applied sciences

MDPI

Article

Deep Learning Architectures for Diagnosis of Diabetic Retinopathy

Alberto Solano [1], Kevin N. Dietrich [1], Marcelino Martínez-Sober [1], Regino Barranquero-Cardeñosa [1], Jorge Vila-Tomás [2,*] and Pablo Hernández-Cámara [2,*]

[1] Intelligent Data Analysis Laboratory, ETSE (Engineering School), Universitat de València, 46100 Burjassot, Spain; alsoca2@alumni.uv.es (A.S.); kenidie@alumni.uv.es (K.N.D.); marcelino.martinez@uv.es (M.M.-S.); regino.barranquero@uv.es (R.B.-C.)

[2] Image Processing Lab., Universitat de València, 46980 Paterna, Spain

[*] Correspondence: jorge.vila-tomas@uv.es (J.V.-T.); pablo.hernandez-camara@uv.es (P.H.-C.)

Abstract: For many years, convolutional neural networks dominated the field of computer vision, not least in the medical field, where problems such as image segmentation were addressed by such networks as the U-Net. The arrival of self-attention-based networks to the field of computer vision through ViTs seems to have changed the trend of using standard convolutions. Throughout this work, we apply different architectures such as U-Net, ViTs and ConvMixer, to compare their performance on a medical semantic segmentation problem. All the models have been trained from scratch on the DRIVE dataset and evaluated on their private counterparts to assess which of the models performed better in the segmentation problem. Our major contribution is showing that the best-performing model (ConvMixer) is the one that shares the approach from the ViT (processing images as patches) while maintaining the foundational blocks (convolutions) from the U-Net. This mixture does not only produce better results ($DICE = 0.83$) than both ViTs (0.80/0.077 for UNETR/SWIN-Unet) and the U-Net (0.82) on their own but reduces considerably the number of parameters (2.97M against 104M/27M and 31M, respectively), showing that there is no need to systematically use large models for solving image problems where smaller architectures with the optimal pieces can get better results.

Keywords: segmentation; medical image; ConvMixer; U-Net; vision transformer

check for updates

Citation: Solano, A.; Dietrich, K.N.; Martínez-Sober, M.; Barranquero-Cardeñosa, R.; Vila-Tomás, J.; Hernández-Cámara, P. Deep Learning Architectures for Diagnosis of Diabetic Retinopathy. *Appl. Sci.* **2023**, *13*, 4445. https://doi.org/10.3390/app13074445

Academic Editors: Yu-Dong Zhang and Jan Egger

Received: 1 February 2023
Revised: 28 March 2023
Accepted: 29 March 2023
Published: 31 March 2023

1. Introduction

Image segmentation is one of the main tasks in the field of computer vision and image analysis. It consists of dividing an image into several regions, each of which corresponds to a different object or background. To do this, each pixel is assigned a label which it shares with pixels of similar characteristics.

Although there are numerous applications of image segmentation, such as satellite image segmentation [1,2] or flood segmentation [3], one of the most important lies in the medical field [4–6]. Medical image segmentation refers to the process of dividing a medical image into different sections or regions containing similar medical features or structures. This technique is essential in medical image interpretation and decision-making in the diagnosis and treatment of diseases. In this work, we assess the problem of diabetic retinopathy, which is a medical disease related to certain morphological attributes of the retinal blood vessels, such as the length, thickness, branching or different angles formed by these vessels [7].

Prior to the use of artificial intelligence in the field of computer vision, researchers used traditional image processing algorithms to perform segmentation [8]. These classical algorithms were based on different classical techniques, such as threshold segmentation, which sets a threshold to determine which pixels belong to an object and which do not [9,10]; edge segmentation, which uses image analysis techniques to detect edges and lines in an image and subsequently segment it [11,12]; or clustering segmentation, which seeks to group pixels into different clusters or groups based on certain characteristics [13,14].

However, in recent decades, image segmentation has evolved significantly from classical algorithms, which had difficulties facing complex images or multiple overlapping objects, to deep learning-based techniques, which are currently the state of the art in image segmentation [15] and in the majority of computer vision tasks. Among the deep learning segmentation algorithms, those based on convolutional neural networks (CNNs), such as U-Net [16], was the most famous and used in the first years of the 2020s, when transformers (models based on self-attention mechanisms) appeared in [17] and were applied to computer vision (vision transformers or ViTs) [18]. In recent years, these have proven to be very effective in segmentation tasks, achieving results comparable or higher to CNNs [18,19]. For this reason, the majority of computer vision applications have changed from using CNNs to using transformer-based architectures despite requiring much more computational capacity. In order to find a balance between complexity and performance, a novel model, ConvMixer [20], emerged last year. It tries to copy the patch-representation from the ViTs, which they thought to be the source of their high performance rather than the mechanisms of self-attention itself and combine it with the feature extraction of CNNs that has already proven to be successful. They also introduce, inspired by the work of the MLP-mixer [21], the idea of combining feature information extracted through point-wise and depth-wise convolutions.

In this work, we have chosen the DRIVE (Digital Retinal Images for Vessel Extraction) dataset as a reference to present the results obtained by the different models [22]. It is, together with STARE [23] and CHASE [24], an image dataset frequently used in the study of ocular diabetic retinopathy segmentation and as a benchmark for medical image segmentation models.

In this work, we trained from scratch and compare the performance of different segmentation models on a well-known medical retinal image dataset (which was created with the aim of assisting medical specialists in diabetic retinopathy diagnosis). Some of the models are based on CNNs, such as the popular U-Net, others are based on ViTs and, finally, we also used a segmentation-adapted version of the ConvMixer model, which can be seen as a combination of both types of architectures. We obtain that a reduced version of the ConvMixer model achieves a good or better result than the other tested models while having around two orders of magnitude fewer parameters and taking less time to train. To the best of our knowledge, this is one of the few works that make use of the ConvMixer model for a medical segmentation task [25,26] and the first to perform a from scratch training comparative of models based on different architectures on a retinopathy problem.

More specifically, the paper is organized as follows: first, in Section 2, we describe the dataset, the different model architectures and the loss function and metric we used to train and evaluate the models. Then, in Section 3, we describe the results of the different models and finally, in Section 4 we discuss the results and their implications. The code to reproduce the results of this work is available at: https://github.com/alberto-solano/drive-convmixer (accessed on 28 March 2023).

2. Materials and Methods

In this section, we develop and explain the dataset and the different model architectures. Furthermore, we specify the evaluation metric and the loss function and other considerations involved in the training of the models.

2.1. DRIVE Dataset

In this work, we have chosen the DRIVE (Digital Retinal Images for Vessel Extraction) dataset as a reference to present the results obtained by the different models [22]. It is, together with STARE [23] and CHASE [24], an image dataset frequently used in the study of ocular diabetic retinopathy segmentation and as a benchmark for medical image segmentation models.

The DRIVE dataset consists of 40 high-resolution images of 584 × 565 pixels (see Table 1) whose blood vessels have been labeled by eye medical experts. Figure 1 shows one of the images and its corresponding label, which is actually a binary mask. More specifically, this dataset is intended to assist specialists in the diagnosis of diabetic retinopathy, where it is necessary to isolate the blood vessels from the retinal fundus in order to proceed with the subsequent inspection. Therefore, the goal to be achieved with this dataset involves a binary semantic segmentation task, in which each pixel must be classified according to whether or not it belongs to a blood vessel.

Table 1. Table summary of the DRIVE dataset main features.

Dataset	Training (Labeled) Images	Testing (Unlabeled) Images	Resolution
DRIVE	20	20	584 × 565

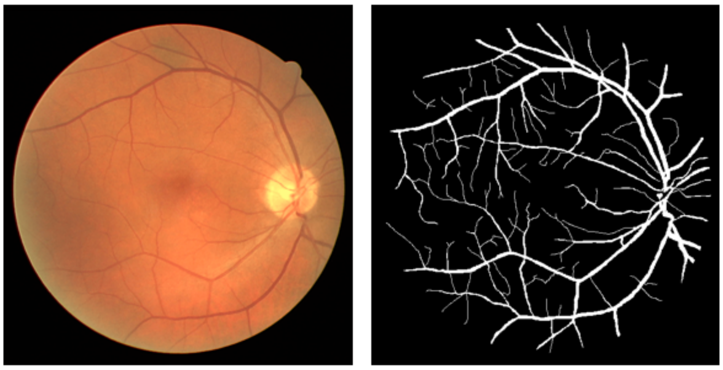

Figure 1. Example of one of the train set images (**left**) and its corresponding label (**right**). The blood vessels are segmented in white and the retinal background in black.

Out of the 40 images that make up the dataset, 20 are distributed without labels and used as test partitions. In order to obtain the test metrics, the results of these images have to be submitted to the DRIVE platform [27]. Then, the dice-coefficient is calculated for every image in the test set and results can be seen in a leaderboard sorted by highest average dice-coefficient. The other 20 labeled images are split between a training (16 images) and a validation (4 images) set. In addition to this, as the amount of images available is very small, we used different data augmentation techniques that we describe in Section 2.1.1 in order to enhance the training set.

2.1.1. Data Augmentation

We perform different data augmentation transformations to the training set images due to the lack of variability of the dataset (only 20 images available for training) and thus increase the generalization capability of the models and prevent over-fitting. We can find this to be a common technique in the literature when dealing with DRIVE and similar datasets. However, unlike other authors, we have not employed cropping and scaling techniques in small image portions [28], so the detail of specific regions of the eye is not explicitly provided to the networks (the zoom applied is very small and, even with the maximum, we always capture the entire eye). Although the values of the transformations applied have been slightly modified depending on the model (specified at Table 2), a generic description of the employed techniques is given below:

- Random rotation: Always applied, with lower and upper bounds for the rotation angle α given by parameter θ, i.e., $\alpha \in [-\theta, \theta]$ rad.
- Random horizontal flip: With probability p_{hf}.
- Brightness adjustment: Being applied 10% of the time with a random factor β between an upper and lower bound $[\beta_{min}, \beta_{max}]$, where $\beta = 0$ gives a complete black image,

$\beta = 1$ leaves the original image unchanged and $\beta > 1$ increases the brightness by that factor.

- Contrast adjustment: Furthermore, being applied with a probability of 10%, based on a factor κ between an upper and lower bound $[\kappa_{min}, \kappa_{max}]$, where $\kappa = 0$ gives a solid gray image, $\kappa = 1$ leaves the original image unchanged and $\kappa > 1$ increases the contrast by that factor.
- Gamma correction: Known as Power Law Transform, applied again with a probability of 10% with a fixed *gain* factor of 1 and a random γ factor, again between some upper and lower bounds around 1. Values smaller than 1 make the dark regions lighter while values larger than 1 make the shadows darker.
- Random affine: Transformation of the image with probability p_α, keeping the center invariant. It combines a translation in both x and y directions, i.e., $[tx, y_{min}, tx, y_{max}]$ and $t \in$, plus a random zoom of the image up to a maximum and, also, an x-shearing parameterized between two values $[s_{min}, s_{max}]$.
- Random gaussian noise: With a fixed zero-mean ($\mu = 0$) and a variable standard deviation σ.

Table 2. Table summary of the hyperparameters used for data augmentation for each model.

Model	$[-\theta, \theta]$	p_{hf}	$[\beta_{min}, \beta_{max}]$	$[\kappa_{min}, \kappa_{max}]$	$[\gamma min, \gamma max]$	p_α	$[tx_{min}, tx_{max}]$	$[ty_{min}, ty_{max}]$	$zoom_{max}$	$[s_{min}, s_{max}]$	σ
U-Net	[−180, 180]	0.4	[0.8, 1.2]	[0.8, 1.2]	[0.9, 1.1]	0.3	[0, 0]	[−0.1, 0.1]	×1.20	[0, 0]	0.1
UNETR	[−15, 15]	0.3	[0.5, 1.5]	[0.6, 1.5]	[0.7, 1.3]	0.2	[−0.1, 0.1]	[−0.1, 0.1]	×1.25	[0, 0]	0.08
Swin-Unet	[−45, 45]	0.5	[0.6, 1.4]	[0.6, 1.4]	[0.7, 1.3]	0.2	[−0.05, 0.05]	[−0.05, 0.05]	×1.20	[0, 0]	0.05
ConvMixer	[−45, 45]	0.3	[0.3, 1.7]	[0.3, 1.8]	[0.5, 1.5]	0.15	[−0.2, 0.2]	[−0.2, 0.2]	×1.25	[−0.1, 0.1]	0.1
ConvMixer-Light	[−45, 45]	0.3	[0.6, 1.6]	[0.6, 1.6]	[0.7, 1.5]	0.15	[−0.1, 0.1]	[−0.1, 0.1]	×1.25	[0, 0]	0.05

Figure 2 shows an example of the different transformations applied to one of the images so that the effect can be seen.

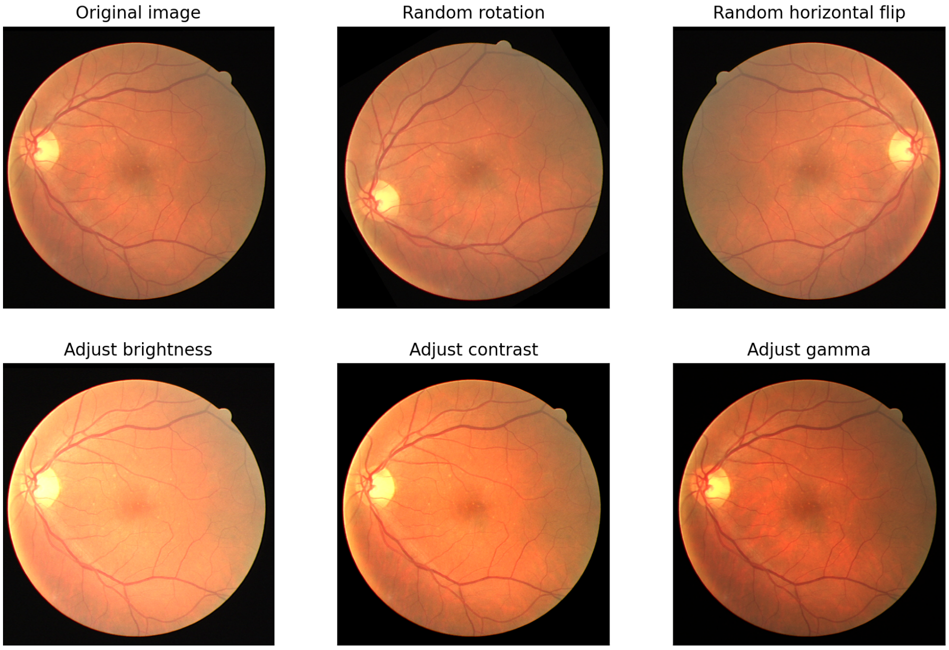

Figure 2. Example of the different data augmentation transformations applied to one of the training images.

2.2. Data Augmentation Hyperparameter Tuning

For the hyperparameter optimization of the whole set of models, we have used a grid search-based approach. Starting from a wide grid, i.e., in which the jumps are larger in magnitude from one value to its contiguous value for the same hyperparameter within the set of possible combinations, we have gradually been decreasing the grid size until converging to a local minimum. We did this process by training the models with the different hyperparameter combinations and we keep the ones that achieve higher results in the validation split.

We have used Weights & Biases [29] for experiment tracking and visualizations to develop insights for this paper, including this hyperparameter tuning task.

2.3. Evaluation Metric

The metric used in this work is the Sørensen-Dice coefficient, commonly known as DICE. It is, in conjunction with the Intersection over Union (IoU), a highly used metric in semantic segmentation problems, preferred when robustness at evaluation time is desired. The main advantage of such metrics, compared to simpler ones such as accuracy, is that they are more robust to class imbalances, which are frequently present in the segmentation context. DICE is calculated as the intersection between the ground truth label and the prediction over the sum of both areas, which is expressed as

$$DICE = \frac{2|X \cap Y|}{|X| + |Y|} = \frac{2TP}{2TP + FP + FN} \tag{1}$$

where X and Y are the real and predicted labels. In our problem, both are binary masks with values 0 for the retinal background and 1 for the blood vessels. It should be noted that DICE values can range from 0, when there is no overlap between the real and predicted labels, to 1, when there is a perfect prediction of the label.

It is worth mentioning that the metric can also be computed by using True Positive (TP), False Positive (FP) and False Negative (FN) values of the confusion matrix between the two classes represented by 0 and 1 mask pixels. In this binary scenario, the DICE coefficient is equivalent to the F_1 classification score, so the problem can be interpreted as a semantic segmentation problem evaluated with the DICE metric or even as a pixel-wise classification task evaluated with the F_1 score.

2.4. Loss Function

To train the different models, we employed a loss function that consists of a weighted combination of the DICE (1) and the Binary Cross-Entropy (BCE) Loss (2), with adjustable weights (θ) for the minority class. The loss function, then, is given by the expression:

$$BCE_\theta(y, \hat{y}) = -\theta\, y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \tag{2}$$

$$Loss(y, \hat{y}) = \alpha DICE(y, \hat{y}) + (1 - \alpha) BCE_\theta(y, \hat{y}). \tag{3}$$

where α and θ are hyperparameters that we fine-tuned individually for each model, and $(y, \hat{y})$ correspond to the ground-truth label and the prediction made by the model, respectively.

2.5. Models

In this subsection, we aim to describe the four models trained with the DRIVE dataset.

2.5.1. CNN's Models: U-Net

The first model we trained is the U-Net [16], which is a fully convolutional model. It is one of the most famous segmentation architectures and when it was released it obtained SOTA results in different segmentation tasks. It is still highly used in the medical field due to its good performance in situations where little data are available. U-Net benefits greatly from data augmentation techniques that allow to artificially increase the number of images available for the training process. Here we used the original architecture developed for medical image segmentation which has approximately 31M trainable parameters [16].

We trained this architecture on our training set for 1000 epochs using an initial learning rate of 0.002, Adam optimizer, and a batch size of 2 images. We reduce the learning rate by a factor of two after not improving the validation DICE for 80 epochs. We found that the best hyperparameters for the loss function are $\theta = 1$ and $\alpha = 0$, which implies that for this model the best result is obtained when the DICE is not included in the loss function.

2.5.2. ViT Models: UNETR and Swin-UNET

After the CNN model, we tried to outperform the results obtained through two different models based on self-attention mechanisms, i.e., two vision transformers.

On the one hand, we trained a model called UNETR, which is a transformer-based architecture developed for 3D medical image segmentation [30]. The main contribution of this model is to include skip connections between the transformer encoder and the convolutional decoder. We used a feature size of 64 and a dropout rate of 0.2 in this architecture, reaching a total of 104M trainable parameters.

For the training process, we conducted a total of 1000 epochs using an initial learning rate of 0.002, Adam optimizer and a batch of 1 image. We found that the best hyperparameters for the loss function are $\theta = 1$ and $\alpha = 0.7$.

On the other hand, we have also trained a model called Swin-Unet [31], which is a U-Net-like pure Transformer for medical image segmentation. The main difference between the UNETR and the Swin-Unet models is that the Swin-Unet uses a transformer-like architecture not only for the encoder but also for the decoder while the UNETR uses a convolutional decoder. It also includes a novel multi-head self-attention module based on shifted window named the “swin transformer block”. In this case, we used the original Swin-Unet architecture which has 27M trainable weights.

We have trained it for 3000 epochs using an initial learning rate of 0.002, Adam optimizer, and a batch size of 4 images. We found that the best hyperparameters for the lost function are $\theta = 2$ and $\alpha = 0.75$.

2.5.3. ConvMixer

Finally, we trained the ConvMixer model [20]. It is an extremely simple model that is similar in spirit to the ViT (input patches representation and isotropic architecture repeating the same block structure) but relies exclusively on convolutions to extract image features and to combine the information across the multiple layers. More specifically, it makes use of depth-wise and point-wise convolutions for mixing the extracted features across spatial and channel dimensions, respectively. Furthermore, it maintains equal size and resolution throughout the network. When this model was presented, it outperformed different ViTs and ResNets on ImageNet 1k [20]. This model was originally developed to perform classification problems so we had to adapt the last part of the model in order to be able to address a segmentation problem. To achieve this, we added a final transpose convolution layer (symmetrical to the initial one that divides the image into patches) which goes from h channels to a single channel, using a kernel size and stride equal to the patches size. The result of this layer is a single-channel mask of the same size as the input image. An additional skip connection is added to this mask which sums (instead of concatenates) the result of this layer with a channel mean of the input image. This is done to improve the convergence during the training. Figure 3 shows the ConvMixer with the additional segmentation output modification.

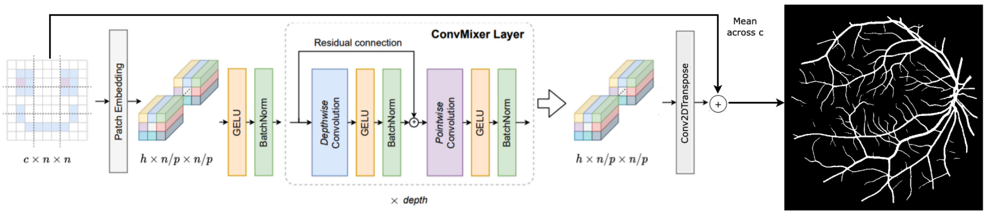

Figure 3. Original ConvMixer architecture plus the final modification to face segmentation problems. Adapted with permission from Ref. [20]. 2021, Asher Trockman.

We have trained two versions of this model, one called ConvMixer and another with less trainable parameters, which we called ConvMixer-Light. In the ConvMixer architecture, we selected a patch size of 3 pixels, an embedding dimension of 375, a depth of 20 layers and kernels of size 3 × 3. With this configuration, the number of trainable parameters increases up to 2.97M.

For the training of this model, we iterated over a total of 1500 epochs, using an initial learning rate of 0.002, Adam optimizer, and a batch size of 2 images. We found the best hyperparameters for the lost function at $\theta = 7$ and $\alpha = 0.8$.

For the ConvMixer-light version, we decreased the depth from 20 to 14 layers and the embedding dimension from 375 to 128. With this modification, we reduced the number of trainable parameters to a total of just 270k.

We have trained the ConvMixer-light model for 1000 epochs, using an initial learning rate of 0.002, Adam optimizer, and a batch size of 2 images. We found that the best hyperparameters for the lost function are $\theta = 1.5$ and $\alpha = 0.7$.

2.6. Training Considerations

To avoid over-fitting, the DICE over the validation set is calculated after each epoch and the best-performing model is stored. At the same time, the learning rate is reduced by a certain factor (cut in half in our case) after not improving the validation DICE for 80 epochs, with a floor limit of 10^{-5}. This helps the weight adjustment when the network gets closer to the minimum. After the training, in order to evaluate the models we restore the weights of the iteration in which the maximum DICE on the validation set was scored.

All the models have been trained with a single NVIDIA P100 GPU (Google Colab Pro).

3. Results

In the following section, we present the different results obtained with the different models as well as the considerations taken into account when training the models. The out of sample metrics shown are obtained from the DRIVE challenge platform after uploading our predictions, so they are considered to be the most independent metric of performance available.

3.1. U-Net

An interesting finding about the improvement produced by the data augmentation techniques was found when removing the noise and zoom transformations. By using the whole transformations set, the achieved DICE was 0.81 and 0.82 for the validation and test sets, respectively, while reducing these metrics down to 0.80 and 0.79, respectively, after removing the noise and zoom transformations.

By observing the results in Figure 4, it can be seen that the U-Net is able to correctly segment the regions of the image that have the thickest blood vessels and even some areas where a non-expert human eye would hardly be able to perceive the presence of these vessels. It can also be noted that the major failure area is focused on the smallest blood vessels.

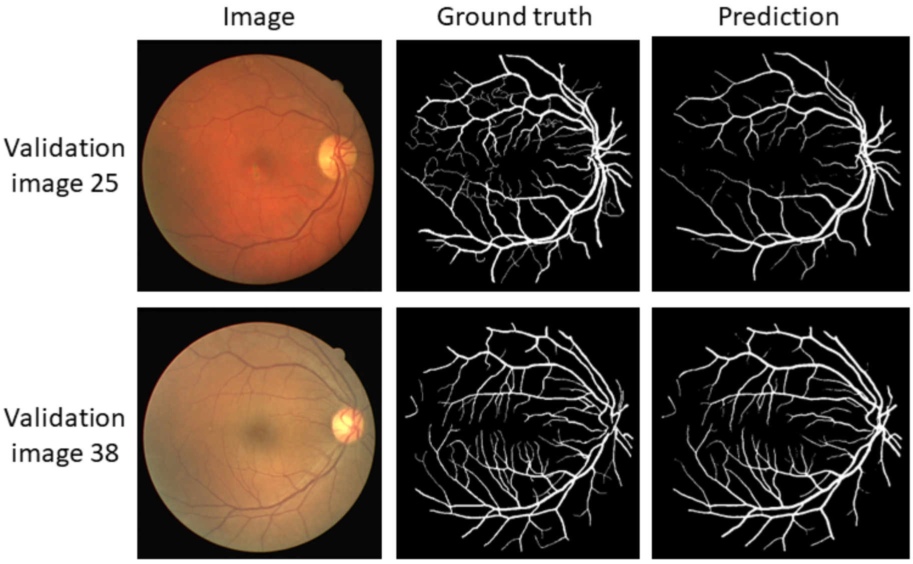

Figure 4. Two validation images, its real mask and the U-Net prediction.

3.2. *ViT's: UNETR y Swin-UNET*

ViT-based models obtained slightly worse results than CNN-based U-Net. More specifically, the UNETR model obtained a DICE of 0.80 both for the validation and test sets, which for the test results is an intermediate point between the U-Net performance with and without the zoom and noisy transformations. The Swin-Unet obtained worse results than the U-Net both in the validation and test sets, with a DICE of 0.76 and 0.77, respectively. Figures 5 and 6, which show the prediction of UNETR and Swin-Unet on two of the validation images, look noisier (look the edge of the eye in Swin-Unet prediction of validation image 25) than the U-Net predictions of Figure 4.

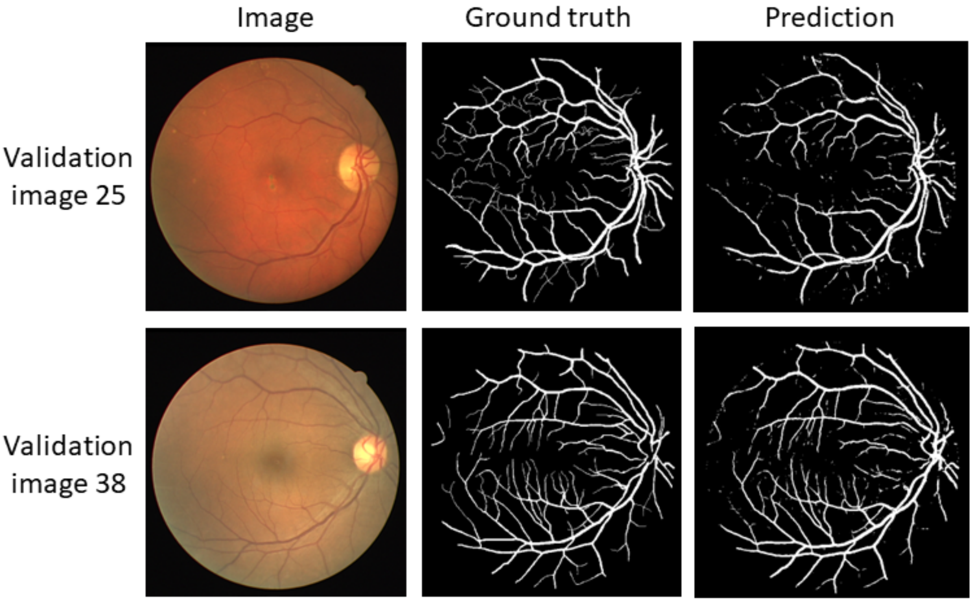

Figure 5. Two validation images, its real mask and the UNETR prediction.

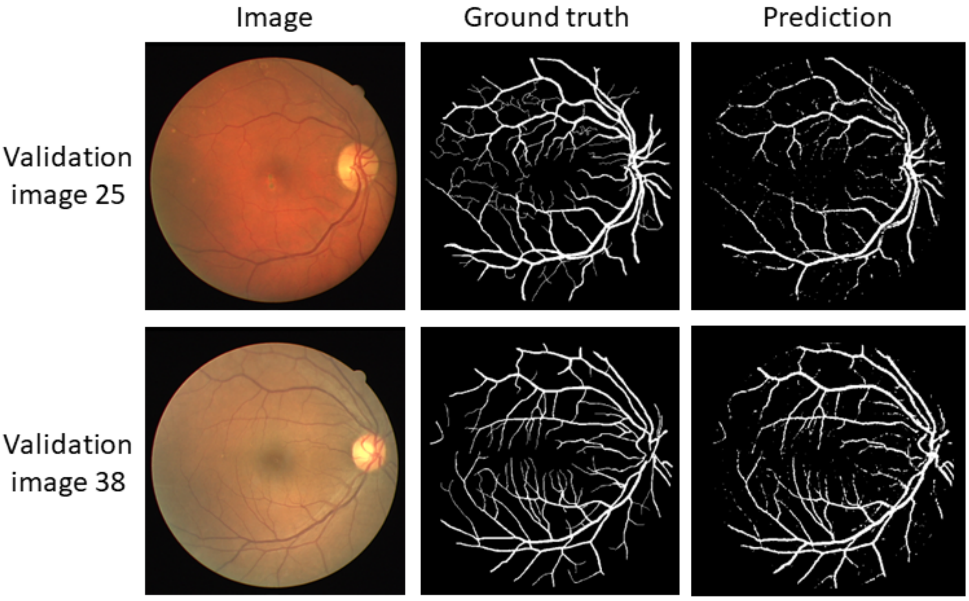

Figure 6. Two validation images, its real mask and the Swin-Unet prediction.

3.3. Convmixer

The ConvMixer model, which has much fewer trainable parameters (3M and 270k for the normal and light versions), gets the best results out of all the models. Its normal version achieved a DICE of 0.82 and 0.83 for the validation and test sets, respectively. When reducing the model size to the light version, its results only change slightly to a DICE of 0.82 both for the validation and test sets. Figures 7 and 8 show the results of the two ConvMixer models on two validation images. It is possible to see how it detects the smallest blood vessels better than the U-Net model as well as getting less noisy predictions than ViT models.

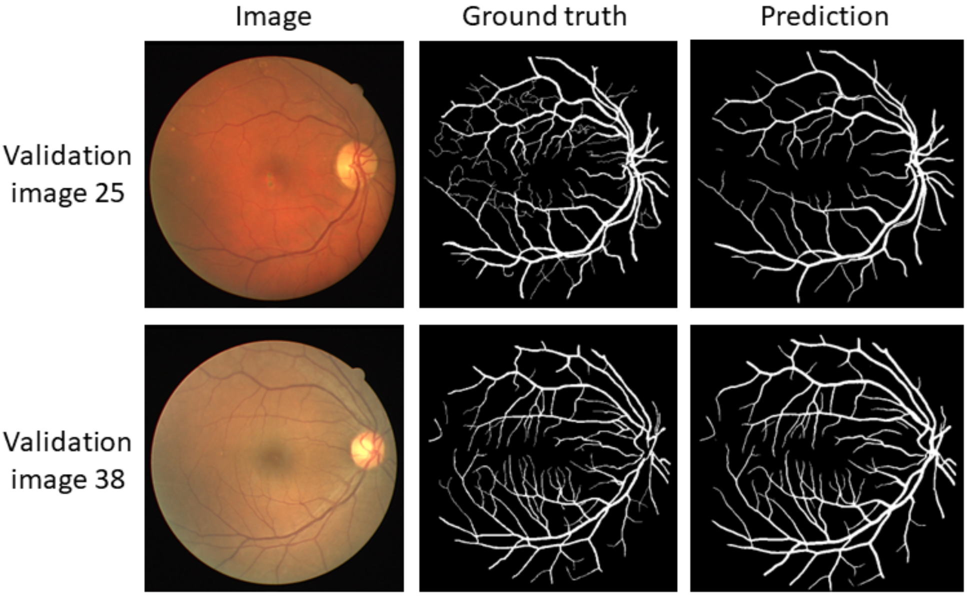

Figure 7. Two validation images, its real mask and the ConvMixer prediction.

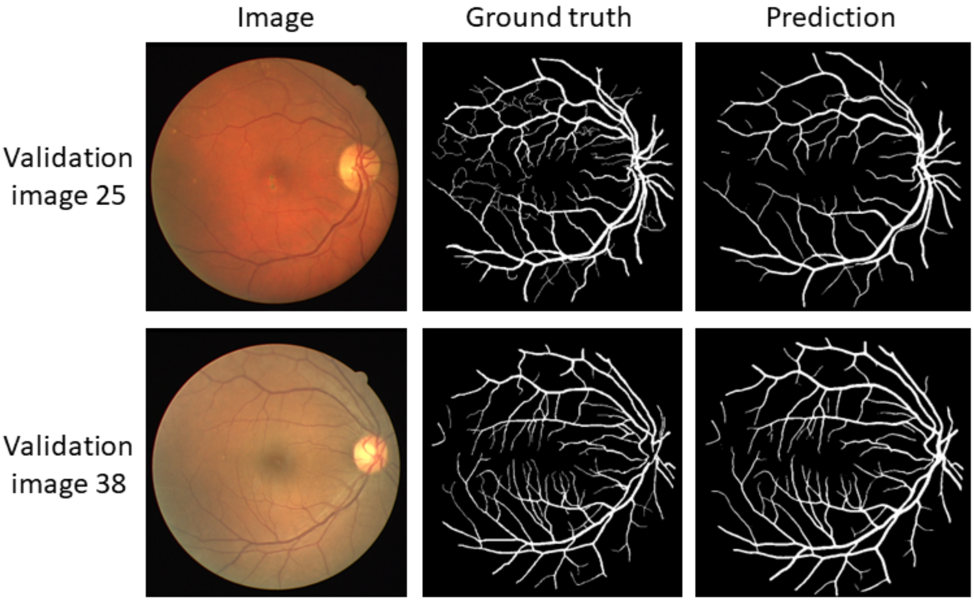

Figure 8. Two validation images, its real mask and the ConvMixer-Light prediction.

3.4. Summary

Finally, in order to see the results clearly, we show both a table and a comparative figure with the results of all the models on three images from the test set.

Table 3 shows a comparison between the five models we have trained and other state-of-the-art models on the DRIVE dataset competition. We include results in the validation and test DICE as well as the type of the model, the number of trainable parameters and the time to process one image. It shows how the ConvMixer model and its Light version obtain the best results of the five models we trained, slightly above the U-Net but with one and two orders of magnitude fewer parameters, respectively. We also saw that the ConvMixer-Light model takes only 4 s per image to train, while other models such as U-Net and UNETR take two and four times more on an NVIDIA T4.

Figure 9 shows the prediction images and DICE scores made by all the trained models for three images of the test set. It can be seen how the segmentation performed by the ConvMixer models (both the normal and the Light version) are the ones that show a higher amount of cleaner blood vessels.

Table 3. Table summary of the results in validation and test DICE of the models we trained (top panel). The models and results of some of the leaders of the DRIVE dataset competition are also shown (bottom panel).

Network	Type	Params.	Val DICE	Test DICE	Process. Time (s)
U-Net	CNN	31M	81	82	8.1
UNETR	ViT	104M	80	80	16.0
Swin-Unet	ViT	27M	76	77	6.5
ConvMixer	CNN	2.97M	**82**	**83**	11.0
ConvMixer-Light	CNN	**0.27M**	**82**	82	**4.2**
IterNet [32]	CNN	13.6M	-	82.18	-
BCDU-Net [33]	CNN-RNN	20.7M	-	82.24	-
LadderNet [34]	CNN	1.5M	-	82.02	-
RV-GAN [35]	GAN	>14M	-	**86.90**	-

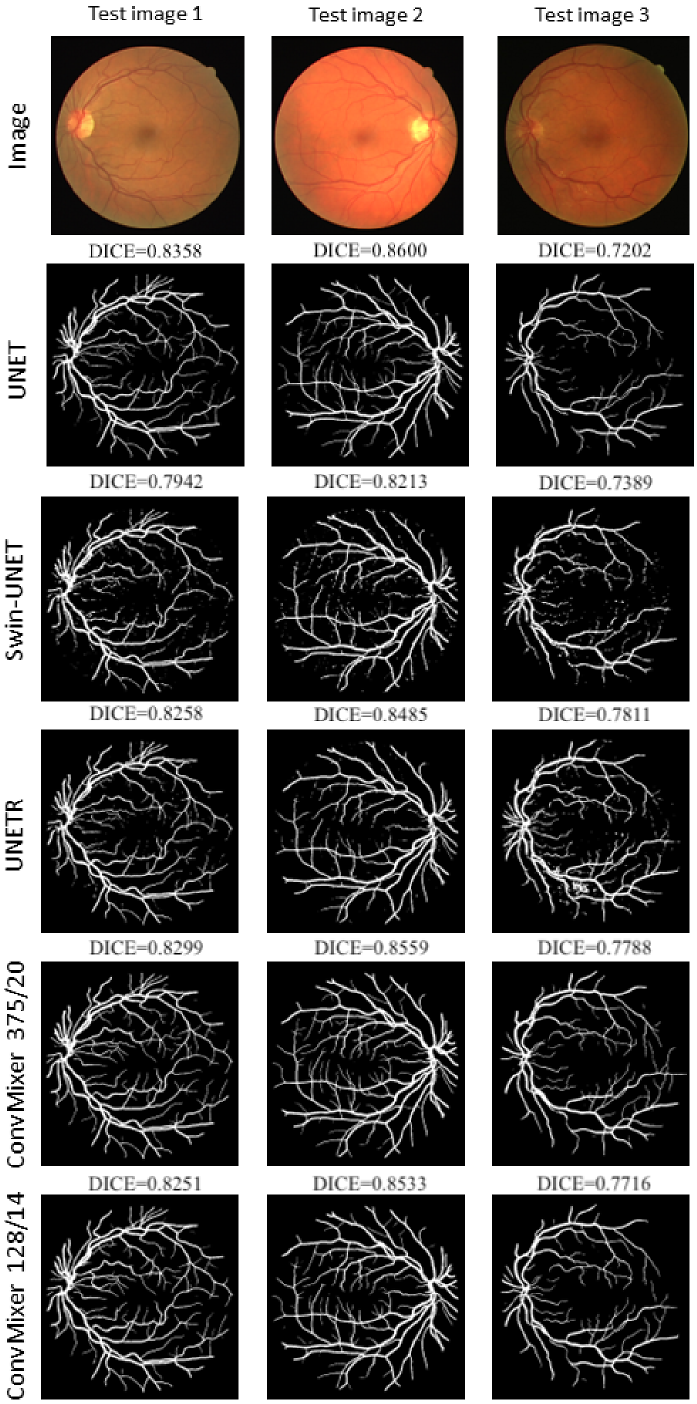

Figure 9. Prediction images and DICE scores of all the trained models in three of the test images.

4. Discussion

During this work, some of the image segmentation state-of-the-art models were studied and applied to the well-known DRIVE vessel segmentation dataset. In addition, a novel classification architecture not yet tested in segmentation problems, the ConvMixer, was also analyzed and applied for the first time on these data, giving a total of five different architectures that were trained to evaluate their performance.

The first model in our study was the U-Net, a convolutional architecture widely used and known in many fields where deep learning is applied, particularly in medical segmentation. In fact, many of the models used to tackle this type of problem are versions of this architecture. Our main findings when training and testing this model are that it can achieve a good result while failing in the smallest vessels, as well as that its performance can benefit from the use of data augmentation techniques such as zoom transformations and noise addition.

In line with the development, the following two trained models were based on the visual transformer architecture (ViT), so they are located in the frame of attention mechanisms. In this case, we obtained slightly worse results than the first convolutional U-Net model, probably due to the limited quantity of data available to train the transformers. These models require a significant amount of data to prove their advantages, hence we expect that a pre-trained version of these ViTs may improve the performance.

Finally, our main milestone was the finding of the ConvMixer net as a promising architecture for segmentation problems. This model, although inspired by some of the ideas on which ViT is based such as patch representation and isotropic architecture, is still convolutional and can avoid the aforementioned drawbacks of transformers. While this model is not state-of-the-art, we showed that a modified version adapted to segmentation can outperform the other analyzed models despite having far fewer parameters (2.97M trainable parameters compared to 31M for U-Net or 27M and up to 104M for visual transformers). The power of this architecture is illustrated by the fact that we were able to train a network even lighter in parameters (300k), called ConvMixer-Light, which still shows better performance than U-Net and ViTs and yields just slightly worse results than the larger ConvMixer version previously trained while taking only 4 s per image to train. In this way, we proved that there is no need to systematically use large models to address image segmentation problems, specifically when there is a limited amount of data available. A smaller architecture with the optimal pieces can obtain better results than other larger models, taking advantage of higher robustness against over-fitting and thus a better generalization power over unseen data.

Our work has shown that a slightly modified version of the ConvMixer model can yield very promising results in a semantic segmentation problem. Nevertheless, further work can be done in this direction to deepen the advantages of this hybrid architecture. A first step may be to extend and test this model to other datasets and benchmarks in image segmentation, even beyond the medical context and also in multi-class problems. In this regard, a retraining approach may be complemented with a generalization test that can be performed by applying the already trained models [36–38] to other similar vessel segmentation datasets. On the other side, in addition to the generalization analysis, an ablation study over the model hyperparameters can provide a better understanding of the strengths of the ConvMixer architecture and its performance. In this work, a first insight was drawn when we found that reducing the depth and the embedding dimensions does not critically affect the model performance (by reducing the model parameters by a factor of 10 the results became slightly worse).

Author Contributions: Conceptualization, A.S., K.N.D. and M.M.-S.; methodology, A.S., K.N.D. and M.M.-S.; software, A.S. and K.N.D.; validation, K.N.D., J.V.-T., P.H.-C. and R.B.-C.; formal analysis, A.S., K.N.D., J.V.-T., P.H.-C. and R.B.-C.; investigation, A.S., K.N.D., J.V.-T., P.H.-C. and R.B.-C.; resources, M.M.-S.; data curation, A.S. and K.N.D.; writing—original draft preparation, A.S., K.N.D., J.V.-T., P.H.-C. and R.B.-C.; writing—review and editing, A.S., K.N.D., J.V.-T., P.H.-C., R.B.-C. and M.M.-S.; visualization, A.S., K.N.D., J.V.-T., P.H.-C. and R.B.-C.; supervision, K.N.D. and M.M.-S.; project administration, M.M.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by MICIIN/FEDER/UE under Grant PID2020-118071GB-I00, Spanish MIU under Grant FPU21/02256 and GVA under Grant CIPROM/2021/056.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Datasets used in this work: DRIVE: Digital Retinal Images for Vessel Extraction https://drive.grand-challenge.org/ (accessed on 28 March 2023), Code: GitHub project https://github.com/alberto-solano/drive-convmixer (accessed on 28 March 2023).

Acknowledgments: We would like to thank Emilio Soria Olivas for their valuable comments and feedback on this work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNNs	Convolutional Neural Networks
ViT	Vision Transformer
DRIVE dataset	Digital Retinal Images for Vessel Extraction dataset
STARE dataset	STructured Analysis of the Retina dataset
CHASE dataset	Child Heart and Health Study in England dataset
TP	True Positive
FP	False Positive
FN	False Negative
IoU	Intersection over Union
BCE	Binary Cross-Entropy

References

1. McGlinchy, J.; Johnson, B.; Muller, B.; Joseph, M.; Diaz, J. Application of UNet Fully Convolutional Neural Network to Impervious Surface Segmentation in Urban Environment from High Resolution Satellite Imagery. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 3915–3918. [CrossRef]
2. Pesaresi, M.; Benediktsson, J.A. A new approach for the morphological segmentation of high-resolution satellite imagery. *IEEE Trans. Geosci. Remote. Sens.* **2001**, *39*, 309–320. [CrossRef]
3. Nemni, E.; Bullock, J.; Belabbes, S.; Bromley, L. Fully convolutional neural network for rapid flood segmentation in synthetic aperture radar imagery. *Remote. Sens.* **2020**, *12*, 2532. [CrossRef]
4. Xie, B.; Li, S.; Li, M.; Liu, C.H.; Huang, G.; Wang, G. SePiCo: Semantic-Guided Pixel Contrast for Domain Adaptive Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**. [CrossRef]
5. Antonelli, M.; Reinke, A.; Bakas, S.; Farahani, K.; AnnetteKopp-Schneider, A.; Landman, B.A.; Litjens, G.; Menze, B.; Ronneberger, O.; Summers, R.M.; et al. The Medical Segmentation Decathlon. *Nat. Commun.* **2021**, *13*, 4128. [CrossRef] [PubMed]
6. Tsoukas, V.; Boumpa, E.; Giannakas, G.; Kakarountas, A. A Review of Machine Learning and TinyML in Healthcare. In Proceedings of the 25th Pan-Hellenic Conference on Informatics, New York, NY, USA, 26–28 November 2021; pp. 69–73. [CrossRef]
7. Fong, D.S.; Aiello, L.; Gardner, T.W.; King, G.L.; Blankenship, G.; Cavallerano, J.D.; Ferris, F.L., III; Klein, R.; for the American Diabetes Association. Retinopathy in Diabetes. *Diabetes Care* **2004**, *27*, s84–s87. [CrossRef] [PubMed]
8. Kaur, J.; Agrawal, S.; Renu, V. A Comparative Analysis of Thresholding and Edge Detection Segmentation Techniques. *Int. J. Comput. Appl.* **2012**, *39*, 29–34. [CrossRef]
9. Zhu, S.; Xia, X.; Zhang, Q.; Belloulata, K. An image segmentation algorithm in image processing based on threshold segmentation. In Proceedings of the 2007 Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, Shanghai, China, 16–18 December 2007; pp. 673–678.
10. Gupta, A.; Issac, A.; Dutta, M.K.; Hsu, H.H. Adaptive Thresholding for Skin Lesion Segmentation Using Statistical Parameters. In Proceedings of the 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), Taipei, Taiwan, 27–29 March 2017; pp. 616–620.
11. Al-Amri, S.S.; Kalyankar, N.; Khamitkar, S. Image segmentation by using edge detection. *Int. J. Comput. Sci. Eng.* **2010**, *2*, 804–807.
12. Canny, J. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698. [CrossRef]
13. Yu, W.; Fritts, J.; Sun, F. A hierarchical image segmentation algorithm. In Proceedings of the IEEE International Conference on Multimedia and Expo, Lausanne, Switzerland, 26–29 August 2002; Volume 2, pp. 221–224. [CrossRef]
14. Dhanachandra, N.; Manglem, K.; Chanu, Y.J. Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm. *Procedia Comput. Sci.* **2015**, *54*, 764–771. .: 10.1016/j.procs.2015.06.090. [CrossRef]
15. Mahony, N.O.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Velasco-Hernández, G.A.; Krpalkova, L.; Riordan, D.; Walsh, J. Deep Learning vs. Traditional Computer Vision. *Adv. Comput. Vis.* **2019**, *943*, 128–144. [CrossRef]

16. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
17. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
18. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
19. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *arXiv* **2021**, arXiv:2103.14030.
20. Trockman, A.; Kolter, J.Z. Patches Are All You Need? *arXiv* **2022**, arXiv:2201.09792.
21. Tolstikhin, I.O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. Mlp-mixer: An all-mlp architecture for vision. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 24261–24272.
22. Staal, J.; Abramoff, M.; Niemeijer, M.; Viergever, M.; van Ginneken, B. Ridge-based vessel segmentation in color images of the retina. *IEEE Trans. Med. Imaging* **2004**, *23*, 501–509. [CrossRef] [PubMed]
23. Hoover, A.D.; Kouznetsova, V.; Goldbaum, M. Locating Blood Vessels in Retinal Images by Piece-wise Threhsold Probing of a Matched Filter Response. *IEEE Trans. Med. Imaging* **2000**, *19*, 203–210. [CrossRef]
24. Fraz, M.M.; Remagnino, P.; Hoppe, A.; Uyyanonvara, B.; Rudnicka, A.R.; Owen, C.G.; Barman, S.A. An Ensemble Classification-Based Approach Applied to Retinal Blood Vessel Segmentation. *IEEE Trans. Biomed. Eng.* **2012**, *59*, 2538–2548. [CrossRef]
25. Toan, N.Q. Aiding Oral Squamous Cell Carcinoma diagnosis using Deep learning ConvMixer network. *medRxiv* **2022**. [CrossRef]
26. Tang, F.; Wang, L.; Ning, C.; Xian, M.; Ding, J. CMU-Net: A Strong ConvMixer-based Medical Ultrasound Image Segmentation Network. *arXiv* **2022**, arXiv:2210.13012.
27. Center, R.U.M. DRIVE: Digital Retinal Images for Vessel Extraction—Grand Challenge. Available online: https://drive.grand-challenge.org/ (accessed on 28 March 2023).
28. Boudegga, H.; Elloumi, Y.; Akil, M.; Hedi Bedoui, M.; Kachouri, R.; Abdallah, A.B. Fast and efficient retinal blood vessel segmentation method based on deep learning network. *Comput. Med. Imaging Graph.* **2021**, *90*, 101902. [CrossRef] [PubMed]
29. Biewald, L. Experiment Tracking with Weights and Biases. 2020. Available online: www.wandb.com (accessed on 28 March 2023).
30. Hatamizadeh, A.; Tang, Y.; Nath, V.; Yang, D.; Myronenko, A.; Landman, B.; Roth, H.; Xu, D. UNETR: Transformers for 3D Medical Image Segmentation. *arXiv* **2021**, arxiv:2103.10504.
31. Cao, H.; Wang, Y.; Chen, J.; Jiang, D.; Zhang, X.; Tian, Q.; Wang, M. Swin-Unet: Unet-like Pure Transformer for Medical Image Segmentation. *arXiv* **2021**, arxiv:2105.05537.
32. Li, L.; Verma, M.; Nakashima, Y.; Nagahara, H.; Kawasaki, R. IterNet: Retinal Image Segmentation Utilizing Structural Redundancy in Vessel Networks. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass Village, CO, USA, 1–5 March 2020.
33. Azad, R.; Asadi-Aghbolaghi, M.; Fathy, M.; Escalera, S. Bi-directional ConvLSTM U-Net with densley connected convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019.
34. Zhuang, J. LadderNet: Multi-path networks based on U-Net for medical image segmentation. *arXiv* **2018**, arXiv:1810.07810.
35. Kamran, S.A.; Hossain, K.F.; Tavakkoli, A.; Zuckerbrod, S.L.; Sanders, K.M.; Baker, S.A. RV-GAN: segmenting retinal vascular structure in fundus photographs using a novel multi-scale generative adversarial network. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Strasbourg, France, 27 September–1 October 2021; pp. 34–44.
36. Ban, Y.; Wang, Y.; Liu, S.; Yang, B.; Liu, M.; Yin, L.; Zheng, W. 2D/3D Multimode Medical Image Alignment Based on Spatial Histograms. *Appl. Sci.* **2022**, *12*, 8261. [CrossRef]
37. Qin, X.; Ban, Y.; Wu, P.; Yang, B.; Liu, S.; Yin, L.; Liu, M.; Zheng, W. Improved Image Fusion Method Based on Sparse Decomposition. *Electronics* **2022**, *11*, 2321. [CrossRef]
38. Liu, H.; Liu, M.; Li, D.; Zheng, W.; Yin, L.; Wang, R. Recent Advances in Pulse-Coupled Neural Networks with Applications in Image Processing. *Electronics* **2022**, *11*, 3264. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.