approach: We present a patch-based deep learning framework, based on a modified U-Net architecture, that automatically segments the retinal blood vessels from fundus images. In particular, we evaluate how various pre-processing techniques: images with either no processing, N4 bias field correction, contrast limited adaptive histogram equalization (CLAHE), or a combination of N4 and CLAHE, can compensate for uneven image background and impact final segmentation performance.

Results: We achieved competitive results on three publicly available datasets as a benchmark for our comparisons of pre-processing techniques. In addition, we introduce Bayesian statistical testing which indicates little practical difference ($\Pr > 0.99$) between pre-processing methods apart from the sensitivity metric. In terms of sensitivity and pre-processing, the combination of N4 correction and CLAHE performs better in comparison to unprocessed and N4 pre-processing ($\Pr > 0.87$) but compared to CLAHE alone, the differences are not significant ($\Pr \approx 0.38 - 0.88$).

Conclusions: We conclude that deep learning is an effective method for retinal vessel segmentation and that CLAHE pre-processing has the greatest positive impact on segmentation performance, with N4 correction helping only in images with extremely inhomogeneous background illumination.

Keywords: retinal vessel segmentation, deep learning, U-Net, fundus imaging, Bayesian hypothesis testing, image background correction.

*Peter Rockett, p.rockett@sheffield.ac.uk

1 Introduction

Retinal diseases are a major public health concern in both the aged and working populations. For example, diabetic retinopathy (DR) is one of the leading causes of blindness in an aging population. There are an estimated 93 million people with DR worldwide, a number which is only expected to rise[1,2]. Non-invasive visual inspection can provide valuable insight into the condition of an eye. The structure of retinal blood vessels is an important indicator of disease, and observing and measuring changes in vessel morphology, such as branching patterns and vessel width, can result in accurate early detection of retinal diseases[3]. The assessment of retinal blood vessels may reveal conditions such as hypertension, diabetic retinopathy, stroke, atherosclerosis, and cardiovascular disease[4].

Screening programs of at-risk groups can provide a mechanism for early detection, and thus prevent blindness, but as programs become more extensive, analysis of large volumes of data is becoming an increasingly challenging task for clinicians[5]. Segmentation of retinal blood vessels can also be useful for several other reasons including longitudinal monitoring for diagnosis and disease progression, computer-assisted laser surgery, and biometric identification[4,6].

The manual segmentation of vessels is time-consuming, expensive, and prone to inter- and intra-rater variability. Automating vessel segmentation can dramatically increase the throughput and efficiency of screening processes, as opposed to lengthy and tedious manual labeling. Automation can also increase reproducibility. and decrease subjectivity. Some sort of human intervention is still usually required when it comes to decision making on the clinical pathway; however, as technology develops and our confidence in the technology grows, automation will undoubtedly play a huge role in the new era of technology-driven precision medicine[7–9].

Over the past 20 years, a large amount of research and published work has been devoted to the automated segmentation of retinal blood vessels. In general, methods may be considered unsupervised or supervised. Unsupervised methods utilize prior knowledge of vessel structure and often rely on rule-based schemes. These algorithms revolve around morphological processing, vessel tracking, filter matching, multi-scale, or model-based approaches. Supervised methods, on the other hand, utilize labeled ground truth data to train a classifier model, and optimization during the training stage is used to infer a functional mapping between input and output data. The trained model is then used to predict outputs for unseen data in a testing stage. A comprehensive review of segmentation methods is beyond the scope of this paper although reviews of previous work can be found in[10] and.[11]

Deep learning has recently emerged as a promising approach for automated retinal vessel segmentation[12] where deep learning refers to neural networks with many layers that are capable of automatically extracting features[13]. An advantage of deep learning, compared to classical machine learning, is its self-organizing capability. Classical machine learning algorithms require carefully extracted, hand-crafted features, which is a time consuming and complex process that requires domain expertise. Deep learning methods, on the other hand, can automatically learn features from raw input data. This has led to a paradigm shift in which researchers are now focused on optimizing network architectures and training processes, as opposed to hand-crafting features. Recently, deep learning methods have been utilized within the medical imaging community to address a variety of problems across many different applications and imaging modalities[14].

Deep learning methods, and more conventional methods, are commonly proposed in conjunction with some form of pre-processing. Researchers have found that pre-processing the images beforehand often gives improved results, however, there is often little justification for why certain methods are chosen, and no consensus on what methods work best. In this paper, we investigate the impact of various pre-processing techniques previously published in the literature on the performance of a deep learning based retinal vessel segmentation method. We evaluate methods in a systematic and statistically robust way in addition to highlighting some limitations currently found in the literature, and suggest future improvements.

1.1 Previous Work

1.1.1 Deep Learning Methods for Retinal Vessel Segmentation

Many state-of-the-art deep learning based segmentation methods employ convolutional neural networks (CNNs). CNNs were introduced by LeCun,[15] but have recently gained wider recognition after the seminal paper of Krizhevsky et al.[16] since they can automatically learn representations of image data with increasingly complex levels of extraction via convolutional layers. Convolutional layers drastically reduce the number of weights to be learned when compared to traditional, fully-connected neural networks due to weights being shared among convolutional layers.

Liskowski and Krawiec[17] were one of the first to publish work detailing a deep learning approach to retinal vessel segmentation. These authors randomly extracted 20,000 patches from each image in a training set; only patches inside the field of view (FOV) were considered. The decision on the class of a particular pixel was based on an $m \times m$ patch centered at that pixel. Several basic architectures were studied; a stack of three fully-connected layers preceding some combination of convolutional layers were used in all. Data augmentation was applied to images throughout the training process in the form of scaling, rotation, flipping, and gamma correction. These authors also explored the notion of 'structure prediction' whereby the model classifies multiple pixels within the center of a patch rather than just a single pixel. The learning of spatial relationships between neighboring pixels allowed the network to better reconstruct the boundaries of vessels.

U-Net, introduced by Ronneberger et al.[18], is a deep convolutional neural network geared towards biomedical image segmentation that has been successfully applied to various tasks[14]. U-Net is a development of the fully-convolutional network architecture[19] with no fully-connected layers present in the network, which allows for unrestricted input image sizes and a reduction of computational cost. A contracting, encoding path is followed by an expanding, decoding path. Shallow layers in the network capture local information while deeper layers (whose retrospective field is much larger) capture global information. The expanding decoder recovers a full resolution pixel-to-pixel label map. Skip connections, that bypass layers, enable features to be concatenated between the encoder and decoder avoiding bottlenecks in the network, and improving generalization[20].

For retinal vessel segmentation, Oliveira et al.[21] used the stationary wavelet transform (SWT) combined with a U-Net-inspired fully convolutional neural network. Patches were randomly extracted from the green channel and images transformed with the SWT. Data augmentation was applied in the form of random flipping. The authors suggest using the SWT provides the network with extra information, which in turn can generate more useful features and lead to better segmentation performance.

Yan et al.[22] utilized a U-Net like model in combination with both pixel-wise and segment-level loss—previous deep learning methods had only used pixel-wise loss. These authors suggested using this novel loss function can balance the importance of thick and thin vessels during training leading to more effective features being learned. Green channel patches of 128×128 pixels were extracted from images in the training set and data augmentation applied in the form of: flipping, rotation, resizing, and adding random noise.

Cherukuri et al.[23] exploited knowledge about the structure of vessels by embedding two geometrical priors into the loss function that was used to jointly train two networks: a representation layer and a residual task network. These two priors had the effect of: i) encouraging diversity in vessel orientation, and ii) adaptively penalizing noise to suppress false positives. The authors claim this leads to better labeling of thin vessels as well as a less heavily parametrized network.

The very active area of machine learning for retinal vessel segmentation has recently been reviewed by Mookiah et al.[24] who also discussed wider issues of performance assessment and comparison. Li et al.[25] have comprehensively reviewed specifically deep learning approaches to fundus image analysis. Readers are referred to these two review papers for a more in-depth survey of state-of-the-art work.

1.1.2 Pre-processing Methods for Retinal Images

It is widely accepted that some form of pre-processing can improve the performance of automated segmentation methods. A quantified justification for the use of these methods, however, is often lacking. Pre-processing of retinal images has two general goals: correcting intensity inhomogeneity, and vessel enhancement. It is common for any pre-processing steps to be preceded by green channel extraction since this exhibits better contrast between vessels and the background which is thought to give better overall segmentation performance. A number of different pre-processing methods can be found in the vessel segmentation literature.

Contrast limited adaptive histogram equalization (CLAHE) is a common pre-processing step for both supervised and unsupervised methods for vessel enhancement[26]. CLAHE is claimed to allow the enhancement of local contrast while not amplifying noise in homogeneous regions of the image[27]. CLAHE operates in small regions of the image called *tiles* where each tile is enhanced by roughly matching the histogram of the output region to a specified histogram distribution (uniform in this case). Neighboring tiles are then combined using bilinear interpolation. The contrast can be limited in regions to avoid the amplification of noise.

Liskowski and Krawiec[17] processed each patch by global contrast normalization (GCN) that was claimed to help the learning process abstract from fluctuations in brightness between patches (i.e. intensity inhomogeneity correction). GCN involves subtracting the mean pixel intensity from each pixel's intensity and dividing by the standard deviation of pixel intensities. This was followed by zero-phase component analysis (ZCA) whitening. ZCA aims to remove any universal correlations between neighboring pixels within the image patch thus allowing the network to concentrate on higher-order correlations.

In terms of intensity inhomogeneity correction, both luminosity and contrast normalization are recognized to improve the performance of automated analysis techniques[28]. Kaba et al.[29] attempted to remove intensity inhomogeneity by applying the N4 bias field correction algorithm[30] to green channel images. N4 correction is primarily used in magnetic resonance imaging (MRI) to deal with uneven illumination caused by the placement of receiver coils, and although relatively unexplored in the field of retinal image analysis, appears well-suited to the task. The algorithm uses a multiscale optimization approach to compute a bias field, which is then subtracted from the original image.

1.1.3 Summary of Previously Published Results

The performance metrics commonly found in the literature to evaluate automated vessel segmentation methods are: sensitivity (Se), specificity (Sp), accuracy (Acc), and area under the receiver operating characteristic (ROC) curve (AUC)[31]; the first three measures are defined in (1).

$$\begin{aligned} Se &= \frac{TP}{TP+FN} \\ Sp &= \frac{TN}{TN+FP} \\ Acc &= \frac{TP+TN}{TP+TN+FP+FN} \end{aligned} \tag{1}$$

where TP = numbers of true positives, TN = numbers of true negatives, FP = numbers of false positive, and FN = numbers of false negative. The perfect classifier has a value of unity for each of these metrics. The comparative values for leading existing methods in each class are summarized in Table 1 (although it should be noted that accuracy has some limitations as a performance metric due to its dependence on prevalence).

From reviewing the literature, it is evident that deep learning approaches have outperformed conventional methods in recent years (with supervised methods generally outperforming unsupervised methods). Higher accuracy and sensitivity (true positive rate) scores from deep learning based methods suggest they are better at labeling vessels within the retinal image than conventional methods. The higher AUC scores also suggest deep learning models have a greater degree of confidence when deciding which pixels are vessels and which pixels are background due to the implied lower false positive rate.

In terms of validating results, a further limitation of methods presented in the literature is the lack of k-fold validation. The convention in the retinal segmentation field has largely been to quote a single performance measure over fixed, prescribed test images, compare this with previously-published values of the same metrics, and claim superiority if those metrics are bettered. Ultimately, the algorithms are being assessed on sampled, noisy data, and so the question arises whether the reported differences are due to chance or due to genuine superiority of a proposed method[24]. This is especially evident when papers are evaluating methods on the commonly-used DRIVE retinal dataset[24]. Authors may be wary of carrying out cross-validation as it could lead to lower performance measures and make their work seem inferior to previously published results. The reporting of single performance measures does, however, carry the danger of methods becoming over-optimized to specific pathologies contained in the pre-defined train/test split – in fact, producing a more trustworthy performance measure is the very motivation for cross validation in the first place. Maier-Hein et al.[32] discuss such issues in connection with the instabilities of ranking in biomedical image analysis competitions.

1.1.4 Research Issues

We are yet to see deep learning methods explore more sophisticated approaches for removing uneven background illumination. In particular, there is scope to evaluate the effectiveness of luminosity and contrast normalization techniques as pre-processing steps for deep-learning-based automated segmentation methods. Consequently, in this paper we systematically investigate the impact of a number of image pre-processing techniques and their combinations on the overall performance of deep learning based retinal vessel segmentation. Starting with the green channel of the respective color images since this exhibits the greatest vessel/background contrast, we have considered four different pre-processing approaches that have previously been applied in the literature, but use of these methods have not been quantitatively justified in the papers that introduced them.

We have evaluated each of the approaches using the three publicly available retinal image datasets in Section 2. Due to the aforementioned issues with limited cross-validation in the literature, as well as using the pre-defined, popular train/test splits for the datasets, we have extended our assessment to a cross-validation approach in an effort to get a more representative measure of generalization. A subsidiary question—that we can only partially address here—is whether the prescribed and widely-used training/testing splits in datasets have unduly influenced progress in

Table 1 Notable segmentation results published in the literature for different conventional supervised methods, deep learning based methods, and unsupervised methods.

Method	DRIVE				STARE				CHASE_DB1			
	Se	Sp	Acc	AUC	Se	Sp	Acc	AUC	Se	Sp	Acc	AUC
2nd Human Observer	0.7760	0.9724	0.9472	–	0.8952	0.9384	0.9349	–	0.8105	0.9711	0.9545	–
Supervised methods: Conventional												
Soares et al.[33]	0.7332	0.9782	0.9466	0.9614	0.7207	0.9747	0.9480	0.9671	–	–	–	–
Fraz et al.[11]	0.7406	0.9807	0.9480	0.9747	0.7548	0.9763	0.9534	0.9768	0.7224	0.9711	0.9469	0.9712
Kaba et al.[29]	0.7466	–	0.9410	–	0.7619	–	0.9456	–	–	–	–	–
Roychowdhury et al.[34]	0.7250	0.9830	0.9520	0.9620	0.7720	0.9730	0.9510	0.9690	0.7201	0.9824	0.9530	0.9532
Supervised methods: Deep learning												
Liskowski & Krawiec et al.[17]	0.8149	0.9749	0.9535	0.9790	0.7867	0.9754	0.9566	0.9785	–	–	–	–
Oliveira et al.[21]	0.8039	0.9803	0.9576	0.9821	0.8315	0.9858	0.9694	0.9905	0.7779	0.9864	0.9653	0.9855
Yan et al.[22]	0.7653	0.9818	0.9542	0.9752	0.7581	0.9846	0.9612	0.9801	0.7633	0.9809	0.9610	0.9781
Cherukuri et al.[23]	0.8425	0.9849	0.9723	0.9870	0.8664	0.9895	0.9803	0.9935	0.8017	0.9908	0.9788	0.9864
Unsupervised methods												
Azzopardi et al.[26]	0.7655	0.9704	0.9442	0.9614	0.7716	0.9701	0.9497	0.9563	0.7585	0.9587	0.9387	0.9487
Zhang et al.[35]	0.7473	0.9764	0.9474	0.9517	0.7676	0.9764	0.9546	0.9614	0.7562	0.9675	0.9457	0.9565

the area together with investigators' desire to report ever 'better' performance metrics? Similar concerns have existed in the mainstream machine learning literature over the near-universal use of common benchmark test datasets[36].

Further, the performance metrics typically employed are all upper-bounded by unity, and these have improved to the point where for many, this limit is being approached quite closely. In this situation any reported improvements are likely to be small and, despite being numerically favorable, the performance difference may well so minor as to be of no practical importance. Hence the 'effect size'—a measure of difference that is of practical importance—is a key adjunct to statistical testing[37]; it is entirely possible for a performance difference to be significant in a statistical test but to be so small to be of no practical value. See also Mookiah et al.[24]. Traditionally, null hypothesis statistical testing (NHST)[38] has been employed to gauge statistical significance between competing methods although over the years NHST has been the subject of much criticism. In addition, NHST procedures do not readily lend themselves to assessing effect sizes. Consequently, in this work we have employed the Bayesian testing procedures proposed of Benavoli et al.[39] that directly incorporate an assessment of effect size. A Bayesian method is advantageous since it directly estimates the quantity of interest—the probability of difference between two methods conditioned on the data—as opposed to NHST which estimates the probability of observing the data given the presumption of no difference (a null hypothesis).

Considering two methods $\mathcal{A}$ and $\mathcal{B}$, and some performance measure S, we can compare the differences $\Delta = S_{\mathcal{A}} - S_{\mathcal{B}}$. The Bayesian signed rank test[39] defines a *region of practical equivalence* (ROPE) for which a difference in performance $|\Delta|$ is so small as to be of no real significance; we have used the ROPE of $\pm 1\%$ (i.e. $\Delta \in [-0.01 \ldots + 0.01]$), for all performance metrics, as suggested by[39] since this appears reasonable although this embodies some element of judgment. The probability that method $\mathcal{A}$ is *worse* than $\mathcal{B}$ (i.e. $S_{\mathcal{A}} < S_{\mathcal{B}}$ overall) can be estimated by integrating the posterior probability density function (PPDF) of Δ between $-\infty$ and -0.01. Similarly, the probability that $\mathcal{A}$ is *better* than $\mathcal{B}$ is obtained by integrating the PPDF of Δ between $+0.01$ and $+\infty$. See Benavoli et al.[39] for further details and discussion.

In this work we have made multiple comparisons between different treatments, a subject that has attracted much attention in the frequentist statistics literature—see, for example, Midway et al.[40]. The basis of a conventional NHST is to make a decision to reject or not-reject a null hypothesis based on a chosen significance level, usually denoted as α, which is the long-term probability of erroneously rejecting the null hypothesis, a so-called type I error. Making multiple such pairwise comparisons inflates the probability of type I error over the family of tests, and in order to control this, it is usual to employ a modification to α, such as the Bonferroni correction[41]. The Bayesian hypothesis testing procedure, on the other hand, estimates the parameters of the posterior distribution of interest[42] rather than making a binary decision about a null hypothesis. Consequently, there is no concept of type I error in a Bayesian test, and thus no notion of having to control type I error over multiple tests. The outcome of the test is a series of three probability values, one for each outcome: $S_{\mathcal{A}} < S_{\mathcal{B}}$, $S_{\mathcal{A}} \approx S_{\mathcal{B}}$, and $S_{\mathcal{A}} > S_{\mathcal{B}}$. The investigator is then able to directly interpret these probability values. Again, see Benavoli et al.[39] for a more detailed discussion.

In the remainder of the paper, we describe our methods in Section 3, and the results obtained in Section 4; this section also contains extensive statistical comparisons. Section 5 analyses the results, and offers some suggestions for future work. Section 7 concludes the paper.

2 Data

Here, we focus exclusively on fundus images of the rear of the eye. There are several publicly available datasets for analyzing fundus images that are commonly used to evaluate the performance of retinal vessel segmentation methods: DRIVE[43], STARE[44], and CHASE_DB1[45]. DRIVE consists of 40 images, each with a resolution of 565 × 584 pixels. Twenty training images and twenty testing images are predefined in the dataset; methods in the literature typically use these partitions when evaluating methods. STARE consists of 20 images with a resolution of 700 × 605 pixels; leave-one-out cross validation is typically used for performance evaluation. CHASE_DB1 comprises 28 images with a resolution of 999 × 960 pixels (14 pairs of images captured from different children); the first 20 images are often taken as the training set, with the remaining 8 being used for testing.

Manual vessel segmentations are provided by two experts for each dataset. The accepted 'gold standard' is to take the first expert's labeling as the ground truth, and human level performance is often claimed when an automated method achieves segmentation results similar to those of the second human observer. These datasets are well studied and have been used as a benchmark to assess automated vessel segmentation methods in most retinal vessel segmentation papers over the past decade.

3 Method

Having taken inspiration from the literature, we implemented a fully convolutional neural network as the basis for our segmentation method. Further details about the architecture can be found in Section 3.2. Before training networks, images were processed using one of four methods, as described in Section 3.1. Images were split into training and testing sets either by adopting the commonly-used train/test splits or in such a way to enable 5-fold cross validation with disjoint training and testing sets. Patches were then randomly extracted from images in the training set followed by model training (Section 3.3). The performance measures of: Se, Sp, Acc and AUC—see (1)—were estimated, and, where appropriate, subjected to Bayesian signed rank tests[39].

3.1 Pre-processing and Patch Extraction

Images from each dataset were processed in one of four different ways:

1. Unprocessed green channel—the 'baseline'
2. N4 bias field correction
3. CLAHE processing
4. N4 bias field correction followed by CLAHE

The first method simply extracted the green channel from color RGB retinal images; each of the other pre-processing methods were applied only to green channel images. We used the ITK implementation of the N4 correction algorithm (https://itk.org/) with 5 scales and a maximum of 25 iterations per resolution level. We found these parameters to perform well at removing uneven background intensity without smoothing the image too much and removing all contrast. Our implementation of CLAHE was taken from the CV2 library (https://opencv.org/).

We used a tile size of 8 × 8 (the default size in CV2), and set the threshold for contrast limiting at 1.5 (slightly lower than CV2's default limit of 2) after a systematic search of the parameter space as, from visual inspection, this seemed to result in marginally less noise in the processed image.

After pre-processing, image intensity values were normalized between $[0 \ldots 1]$ for inputting to the convolutional neural network (CNN). Examples of the application of each method can be seen in Figures 1 and 2. The unprocessed green channel image can be seen in the first column, followed by N4 correction, CLAHE, and N4 + CLAHE, respectively.

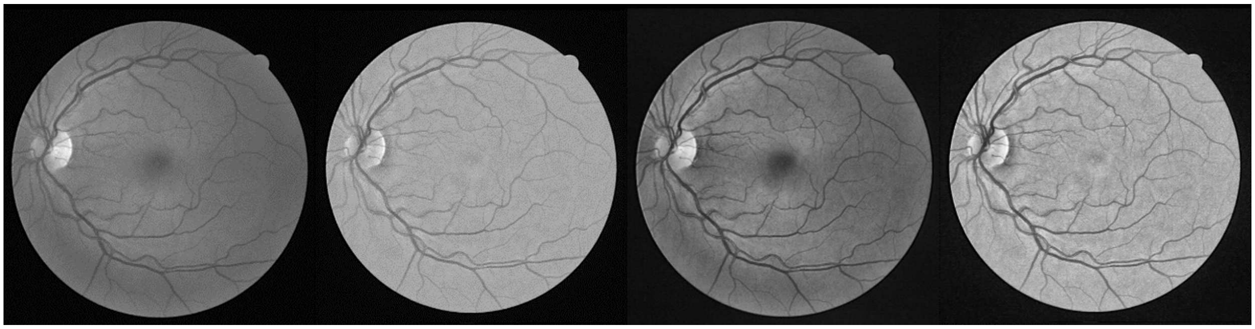

Fig 1 An example of pre-processing methods for an image with typical inhomogeneous background illumination from the DRIVE dataset. (a) Unprocessed image. (b) N4 bias field correction. (c) CLAHE. (d) N4 + CLAHE.

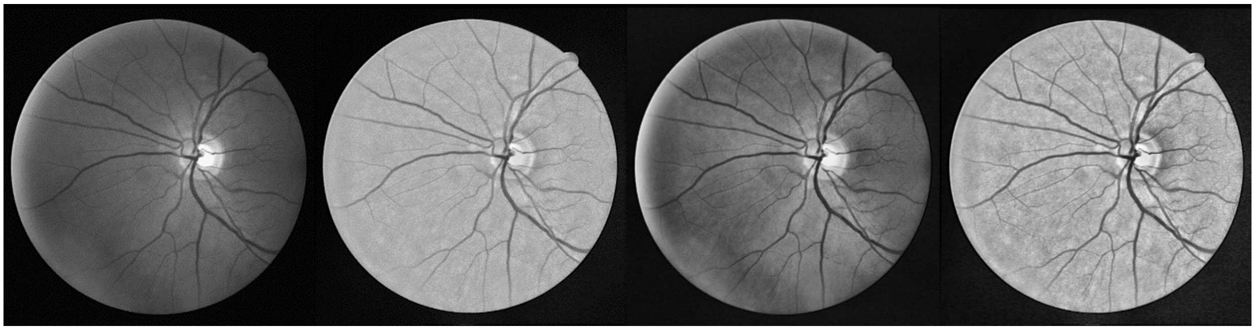

Fig 2 An example of pre-processing methods for an image with more noticeably inhomogeneous background illumination from the CHASE_DB1 dataset. (a) Unprocessed image. (b) N4 bias field correction. (c) CLAHE. (d) N4 + CLAHE.

After all pre-processing, 3,000 patches were randomly extracted from each training set image. Patches were composited into new images allowing a 3,000-fold expansion in the size of the training set, this dataset augmentation producing much larger training datasets with increased variability. For example, for the DRIVE dataset this increased the original 20 images to 20 × 3,000 = 60,000 training images. As suggested in Yan et al.[22], we used a patch size of 128 × 128 pixels. Patches were extracted from within and outside the image's FOV. For each method, patches were extracted from the same random set of locations in each of the four processed images to eliminate a source of possible variability. This allows for a more direct comparison between pre-processing methods.

3.2 Network Architecture

The network architecture used in this work is shown in Figure 3. Our design is based on U-Net, but some modifications were made. Firstly, the depth of the network was reduced by removing max pooling and convolutional layers. We used much smaller input images than in the original paper, thus there was no need for the original number of layers. Additional max pooling layers would have further decreased the resolution of feature maps deeper in the network; very low resolution feature maps are less likely to contribute a meaningful representation. We experimented with increasing the depth of the network, and it had no effect on the segmentation performance. Our implementation was also more computationally efficient due to a reduction of the total number of weights. In addition to changing the depth, we doubled the number of convolutional layers prior to max pooling, as suggested in Szegedy et al.[46] Batch normalization (BN) was also incorporated after the convolutional layers to improve performance and stability during training[47].

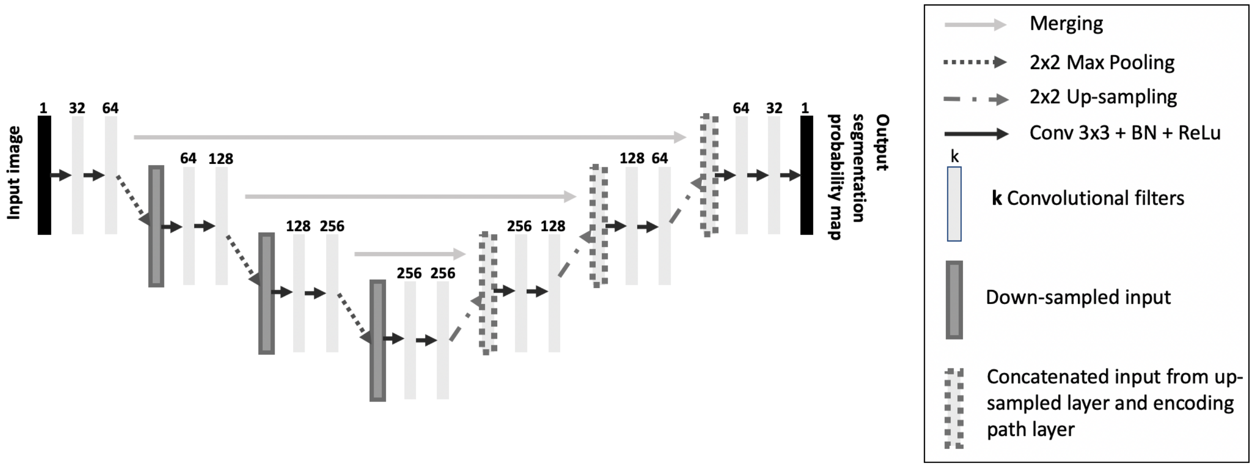

Fig 3 Diagram of deep learning network architecture and key.

Rectified linear unit (ReLU) activation functions follow the BN layers. At present, ReLU functions are widely considered to be the best non-linear function to use in deep neural networks as their constant gradients allow for faster learning in networks with many layers.[13] A sigmoidal function was used at the output layer to map values between 0 and 1, and so approximate the posterior probabilities of each pixel being a blood vessel.

3.3 Training

Before training, the weights in the network were randomly initialized with values drawn from a uniform distribution in the range $\pm 6 \,/\sqrt{n_j + n_{j+1}}$, where n_j is the number units in the j-th layer of the network, since this initialization tends to approximately preserve the variance of the back-propagated gradients and so improve learning[48]. The same pre-initialized network was used as a starting point each time a new network was trained allowing for robust comparisons between methods; trajectories of randomly initialized weights tend to explore markedly different modes in function space.[49] Data augmentation and random shuffling were performed on-the-fly to reduce overfitting. Data augmentation included flips in the horizontal and vertical axes, and random 5-degree rotations. We employed seeded randomness to ensure the order and orientation in which patches were being fed into the network was identical for each pre-processing method. During training, 10% of the training set was isolated for validation (the same group of patch locations were used as the validation set for each pre-processing method). Each network was trained for 10 epochs with a batch size of 25, and a learning rate of 0.005. If the validation loss plateaued or began to increase, the training process was automatically stopped.

Adaptive moment estimation (ADAM)[50] was used for gradient-based optimization with binary cross-entropy (BCE) (2) as the loss function:

$$BCE = \frac{-1}{N} \sum_{i=0}^{N} y_i \times \log(\hat{y}_i) + (1 - y_i) \times \log(1 - \hat{y}_i) \tag{2}$$

where y is the label (1 for vessel and 0 for background), and $\hat{y}$ is the estimated probability of a pixel being a vessel, and N is the number of pixels in the segmentation map produced by the network. The method was implemented using Keras (https://keras.io/) with Tensorflow as the backend. The maximum amount of time a model took to train was around 4 hours when using an Nvidia Titan RTX GPU. The inference time during testing was less than 2 seconds per image.

3.4 Datasets and Testing

We evaluated our method using the three datasets: DRIVE, STARE, and CHASE_DB1 described in Section 2. To compare our method with those found in the literature, we initially used the common train/test splits detailed above. We then performed 5-fold cross validation on the DRIVE and CHASE_DB1 datasets.

To evaluate our method, we used the four performance metrics commonly employed in the retinal segmentation literature: Se, Sp, Acc, and AUC, as defined in (1). Only pixels within the field of view (FOV) were considered. FOV masks are publicly available for the DRIVE dataset, and we manually created binary FOV masks for STARE and CHASE_DB1; simple thresholding was used since the boundary between the retinas and background is obvious. Otsu's method may, however, be more appropriate given a larger dataset[51].

During inference, each testing image was split into non-overlapping patches of 128 × 128 pixels. Zero-padding was used to ensure the image dimensions were integer multiples of the input patch size. Output patches were combined after inference to construct a full resolution vessel probability map. Final binary segmentation maps were obtained by thresholding the probabilities at 0.5, a threshold that selects the class label with the largest posterior probability under the assumption of equal misclassification costs.

3.5 Statistical Testing

Due to shortcomings of NHST, we have used the Bayesian signed rank test proposed in[39] to compare results. We have used the Python implementations of this test available at http://alessiobenavoli.com/research/bayesian-hypothesis-testing-in-machine-learning/.

4 Results

4.1 Qualitative Results

To gain some insight into the characteristics and mislabeling modes of classifiers trained with different image pre-processing methods, it is useful to *qualitatively* scrutinize representative probability and segmentation maps of images in testing sets. Figures 4 and 5 show typical images from testing sets in the DRIVE and CHASE_DB1 datasets, respectively.

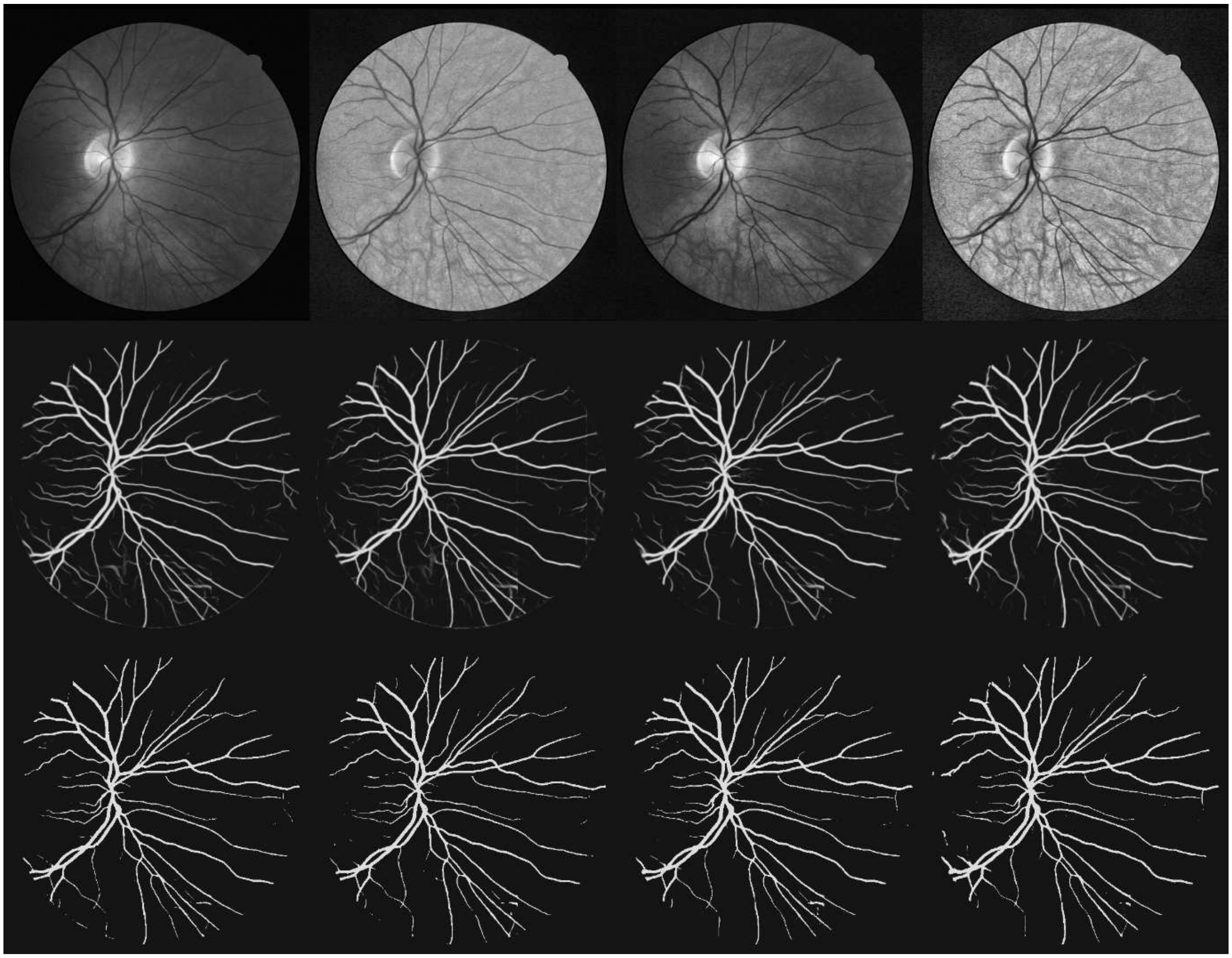

Fig 4 Automated segmentation example from the DRIVE dataset. The first row shows each pre-processing method: unprocessed, N4, CLAHE, and N4 + CLAHE, respectively. The second row shows the probability map output for each pre-processing method. The third row shows the final segmentation map (after thresholding) for each pre-processing method.

After close study of Figure 4, a typical result from the DRIVE dataset, the following observations can be made:

- It appears that the network can distinguish between retina and background better after CLAHE has been applied. Numerous false positives can be seen around the border region in the unprocessed images.
- The proposed method does a good job of distinguishing between vessels and other pathologies present in the image. A combination of background correction and increased contrast seems to help with this separation; this is particularly evident in the bottom left of the retina.
- The network struggles to find some of the smaller vessels in the image, irrespective of which pre-processing method was used.

Additional observations can be made after studying Figure 5, a typical set of images from the CHASE_DB1 dataset:

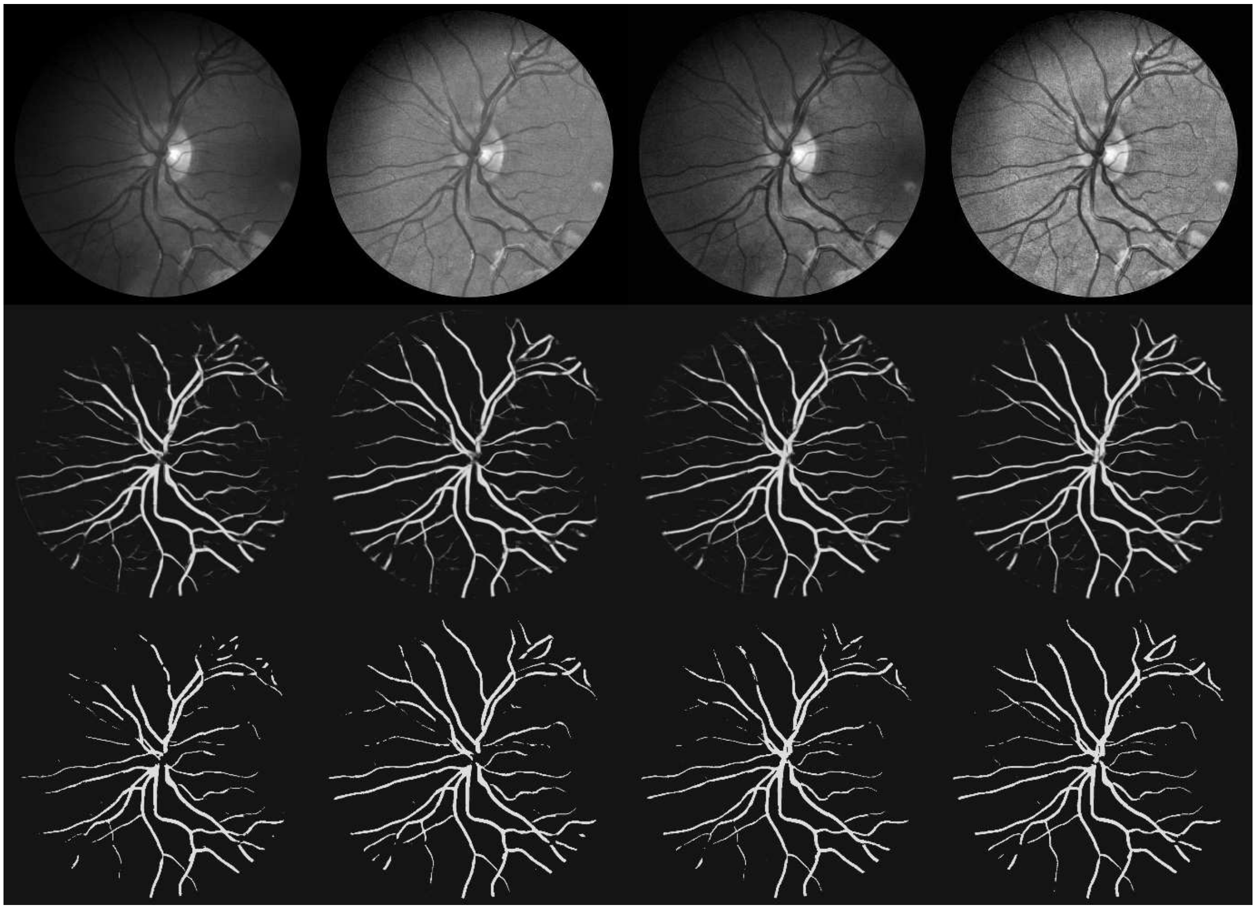

Fig 5 Automated segmentation example from the CHASE_DB1 dataset. The first row shows each pre-processing method: unprocessed, N4, CLAHE, and N4 + CLAHE, respectively. The second row shows the probability map output for each pre-processing method. The third row shows the final segmentation map (after thresholding) for each pre-processing method.

- The network finds it difficult to locate vessels in darker areas of the image. More vessels are detected in the top left corner of the retina after N4 bias field correction than in the unprocessed image.
- CLAHE can help segment vessels within the optical disk.
- N4 bias field correction does not do a perfect job when faced with extremely uneven background illumination. It is difficult to find a balance of parameters that suit all images in the dataset.
- The classifier again struggles to label the finer vessels in the image.

4.2 Quantitative Results

4.2.1 Prescribed Training/Testing Splits

Table 2 shows quantitative results for each pre-processing method using the commonly-adopted training/testing splits for DRIVE, STARE, and CHASE_DB1.

4.2.2 Cross Validation Results

Table 3 shows results for each pre-processing method when using 5-fold cross-validation. Results show better-than-human level performance for accuracy and specificity. A very slight change can be seen between pre-processing methods on the DRIVE dataset; there is a maximum increase in sensitivity of 1.8% between methods. The same can be seen for CHASE_DB1, however, the maximum increase in sensitivity is larger at 7.4%.

4.3 Statistical Comparisons

To conduct statistical comparisons of our results, we have used the Bayesian signed rank test described in Section 3.5. Tables 4–6 show statistical comparisons between pre-processing methods for each dataset when using the common train/test splits. Tables 7 and 8 show statistical comparisons between pre-processing after 5-fold cross validation; the commonly-used training/testing arrangement for STARE is leave-one-out (i.e. N-fold cross validation) hence reference should also be made to Table 5 for comparing the STARE cross-validation results.

The results in Tables 4 to 8 are a set of pairwise comparisons in which a classifier (with one pre-processing method) shown on each row is compared with each of the other classifiers (with different pre-processing methods) shown in the corresponding columns of that row. The outcome of each pairwise comparison is a tuple of three numbers shown column-wise and representing, in order, the posterior probabilities that:

- The median difference between the compared classifiers falls in the interval -0.01 to $-\infty$ meaning that the row-classifier performs *worse* than the column-classifier with which it being compared.
- The median difference between the compared classifiers falls in the region of practical equivalence (ROPE) and that the effect size of any difference is of no/little practical significance.
- The median difference between the compared classifiers falls in the interval $+0.01$ to $+\infty$ meaning that the row-classifier performs *better* than the column-classifier with which it is being compared.

These three probabilities are presented in the following tables for each performance measure, and for each pair of compared classifiers. To take an illustrative example from Table 4, the comparison between the sensitivities (Se) of the 'Unprocessed' (row) classifier, and the N4 classifier (column 1), $\Pr(\text{negative difference}) = 0.0$, $\Pr(\text{in ROPE}) = 0.0488$, and $\Pr(\text{positive difference}) = 0.9512$. Hence with 95.12% probability, we conclude that the classifier with N4 pre-processing has a superior Se score to the classifier with no pre-processing. That Bayesian testing allows us to make such a clear statement is in sharp contrast to NHST for which such a statement would be an (all too common) misinterpretation; this added clarity is a notable advantage of the Bayesian approach.

Table 2 Segmentation performance results for each pre-processing method for the DRIVE, STARE and CHASE_DB1 datasets using the commonly-used, prescribed training/testing splits. Bold = best result

Dataset	Method	Se	Sp	Acc	AUC
DRIVE	2nd Human Observer	**0.7760**	0.9724	0.9472	-
	Unprocessed	0.6819	0.9902	0.9508	0.9677
	N4	0.6659	**0.9913**	0.9497	0.9727
	CLAHE	0.7563	0.9842	0.9550	**0.9766**
	CLAHE + N4	0.7708	0.9826	**0.9554**	**0.9766**
STARE	2nd Human Observer	**0.8952**	0.9384	0.9349	-
	Unprocessed	0.7498	0.9876	0.9634	0.9793
	N4	0.7491	**0.9880**	0.9634	0.9796
	CLAHE	0.7720	0.9858	0.9640	0.9802
	N4 + CLAHE	0.7799	0.9853	**0.9642**	**0.9811**
CHASE_DB1	2nd Human Observer	**0.8105**	0.9711	0.9545	-
	Unprocessed	0.7396	**0.9823**	0.9572	0.9716
	N4	0.7867	0.9785	0.9587	0.9758
	CLAHE	0.7836	0.9801	0.9611	0.9774
	N4 + CLAHE	0.7950	0.9771	**0.9625**	**0.9814**

Table 3 Cross validation segmentation performance results for each pre-processing method on DRIVE and CHASE_DB1; the corresponding leave-one-cross-validation results for STARE are shown in Table 2. Bold = best result.

Dataset	Method	Se	Sp	Acc	AUC
DRIVE	2nd Human Observer	**0.7760**	0.9724	0.9472	-
	Unprocessed	0.6887	0.9877	0.9494	0.9703
	N4	0.6844	**0.9887**	0.9497	0.9692
	CLAHE	0.6975	0.9883	0.9508	0.9707
	CLAHE + N4	0.7013	0.9876	**0.9514**	**0.9711**
CHASE_DB1	2nd Human Observer	**0.8105**	0.9711	0.9545	-
	Unprocessed	0.7157	**0.9889**	0.9623	0.9739
	N4	0.7313	0.9887	0.9639	**0.9811**
	CLAHE	0.7649	0.9874	0.9656	0.9798
	CLAHE + N4	0.7688	0.9867	**0.9657**	0.9803

Table 4 Bayesian signed rank test evaluation of results from the DRIVE dataset after using the given train/test split. Each pre-processing method is compared to another in a paired fashion. The numbers in each cell (from the top, downwards) represent $P(\Delta \leq 1\%)$, $P(-1\% < \Delta < 1\%)$, and $P(\Delta \geq 1\%)$, respectively.

	N4				CLAHE				N4 + CLAHE			
	Se	Sp	Acc	AUC	Se	Sp	Acc	AUC	Se	Sp	Acc	AUC
	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.1246	0.0	0.0	0.0	0.1304
Unprocessed	0.0488	1.0	1.0	1.0	0.0	1.0	1.0	0.8754	1.0	1.0	1.0	0.8696
	0. 9512	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
					1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0001
N4					0.0	1.0	1.0	1.0	1.0	0.9752	1.0	0.9999
					0.0	0.0	0.0	0.0	0.0	0.0248	0.0	0.0
									0.9851	0.0	0.0	0.0
CLAHE									0.0149	1.0	1.0	1.0
									0.0	0.0	0.0	0.0

Table 5 Bayesian signed rank test evaluation of results from the STARE dataset. Each pre-processing method is compared to another in a paired fashion. The numbers in each cell (from the top, downwards) represent $P(\Delta \leq 1\%)$, $P(-1\% < \Delta < 1\%)$, and $P(\Delta \geq 1\%)$, respectively.

	N4				CLAHE				N4 + CLAHE			
	Se	Sp	Acc	AUC	Se	Sp	Acc	AUC	Se	Sp	Acc	AUC
Unprocessed	0.4085	0.0	0.0	0.0	0.9927	0.0	0.0	0.0	0.9988	0.0	0.0	0.0
	0.0958	1.0	1.0	1.0	0.0009	1.0	1.0	1.0	0.0012	1.0	1.0	1.0
	0.4957	0.0	0.0	0.0	0.0064	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N4					0.9921	0.0	0.0	0.0	0.9998	0.0	0.0	0.0
					0.0079	1.0	1.0	1.0	0.0002	1.0	1.0	1.0
					0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CLAHE									0.2039	0.0	0.0	0.0008
									0.7783	1.0	1.0	0.9992
									0.0178	0.0	0.0	0.0

Table 6 Bayesian signed rank test evaluation of results from the CHASE_DB1 dataset after using the popular train/test split. Each pre-processing method is compared to another in a paired fashion. The numbers in each cell (from the top, downwards) represent $P(\Delta \leq 1\%)$, $P(-1\% < \Delta < 1\%)$, and $P(\Delta \geq 1\%)$, respectively.

	N4				CLAHE				N4 + CLAHE			
	Se	*Sp*	*Acc*	AUC	*Se*	*Sp*	*Acc*	AUC	*Se*	*Sp*	*Acc*	AUC
Unprocessed	0.9998 0.0002 0.0	0.0 1.0 0.0	0.0 1.0 0.0	0.0015 0.9985 0.0	1.0 0.0 0.0	0.0 1.0 0.0	0.0 1.0 0.0	0.0115 0.9885 0.0	0.9960 0.0040 0.0	0.0 1.0 0.0	0.0 1.0 0.0	0.2354 0.7646 0.0
N4					0.0790 0.4663 0.4547	0.0 1.0 0.0	0.0 1.0 0.0	0.0001 0.9999 0.0	0.3702 0.6296 0.0	0.0 1.0 0.0	0.0 1.0 0.0	0.0020 0.9980 0.0
CLAHE									0.5885 0.4113 0.0002	0.0 1.0 0.0	0.0 1.0 0.0	0.0 1.0 0.0

Table 7 Bayesian signed rank test evaluation of results from the DRIVE dataset after using 5-fold cross validation. Each pre-processing method is compared to another in a paired fashion. The numbers in each cell (from the top, downwards) represent $P(\Delta \leq 1\%)$, $P(-1\% < \Delta < 1\%)$, and $P(\Delta \geq 1\%)$, respectively.

	N4				CLAHE				N4 + CLAHE			
	Se	Sp	Acc	AUC	Se	Sp	Acc	AUC	Se	Sp	Acc	AUC
Unprocessed	0.0380 0.0 0.9620	0.0 1.0 0.0	0.0 1.0 0.0	0.0 1.0 0.0	0.8572 0.0803 0.0625	0.0 1.0 0.0	0.0 1.0 0.0	0.0 1.0 0.0	0.8716 0.0940 0.0344	0.0 1.0 0.0	0.0 1.0 0.0	0.0 1.0 0.0
N4					0.9998 0.0002 0.0	0.0 1.0 0.0	0.0 1.0 0.0	0.0 1.0 0.0	1.0 0.0 0.0	0.0 1.0 0.0	0.0 1.0 0.0	0.0 1.0 0.0
CLAHE									0.1037 0.8788 0.0175	0.0 1.0 0.0	0.0 1.0 0.0	0.0 1.0 0.0

Table 8 Bayesian signed rank test evaluation of results from the CHASE_DB1 dataset after using 5-fold cross validation. Each pre-processing method is compared to another in a paired fashion. The numbers in each cell (from the top, downwards) represent $P(\Delta \leq 1\%)$, $P(-1\% < \Delta < 1\%)$, and $P(\Delta \geq 1\%)$, respectively.

	N4				CLAHE				N4 + CLAHE			
	Se	Sp	Acc	AUC	Se	Sp	Acc	AUC	Se	Sp	Acc	AUC
Unprocessed	0.8350	0.0	0.0	0.0065	1.0	0.0	0.0	0.0005	1.0	0.0	0.0	0.0004
	0.1611	1.0	1.0	0.9935	0.0	1.0	1.0	0.9995	0.0	1.0	1.0	0.9996
	0.0038	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
N4					0.9948	0.0	0.0	0.0	1.0	0.0	0.0	0.0
					0.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0
					0.0052	0.0	0.0	0.0	0.0	0.0	0.0	0.0
CLAHE									0.5508	0.0	0.0	0.0
									0.3765	1.0	1.0	1.0
									0.0728	0.0	0.0	0.0

5 Discussion

5.1 Comparison with the State-of-the-art

When looking at results from the commonly-used training/testing splits, our proposed method in combination with N4 bias field correction, and CLAHE outperforms all 'classical' supervised and unsupervised vessel segmentation methods (Table 1). We also achieved consistently better results than the deep learning method proposed in[22], and mostly better results than those of Liskowski and Krawiec[17]. The deep learning method proposed by Oliveira et al.[21] shows marginally better performance where the differences in accuracies between the two methods over the DRIVE, STARE, and CHASE_DB1 datasets are 0.23%, 0.54%, and 0.29%, respectively. As pointed out by Mookiah et al.[24], since the performance measures are upper-bounded by unity and that limit is being quite closely approached, it is not clear whether the differences above represent an effect size of any significance. Regardless, the scope of the present paper is to investigate the effects of various background pre-processing steps rather than achieve better than state-of-the-art performance, and our architecture fully meets the requirement of being competitive.

After 5-fold cross validation (CV), a decrease in performance can be seen in DRIVE (most noticeably, in sensitivity) suggesting results presented in the literature could be unrepresentative of the overall dataset, and may be over-optimized to the given train/test split. A similar trend can be seen in CHASE_DB1 in terms of a reduction in sensitivity. There are, however, some noticeable increases in specificity and accuracy for the CV results.

Comparing the relative performances over the prescribed splits (Table 2) with the corresponding results for cross validation (Table 3), sixteen of the measures increase in value with CV whereas thirteen are lower, with one unchanged. We regard the CV results as more representative of generalization performance because they effectively average over the variability in the datasets.

It is difficult to conclusively say one architecture/method is better than another without performing comparative statistical validation. The small differences could be explained by chance, or represent no practical difference. We suggest that all future biomedical segmentation papers should routinely include these types of statistical tests as opposed to simply stating that the performance measure for one method is "better" than another without any statistical justification, or any assessment of the practical differences (i.e. effect size). Indeed, such a shift was initiated in the mainstream computer vision field in the early 1990s[52].

5.2 The Impact of Pre-processing on Segmentation Performance

For the prescribed testing partitions (or the prescribed leave-one-out assessment in the case of STARE)—see Table 2—the images processed with N4 bias field correction followed by CLAHE consistently yielded the highest accuracy and AUC measures. When compared to the unprocessed images, there was a slight improvement in accuracies of 0.48%, 0.08% and 0.55% for DRIVE, STARE and CHASE_DB1, respectively. N4 processing alone yielded the highest Sp measures for DRIVE and STARE although, interestingly, the highest Sp for the CHASE_DB1 was produced by the unprocessed images. All our methods, irrespective of the pre-processing used, exceed second observer performance on accuracy and specificity. Sensitivity, however, is consistently higher for the second human observer across all datasets. This suggests a human decision maker is less prone to false negatives, possibly due to better detection of finer vessels and/or vessel boundaries by taking contextual cues into account. Of the automated methods, the best performance was obtained with N4 + CLAHE.

The results obtained suggest pre-processing can make a difference to segmentation performance. If we consider the results from 5-fold cross validation for DRIVE and CHASE_DB1 (and leave-one-out validation of STARE) as being the most representative, statistical evaluation shows significant changes in the sensitivity metric (Se) over different pre-processing methods. Statistical evaluation also suggests there is no significant difference in accuracy, specificity, and AUC thus pre-processing methods can be considered practically equivalent regarding those three metrics. There is arguably a case for reducing the region of practical equivalence as we are dealing with very small margins of improvement in measures already above 95%. This, however, is rather subjective.

Another noteworthy feature of the results in Table 4–8 is that in many cases, the $\Pr(\text{in ROPE})$ is unity (to 4 d.p.) indicating that all the probability mass of the posterior falls within the ROPE interval implying no practical difference in these measures with (almost) complete certainty.

CLAHE pre-processing appears to be the most impactful method in terms of increasing the sensitivity, which indicates that more vessels are being successfully detected. Better contrast may allow the network to more clearly identify separation between vessel boundaries and background allowing for more complete segmentation. When it came to finding the very fine vessels in the images, CLAHE did not seem to make much difference, which may be due to the inherent low-pass filtering present in the U-Net architecture (as a result of the maxpooling and upsampling operations). Further experiments are needed to investigate the problem of very fine vessel detection.

Inhomogeneous background illumination correction with the N4 algorithm added some benefit when combined with CLAHE. CHASE_DB1 gained the most from this combination of background correction. Images in the CHASE_DB1 dataset appear to contain a larger amount of variation in background intensity than images in DRIVE and STARE, which could explain why background correction was more effective for CHASE_DB1. It was difficult for the network to find vessels in very dark parts of the image when no background correction had been applied. Standalone N4 correction outperformed unprocessed images in CHASE_DB1, however, unprocessed images performed better in DRIVE, and performance was almost identical between these two methods in STARE. Even when combined with CLAHE, N4 correction did not seem to significantly influence the performance. The probability of the mean differences in sensitivity being practically equivalent was significant when comparing CLAHE and CLAHE + N4.

Finally, a number of authors, for example, Chekuri et al.[23] have reported improved performance with using different loss functions in the network training. We have employed the common binary cross-entropy loss across all comparisons, but it is an interesting area of future work to quantitatively explore the influence of other loss functions on performance.

6 Code, Data, and Materials Statement

The datasets used in this work are publicly available and downloadable from the links given in Section 2. Section 3 provides links to the open source image processing methods used. The pre-processing, deep network and analysis code are all available at https://github.com/pirlite2/spie-jmi.

7 Conclusions

In order to investigate the impact of pre-processing techniques on the segmentation of retinal images, we have presented a deep learning framework for automated vessel segmentation in retinal fundus images that achieves results that are competitive with recent deep learning approaches described in the literature. Contrast limited adaptive histogram equalization (CLAHE) was found to be the most effective pre-processing method. Removal of uneven background intensity using the N4 bias field correction method is only really useful in cases of extreme inhomogeneity.

Further, we have assessed performance not only over the prescribed/commonly-used test sets of images, but also using cross-validation and statistical testing. In particular, we have employed a Bayesian statistical test due to Benavoli et al. which addresses many of the shortcomings with conventional null hypothesis statistical testing. We believe our results highlight the importance of validation methods as we have gained better insight into the influence of pre-processing, and have been able to draw robust, statistically-founded conclusions that would not have been apparent had we just considered results from the commonly-used training/testing splits often seen in the literature.

Disclosures

The authors declare no conflicts of interest, financial or otherwise.

References

1 J. W. Y. Yau, S. L. Rogers, R. Kawasaki, *et al.*, "Global prevalence and major risk factors of diabetic retinopathy," *Diabetes Care* **35**(22301125), 556–564 (2012). 10.2337/dc11-1909.

2 World Health Organization, "Global report on diabetes," tech. rep. (2016).

3 M. D. Abràmoff, M. K. Garvin, and M. Sonka, "Retinal imaging and image analysis," *IEEE Rev. Biomed. Eng.* **3**, 169–208 (2010). 10.1109/rbme.2010.2084567.

4 B. Bowling, *Kanski's Clinical Ophthalmology : A Systematic Approach*, Elsevier, Edinburgh, 8th ed. (2015).

5 B. J. Fenner, R. L. M. Wong, W.-C. Lam, *et al.*, "Advances in retinal imaging and applications in diabetic retinopathy screening: A review," *Ophthalmology and Therapy* **7**, 333–346 (2018). 10.1007/s40123-018-0153-7.

6 C. Köse and C. İki˙baş, "A personal identification system using retinal vasculature in retinal fundus images," *Expert Syst Appl* **38**(11), 13670–13681 (2011). 10.1016/j.eswa.2011.04.141.

7 P. Hamet and J. Tremblay, "Artificial intelligence in medicine," *Metabolism* **69**, S36–S40 (2017). 10.1016/j.metabol.2017.01.011.

8 A. S. Ahuja, "The impact of artificial intelligence in medicine on the future role of the physician," *PeerJ* **7**(31592346), e7702–e7702 (2019).

9 T. Davenport and R. Kalakota, "The potential for artificial intelligence in healthcare," *Future Healthcare Journal* **6**(31363513), 94–98 (2019).

10 P. Vostatek, E. Claridge, H. Uusitalo, *et al.*, "Performance comparison of publicly available retinal blood vessel segmentation methods," *Comput. Med. Imaging Graph.* **55**(Supplement C), 2–12 (2017). 10.1016/j.compmedimag.2016.07.005.

11 M. M. Fraz, P. Remagnino, A. Hoppe, *et al.*, "An ensemble classification-based approach applied to retinal blood vessel segmentation," *IEEE Trans. Biomed. Eng.* **59**, 2538–2548 (2012). 10.1109/tbme.2012.2205687.

12 T. A. Soomro, A. J. Afifi, L. Zheng, *et al.*, "Deep learning models for retinal blood vessels segmentation: A review," *IEEE Access* **7**, 71696–71717 (2019). 10.1109/access.2019.2920616.

13 Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature* **521**(7553), 436–444 (2015). 10.1038/nature14539.

14 G. Litjens, T. Kooi, B. E. Bejnordi, *et al.*, "A survey on deep learning in medical image analysis," *Med. Image Anal.* **42**, 60–88 (2017). 10.1016/j.media.2017.07.005.

15 Y. LeCun, L. D. Jackel, B. Boser, *et al.*, "Handwritten digit recognition: applications of neural network chips and automatic learning," *IEEE Commun. Mag.* **27**(11), 41–46 (1989). 10.1109/35.41400.

16 A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *25th International Conference on Neural Information Processing Systems - Volume 1*, *NIPS'12*, 1097–1105 (2012). 10.1145/3065386.

17 P. Liskowski and K. Krawiec, "Segmenting retinal blood vessels with deep neural networks," *IEEE Trans. Med. Imag.* **35**, 2369–2380 (2016). 10.1109/tmi.2016.2546227.

18 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)*, 234–241 (2015). 10.1007/978-3-319-24574-4_28.

19 J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431–3440, (Boston, MA) (2015). 10.1109/cvpr.2015.7298965.

20 M. Drozdzal, E. Vorontsov, G. Chartrand, *et al.*, "The importance of skip connections in biomedical image segmentation," in *Deep Learning and Data Labeling for Medical Applications*, 179–187 (2016). 10.1007/978-3-319-46976-8_19.

21 A. Oliveira, S. Pereira, and C. A. Silva, "Retinal vessel segmentation based on fully convolutional neural networks," *Expert Syst Appl* **112**, 229–242 (2018).

22 Z. Yan, X. Yang, and K. Cheng, "Joint segment-level and pixel-wise losses for deep learning based retinal vessel segmentation," *IEEE Trans. Biomed. Eng.* **65**(9), 1912–1923 (2018). 10.1109/tbme.2018.2828137.

23 V. Cherukuri, V. Kumar B.G., R. Bala, *et al.*, "Deep retinal image segmentation with regularization under geometric priors," *IEEE Transactions on Image Processing* **29**, 2552–2567 (2020).

24 M. R. K. Mookiah, S. Hogg, T. J. MacGillivray, *et al.*, "A review of machine learning methods for retinal blood vessel segmentation and artery/vein classification," *Medical Image Analysis* **68**, 101905 (2021).

25 T. Li, W. Bo, C. Hu, *et al.*, "Applications of deep learning in fundus images: A review," *Medical Image Analysis* **69**, 101971 (2021).

26 G. Azzopardi, N. Strisciuglio, M. Vento, *et al.*, "Trainable COSFIRE filters for vessel delineation with application to retinal images," *Med. Image Anal.* **19**(1), 46–57 (2015).

27 S. M. Pizer, E. P. Amburn, J. D. Austin, *et al.*, “Adaptive histogram equalization and its variations,” *Computer Vision, Graphics, and Image Processing* **39**(3), 355–368 (1987). 10.1016/s0734-189x(87)80186-x.

28 M. Foracchia, E. Grisan, and A. Ruggeri, “Luminosity and contrast normalization in retinal images,” *Med. Image Anal.* **9**, 179–190 (2005). 10.1016/j.media.2004.07.001.

29 D. Kaba, C. Wang, Y. Li, *et al.*, “Retinal blood vessels extraction using probabilistic modelling,” *Health Information Science and Systems* **2**(1), 1–10 (2014). 10.1186/2047-2501-2-2.

30 N. J. Tustison, B. B. Avants, P. A. Cook, *et al.*, “N4ITK: Improved N3 bias correction,” *IEEE Trans. Med. Imag.* **29**, 1310–1320 (2010). 10.1109/tmi.2010.2046908.

31 A. P. Bradley, “The use of the area under the ROC curve in the evaluation of machine learning algorithms,” *Pattern Recogn* **30**(7), 1145–1159 (1997). 10.1016/s0031-3203(96)00142-2.

32 L. Maier-Hein, M. Eisenmann, A. Reinke, *et al.*, “Why rankings of biomedical image analysis competitions should be interpreted with care,” *Nature Communications* **9**(1), 5217 (2018).

33 J. V. B. Soares, J. J. G. Leandro, R. M. Cesar, *et al.*, “Retinal vessel segmentation using the 2-D Gabor wavelet and supervised classification,” *IEEE Trans. Med. Imag.* **25**, 1214–1222 (2006). 10.1109/tmi.2006.879967.

34 S. Roychowdhury, D. Koozekanani, and K. Parhi, “Blood vessel segmentation of fundus images by major vessel extraction and sub-image classification,” *IEEE J. Biomed. Health Inform.* **19**(3), 1118–1128 (2014). 10.1109/jbhi.2014.2335617.

35 J. Zhang, B. Dashtbozorg, E. Bekkers, *et al.*, “Robust retinal vessel segmentation via locally adaptive derivative frames in orientation scores,” *IEEE Trans. Med. Imag.* **35**(12), 2631–2644 (2016). 10.1109/tmi.2016.2587062.

36 C. Soares, “Is the UCI repository useful for data mining?,” in *11th Portuguese Conference on Artificial Intelligence (EPIA 2003)*, 209–223, (Beja, Portugal) (2003). 10.1007/978-3-540-24580-3_28.

37 J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, Lawrence Erlbaum Associates, Hillsdale, NJ, 2nd ed. (1988).

38 T. H. Wonnacott and R. J. Wonnacott, *Introductory Statistics*, John Wiley & Sons, 5th ed. (1990).

39 A. Benavoli, G. Corani, J. Demšar, *et al.*, “Time for a change: A tutorial for comparing multiple classifiers through Bayesian analysis,” *J Mach Learn Res* **18**(77), 1–36 (2017).

40 S. Midway, M. Robertson, S. Flinn, *et al.*, “Comparing multiple comparisons: practical guidance for choosing the best multiple comparisons test,” *PeerJ* **8**, e10387 (2020).

41 H. Abdi, “The Bonferonni and Šidák corrections for multiple comparisons,” in *Encyclopedia of Measurement and Statistics*, N. J. Salkind and K. Rasmussen, Eds., SAGE, Thousand Oaks, CA (2007).

42 J. K. Kruschke, “Bayesian estimation supersedes the t-test,” *Journal of Experimental Psychology. General* **142**(2), 573–588 (2013).

43 J. Staal, M. D. Abramoff, M. Niemeijer, *et al.*, “Ridge-based vessel segmentation in color images of the retina,” *IEEE Trans. Med. Imag.* **23**, 501–509 (2004). 10.1109/TMI.2004.825627.

44 A. D. Hoover, V. Kouznetsova, and M. Goldbaum, “Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response,” *IEEE Trans. Med. Imag.* **19**, 203–210 (2000). 10.1109/42.845178.

45 C. G. Owen, A. R. Rudnicka, R. Mullen, *et al.*, "Measuring retinal vessel tortuosity in 10-year-old children: Validation of the computer-assisted image analysis of the retina (CA-IAR) program," *Investigative Ophthalmology & Visual Science* **50**, 2004–2010 (2009). 10.1167/iovs.08-3018.

46 C. Szegedy, V. Vanhoucke, S. Ioffe, *et al.*, "Rethinking the inception architecture for computer vision," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818–2826 (2016). 10.1109/cvpr.2016.308.

47 S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *32nd International Conference on International Conference on Machine Learning - Volume 37*, 448–456, (Lille, France) (2015).

48 X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *13th International Conference on Artificial Intelligence and Statistics*, **9**, 249–256, (Chia Laguna Resort, Sardinia, Italy) (2010).

49 S. Fort, H. Hu, and B. Lakshminarayanan, "Deep ensembles: A loss landscape perspective," *CoRR* **abs/1912.02757** (2019).

50 D. P. Kingma and J. Ba, "ADAM: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015*, (San Diego, CA) (2015).

51 N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Sys. Man Cybern.* **9**, 62–66 (1979). 10.1109/TSMC.1979.4310076.

52 R. C. Jain and T. O. Binford, "Ignorance, myopia, and naiveté in computer vision systems," *CVGIP: Image Understanding* **53**(1), 112–117 (1991). 10.1016/1049-9660(91)90009-E.

List of Figures

1 An example of pre-processing methods for an image with typical inhomogeneous background illumination from the DRIVE dataset. (a) Unprocessed image. (b) N4 bias field correction. (c) CLAHE. (d) N4 + CLAHE.

2 An example of pre-processing methods for an image with more noticeably inhomogeneous background illumination from the CHASE_DB1 dataset. (a) Unprocessed image. (b) N4 bias field correction. (c) CLAHE. (d) N4 + CLAHE.

3 Diagram of deep learning network architecture and key.

4 Automated segmentation example from the DRIVE dataset. The first row shows each pre-processing method: unprocessed, N4, CLAHE, and N4 + CLAHE, respectively. The second row shows the probability map output for each pre-processing method. The third row shows the final segmentation map (after thresholding) for each pre-processing method.

5 Automated segmentation example from the CHASE_DB1 dataset. The first row shows each pre-processing method: unprocessed, N4, CLAHE, and N4 + CLAHE, respectively. The second row shows the probability map output for each pre-processing method. The third row shows the final segmentation map (after thresholding) for each pre-processing method.

List of Tables

1 Notable segmentation results published in the literature for different conventional supervised methods, deep learning based methods, and unsupervised methods.

2 Segmentation performance results for each pre-processing method for the DRIVE, STARE and CHASE_DB1 datasets using the commonly-used, prescribed training/testing splits. Bold = best result

3 Cross validation segmentation performance results for each pre-processing method on DRIVE and CHASE_DB1; the corresponding leave-one-cross-validation results for STARE are shown in Table 2. Bold = best result.

4 Bayesian signed rank test evaluation of results from the DRIVE dataset after using the given train/test split. Each pre-processing method is compared to another in a paired fashion. The numbers in each cell (from the top, downwards) represent $P(\Delta \leq 1\%)$, $P(-1\% < \Delta < 1\%)$, and $P(\Delta \geq 1\%)$, respectively.

5 Bayesian signed rank test evaluation of results from the STARE dataset. Each pre-processing method is compared to another in a paired fashion. The numbers in each cell (from the top, downwards) represent $P(\Delta \leq 1\%)$, $P(-1\% < \Delta < 1\%)$, and $P(\Delta \geq 1\%)$, respectively.

6 Bayesian signed rank test evaluation of results from the CHASE_DB1 dataset after using the popular train/test split. Each pre-processing method is compared to another in a paired fashion. The numbers in each cell (from the top, downwards) represent $P(\Delta \leq 1\%)$, $P(-1\% < \Delta < 1\%)$, and $P(\Delta \geq 1\%)$, respectively.

7 Bayesian signed rank test evaluation of results from the DRIVE dataset after using 5-fold cross validation. Each pre-processing method is compared to another in a paired fashion. The numbers in each cell (from the top, downwards) represent $P(\Delta \leq 1\%)$, $P(-1\% < \Delta < 1\%)$, and $P(\Delta \geq 1\%)$, respectively.

8 Bayesian signed rank test evaluation of results from the CHASE_DB1 dataset after using 5-fold cross validation. Each pre-processing method is compared to another in a paired fashion. The numbers in each cell (from the top, downwards) represent $P(\Delta \leq 1\%)$, $P(-1\% < \Delta < 1\%)$, and $P(\Delta \geq 1\%)$, respectively.