

Received September 12, 2019, accepted September 24, 2019, date of publication September 30, 2019, date of current version October 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2944692

# TPU Cloud-Based Generalized U-Net for Eye Fundus Image Segmentation

JAVIER CIVIT-MASOT<sup>1</sup>, FRANCISCO LUNA-PEREJÓN<sup>2</sup>, SATURNINO VICENTE-DÍAZ<sup>2</sup>,  
JOSÉ MARÍA RODRÍGUEZ CORRAL<sup>3</sup>, AND ANTÓN CIVIT<sup>2</sup>

<sup>1</sup>COBER S.L., 41012 Seville, Spain

<sup>2</sup>School of Computer Engineering, 41012 Seville, Spain

<sup>3</sup>School of Engineering, Avenida de la Universidad de Cádiz, 11519 Cádiz, Spain

Corresponding author: Javier Civit-Masot (javi.civit@gmail.com)

This work was supported in part by the NPP project through SAIT (2015–2018), and in part by the Spanish Government Grant (with support from the European Regional Development Fund) COFNET under Grant TEC2016-77785-P.

**ABSTRACT** Medical images from different clinics are acquired with different instruments and settings. To perform segmentation on these images as a cloud-based service we need to train with multiple datasets to increase the segmentation independency from the source. We also require an efficient and fast segmentation network. In this work these two problems, which are essential for many practical medical imaging applications, are studied. As a segmentation network, U-Net has been selected. U-Net is a class of deep neural networks which have been shown to be effective for medical image segmentation. Many different U-Net implementations have been proposed. With the recent development of tensor processing units (TPU), the execution times of these algorithms can be drastically reduced. This makes them attractive for cloud services. In this paper, we study, using Google's publicly available colab environment, a generalized fully configurable Keras U-Net implementation which uses Google TPU processors for training and prediction. As our application problem, we use the segmentation of Optic Disc and Cup, which can be applied to glaucoma detection. To obtain networks with a good performance, independently of the image acquisition source, we combine multiple publicly available datasets (RIM-One V3, DRISHTI and DRIONS). As a result of this study, we have developed a set of functions that allow the implementation of generalized U-Nets adapted to TPU execution and are suitable for cloud-based service implementation.

**INDEX TERMS** Deep learning, segmentation as a service, TPU, U-Net, optic disc and cup, glaucoma.

## I. INTRODUCTION

### A. CLOUD BASED MEDICAL IMAGE SEGMENTATION

Segmentation is the process of automatic or semi-automatic detection of limits within a 2D or 3D image. A well-known difficulty in the segmentation of medical images is the high variability in the data sources and capture technologies. First, anatomy shows very significant variations. In addition, many different image acquisition systems are used (X-ray, CT, MRI, PET, SPECT, endoscopy, etc.) to create biomedical images. The segmentation result can also be used to obtain additional diagnostic information. Among the possible applications, we can find automatic measurement of organs, cell count or simulations based on the acquired information.

The associate editor coordinating the review of this manuscript and approving it for publication was Yunjie Yang<sup>1</sup>.

The application of Deep Learning methods to medical image analysis has quickly grown in recent years [1] due to their success with different problems, including segmentation. The effectiveness of these systems improves with the number and variety of the training set images. This suggests the development of cloud-based services that can be trained with several dataset initially and retrained with new datasets samples periodically. Reducing training times is an important requirement in this scenario, and Google TPUs are currently one of the most powerful resources available to train and carry out predictions for cloud-based segmentation. Another important aspect is that, in a cloud-based service, images will come from very different sources and, thus the networks must be trained as independently as possible from the acquisition source. Several segmentation researchers [2], [3] have used several datasets. However, they always train and test with each of these datasets independently. This methodology is not

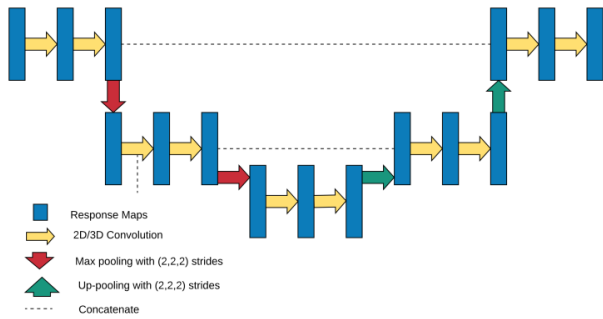


FIGURE 1. Basic U-Net Architecture modified to three stages.

suitable for our application scenario. In this paper we use a new approach where we preprocess and mix the data from several datasets and use it to create independent datasets for training and validation. There are some works where multiple datasets are used simultaneously (e.g. [4]) but they are not related to image segmentation.

U-Net is a fully convolutional deep learning network that has been shown to be effective in several medical segmentation problems. In this work, we focus on the study of generalized U-Net architectures as a method to solve the image segmentation problem in the cloud.

**B. CONVOLUTION / DECONVOLUTION NETWORKS: U-NET**

U-Net is one of the most commonly used fully convolutional network (FCN) for the segmentation of biomedical images. It was initially published in MICCAI 2015 and the original paper currently has more than 3800 citations [5].

The basic architecture of the network is shown in Fig. 1. The network consists of descending layers formed by two convolution layers with RELU activation and dropout. The result of each layer is sub-sampled using a  $2 \times 2$  max pool layer and used as input to the next layer. The 5th layer corresponds to the lowest level of the network and has a structure like the other descending layers. From this layer the data is oversampled (in the original version by transposed convolution), merged with the output data of the corresponding downwards layer and applied to a block similar to those used in the descending layers. The last layer of the network is a convolution layer with a width equal to the number of classes to be segmented.

The details of the implementation are different on most U-Net based projects. They can vary among other things in the following characteristics:

- Layer width: Traditionally, when going down in the network the width of the layer is doubled, and when going up it is divided by 2. This, however, is not always the case. For example, in [2] the structure of Fig. 2 is used. When the relation between the width of a layer to that of the one above it in the network is constant, this parameter is called layer increment ratio (IR). In the original U-Net this increment ratio is 2, but implementations may use smaller values to maintain a reasonable number of trainable parameters in the network.

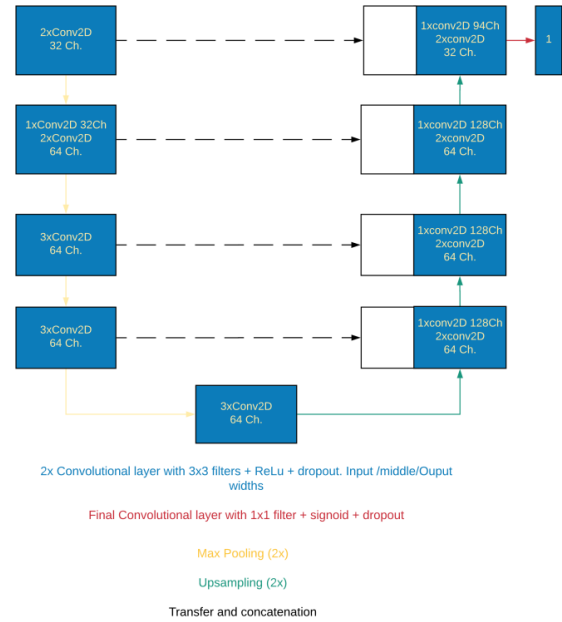


FIGURE 2. Sevastopolsky's U-Net.

- Transposed convolution or upsampling: In the U-Net up stages we double the resolution of the image in every stage. There are two main approaches for this, either we directly replicate the data to create a higher resolution image, or we use transpose convolution, i.e. a trainable upsampling convolutional layer whose parameters will change during training. Many current U-Net implementations use direct oversampling instead of transposed convolution. An evaluation of this topic can be found in [6]. In our case, we must use transpose convolution for TPU implementations, as direct upsampling is not supported on this architecture.
- Drop-out and Normalization layers: These are used to avoid overfitting the data. This happens when the system learns all the details of the training dataset but can't generalize the prediction when validating with other datasets.
- Optimization algorithms: A decision that has great impact on learning process speed, as well as the accuracy of obtained predictions, is the choice of the optimization strategy [7].

In our work, we will focus on the influence of the layer widths and the use of normalization and drop-out, as these are some of the aspects that vary widely between different implementations and affect both learning speed and prediction quality. We will make trials with U-Net implementations that are deeper than the standard 5-layer network and with different layer increment ratios.

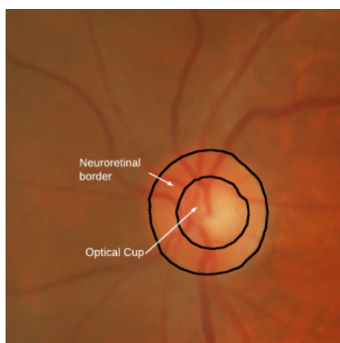
**C. GLAUCOMA, OPTIC DISC AND CUP**

Glaucoma denotes a set of conditions that causes damage to the optic nerve in the back of the eye and, thus, can lead

to loss of vision. Glaucoma is one of the most common causes of blindness, and it is estimated that it will affect around 80 million people worldwide by 2020 [8]. It is often mentioned as the “silent thief of vision” because, in its early stages, patients have little pain or vision loss symptoms.

It is only when the disease progresses that the symptoms that potentially lead to total blindness begin to be noticed. Early detection and timely treatment are the key to help preventing loss of vision. Many risk factors are associated with glaucoma, but intraocular hypertension (IH) is the most widely accepted.

IH can cause irreversible damage to the optic nerve or optic disc (OD). The optic disc is the beginning of the optic nerve and is the point where the axons of retinal ganglion cells come together. It is also the entry point for the major blood vessels that supply the retina, and it corresponds to a small blind spot in each eye. The optic disc can be visualized by various medical imaging techniques, including color fundus photography. The OD can be divided into two regions as shown in Fig. 3: a peripheral zone called the neuroretinal border and a white central region called the optic cup (OC).



**FIGURE 3.** Neuroretinal border and Cup.

Glaucoma often produces pathological cupping of the optic disc. The neuroretinal border contains nerve fibers. The cup is a pit with no nerve fibers. As glaucoma advances, the cup enlarges until it occupies most of the disc area. The ratio of the diameter of OC to OD is known as CDR and is a well-established indicator for the diagnosis of glaucoma [9].

The precise segmentation of OD and OC is essential for the measurement of the CDR. Human segmentation of OD and OC in fundus images is a slow and error prone process, which is not very useful in work-loaded ophthalmological clinics. Thus, automated segmentation can be very attractive as, in many cases, it can be more objective and faster than human segmentation.

In the literature, multiple approaches have been proposed for the segmentation of OD and OC in fundus images. The existing methods for automated OD and OC segmentation in background images can be classified into three main categories [10]: templates based on form matching using traditional machine learning with random forests, support vector machines, K-means, etc. (e.g. [11], [12], [13]),

active contours and deformable models (e.g. [14]), and more recently, deep learning-based methods (e.g. [3], [15], [16], [17]).

In this work, we will deal only with the techniques for OD/OC segmentation based on deep learning through completely convolutional networks (e.g. [2]), since our objective is to study the influence of the architecture and parameters of this type of networks in the segmentation of the optic disc and the optic cup. When the segmentation of both items is accurate, the CDR can be trivially calculated.

## II. MATERIALS AND METHODS

For this work, a toolset of functions was developed to generalize U-Net models, allowing a quick and adequate implementation on cloud-based GPU and TPU architectures. We used the cooperative iPython notebook development environment Google Colaboratory (<https://colab.research.google.com>). The environment has very good support for Keras [18], with the possibility of implementing and training networks based on GPUs and TPUs [19] in Google Cloud.

TPUs are a new type of processors designed for deep learning network acceleration that use a systolic array for multiplication and can decrease the learning times for convolutional neural networks (CNNs) several times. Training on TPUs allows us to test wider and deeper architectures that will be out of the memory limits of many current single GPU systems. The higher training speeds also allows us to prune the network to make it lighter with small effects in prediction efficiency.

Although we will do most of our training and predictions directly on TPUs, we will perform a small set of trials on GPUs to verify this claim for U-Net based segmentation.

We initially based our work on the notebooks by [2], but with very significant modifications:

- We use a completely different dual image generator for both for training and testing. For TPU training, we need larger static datasets and thus we make use of static data augmentation including images with modified brightness and modified parameters for adaptive histogram equalization. This, together with the use of images from three different publicly available datasets for training and validation, improves the system robustness allowing the use of images acquired with different instruments. Aggressive data augmentation has been shown as a very effective approach to avoid overfitting in image segmentation [20].
- We use the version of Keras included in TensorFlow. This is necessary to be able to execute it on TPUs. To our best knowledge, this is the first time that generalized U-Nets have been implemented and trained on TPUs.
- We use Pröve’s parameterizable recursive U-net model (<https://github.com/pietz/unet-keras>). This model allows us to easily change many parameters necessary to compare different implementations of U-Net. Specifically, we can change the network depth and width, the use of

drop-out and batch normalization, the use of upsampling (although this type of layer is not currently supported by Keras in TPUs) or transpose convolution and the width ratio between successive layers (IR). IR was originally introduced in [21] and is widely used as an effective pruning method. In our work we will always choose the networks with smaller IR and, thus, smaller number of trainable parameters when two networks produce similar results. Even though we don't train or perform predictions in the user device, in which case pruning would be essential, when using cloud-based resources, pruning improves timing and reduces operational costs. Reducing the network initial width and its depth are alternative pruning methods that we also explore.

We use 120 image batches for both training and testing, and we train for 15 epochs using 150 training steps and 30 testing steps per epoch. We use an Adam optimizer algorithm in most cases with a .00075 learning rate, although in a few cases we have had to lower this value to ensure convergence. These values have proven suitable for TPU-based training in U-Net architectures and provide good results with training times below 30 minutes even for the most complex implementations. In this training times include the recompilation processes carried out by the TPU XLA just-in-time compiler.

Regarding the datasets, we use publicly available RIM-ONE v3, DRISHTI and DRIONS datasets. The use of multiple datasets simultaneously, both for training and for validation, allows a greater independence from the capture devices. RIM ONE-v3 [22], from the MIAG group at the University of La Laguna (Spain), consists of 159 fundus images which have been labeled by expert ophthalmologists for both disc and cup. DRISHTI-GS [23], from Aravind Eye Hospital, Madurai (India), consists of 101 fundus images also labeled for disc and cup. DRIONS-DB [24] from Miguel Servet Hospital, Saragossa (Spain), consists of 110 images on which only the optic cup has been labelled.

We provide a Google Colaboratory iPython notebook at GitHub (<https://github.com/javicivit/TPU-UNET>) for disc and cup segmentation. The code for both cases is the same, and the only difference is the loading and pre-processing of images and masks.

As already mentioned, to perform disc and cup detection as a service in the cloud, it is necessary that we are independent, as much as possible, from the specific characteristics of the captured image. As an example, in Fig. 4 we can see that images coming from the three different datasets have very different characteristics.

Our approaches for disc and cup segmentation are very similar. Fig. 5 shows the methodology used for cup segmentation. In this case we use the only two datasets that include the required data (RIM-ONE and DRISHTI). Originally, we start by clipping and resizing the original images in the datasets. When we segment the disc, we remove a 10% border in all the edges of the image to reduce black borders in the images. When we segment the cup, we select the area that contains the disc plus an additional 10% from the original images.

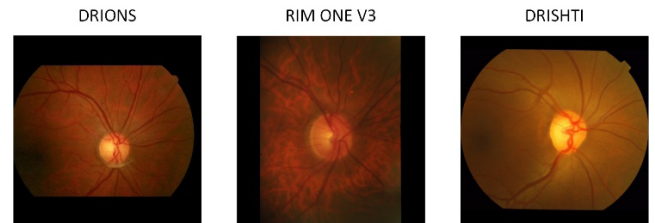


FIGURE 4. Images from different datasets.

After clipping, we resize the images to  $128 \times 128$  pixels and perform a clip limited contrast equalization.

After the equalization, we split the dataset. For each dataset, we use 75% of the images for training and 25% for validation. It is essential to split the datasets before performing any data augmentation, in order to ensure that the training and validation sets are completely independent from each other. After splitting we perform, for each dataset, static data augmentation by creating images with modified brightness and different adaptive contrast parameters.

After the static data augmentation, we merge the data from the different datasets. This process is done independently for the training and validation dataset. In the fusion process, we perform data replication and shuffling so that we provide longer vectors as input for our dynamic image generators. The image generators [25] do data augmentation by performing random rotations, shifting, zooming and flipping on the extended fused dataset images.

### III. RESULTS

Regarding the Disc segmentation (Table 1), for our experiments we initially use a network that is very similar to the original U-Net: 5-stage, no batch normalization and default dropout rates (0.3). We always use transpose convolution, as in the original implementation as direct upsampling is not currently supported on TPUs.

TABLE 1. Disc Segmentation results.

D/W/BN/IR	Train/Test	Best/Worst /Std.	RRP	MTP
5/32/N/1.5G	84/70	97/55/10	75	3.5
5/32/Y/1.5G	94/91	99/69/7	95	3.5
5/40/Y/1.2G	90/79	98/64/9	95	1.1
6/40/Y/1.3G	95/91	98/64/9	96	3.3
<b>6/40/Y/1.1G</b>	<b>95/91</b>	<b>97/59/9</b>	<b>95</b>	<b>.9</b>
7/40/Y/1.2G	95/92	98/61/11	97	2.6
<b>7/64/Y/1.3T</b>	<b>96/94</b>	<b>99/62/8</b>	<b>97</b>	<b>14</b>

Table 1 shows the Dice coefficients for the learning and for the test sets for several evaluated network alternatives. The first row in the table defines the main U-Net architecture parameters, i.e. the network depth (D), the number of filters in the first layer (W), the use of batch normalization and the increment ratio (IR). As an example, 6/40/Y/1.1 means that we use a 6 layer generalized U-Net with 40 channels in



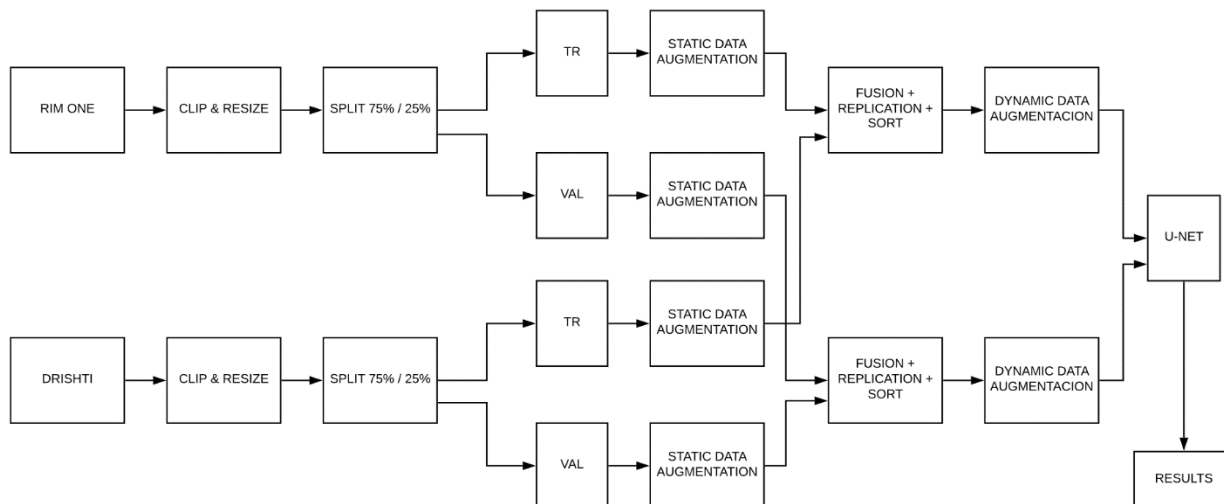


FIGURE 5. Multidataset based training approach. The diagram shows only two datasets for simplicity.

TABLE 2. Cup Segmentation results.

D/W/BN/IR	Train/Test	Best/Worst /Std.	RRP	MTP
4/72/Y/2.0T	98/94	99/60/9	74	44
4/72/Y/1.2T	97/93	99/55/12	74	4.7
4/96/Y/2.0T	98/94	99/58/10	80	78
5/32/N/1.5G	91/85	97/61/8	61	3.5
5/32/Y/1.5G	96/93	99/45/11	72	3.5
<b>5/64/Y/1.3T</b>	<b>96/90</b>	<b>99/53/11</b>	<b>77</b>	<b>4.9</b>
6/64/Y/1.1T	96/92	99/56/11	72	2.4
6/64/Y/1.3T	97/93	99/68/9	77	8.5
<b>6/72/Y/1.2T</b>	<b>97/94</b>	<b>99/62/10</b>	<b>77</b>	<b>5.6</b>
<b>6/96/Y/1.1T</b>	<b>97/94</b>	<b>99/56/11</b>	<b>77</b>	<b>5.3</b>
6/96/Y/1.2T	97/94	99/51/11	78	10.2

the first layer, batch normalization and a 1.1 layer to layer channel increment ratio (IR).

Apart from the base case and its modification including batch normalization, we provide data from pruned networks where we try to obtain the same or greater performance with a smaller number of trainable parameters. To achieve this goal, we decrease the increment ratio while increasing the number of filters in the first layer, the depth of the network or both. The column MTP in tables 1 and 2 shows the millions of trainable parameters in the network.

The Dice coefficient is defined, as usual, as twice the number of active pixels in the intersection of the true and the predicted masks divided by the sum of the active pixels in both masks. In the tables, our Dice coefficient are shown as percentages. For each proposed network architecture, we provide the mean Dice coefficient for the training and testing sets, the Dice coefficient for the best and worst predicted images in the testing set, and the standard deviation for the Dice coefficient over the testing set.

We define a new additional parameter (Radii Ratio parameter- RRP) that is very useful to estimate the accuracy of the CDR. This parameter is defined as the percentage of

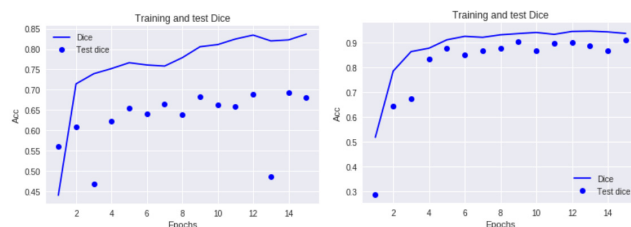


FIGURE 6. Batch Normalization effect. The left side learning curve corresponds to a network without batch normalization. The right side one corresponds to the equivalent network with batch normalization.

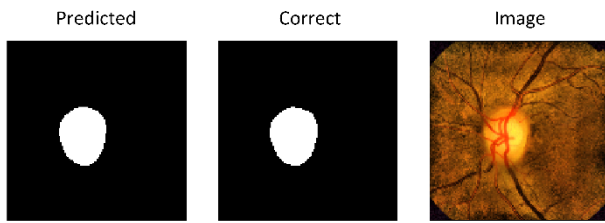
test images for which the radius of the predicted disc has less than a 10% error when compared with the ground truth mean radius. As an example, when using the deepest network included in the table, for 97% of the images our estimation of the disc radius has an error smaller than 10% (RRP=97).

The base case is the only architecture in the table where we don't use batch normalization. When training on a single dataset, batch normalization has a moderate effect on the network performance. However, when training using multiple datasets, we see a clear overfitting effect when we don't use batch normalization. This can be clearly seen in Fig. 6, where the learning curve on the left side, which corresponds to a network without batch normalization, clearly overfits the data, while the curve on the right side, which corresponds to the same system with batch normalization, shows much better results for both training and testing sets.

We can see that deep networks with few parameters like the 6/40/Y/1.1, which has only 917492 trainable parameters, achieve good results for disc segmentation. In this specific case, the CNN achieves a RRP of 95. The best and the worst segmentations for this network are shown in Fig. 7. This network is highlighted in the table.

As a reference, we include in Table 1 a very wide and deep network (7/64/Y/1.3) which has over 14 million trainable

Best case: Image #00  
Dice coefficient: 0.9891



Worst case: Image #46  
Dice coefficient: 0.6262

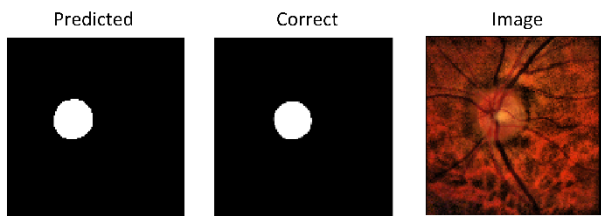


FIGURE 7. Best and worst disc segmentation with 6/40/Y/1.1 net.

parameters. Although the performance of this network is better than in any other case, the small improvement does not justify the additional complexity of the network. We also highlight this case in the table.

We can consider the effects of dynamic data augmentation by training the system without using the dual image generator. If we only use the images with contrast and brightness modifications, the prediction results are significantly worse. As an example, if we use this approach for our base case (with batch normalization), the worst-case Dice is 38% and the RRP falls to 86%.

We have not included the training time in the table as, in our case, this is almost independent of the network complexity. In all our experiments the training time was between 25 and 28 minutes. This seems to be caused by the dynamic data augmentation implemented in the dual image generator. As TensorFlow TPU support is currently not well documented, we initially considered the possibility that the generator might be running on CPU. The generator must produce the image batches that are used for training and testing.

Currently colab notebooks run on an Intel(R) Xeon(R) CPU @ 2.30 GHz using a single core with two threads [26]. In our training experiments we train for 15 epochs with 150 train and 30 test batches per epoch. As we use 120 image batches, we have to generate 270000 training images and 54000 testing images. This represents less than 5ms per generated image. Changing our code so that we don't use dynamic data augmentation produces only slightly better training times for a similar number of training and testing images. Thus, it seems that the training time is not dominated by this factor as we originally supposed.

We provide a GPU version of our notebook in GitHub to allow the calculation of the TPU/GPU speedup. This speedup

has some dependency on the network characteristics. For example, with a 6/40/Y/1.1 network we get a 2.5 training speed improvement, while for a 5/32/Y/1.5 we get a 2.2 speedup. As Keras support for TPUs is in early beta stage, performance comparisons will surely change in the future. Many architectures can't be trained with our default batch size on the Tesla T4 GPUs in Google colab due to memory limitations. In these cases, the speedup training on TPUs can be above 3.0. We finish the architecture name in tables 1 and 2 with a T if the architecture must be trained on TPUs to keep the 120 image batches, and with a G otherwise.

As already mentioned in the cup case, we start by selecting the disc area. After this, the segmentation process is identical to the one used for disc segmentation. In Table 2 we show the mean Dice coefficient for the training and testing sets, the Dice coefficient for the best and worst image in the testing set, the standard deviation of the Dice on the testing set, and the Radii ratio parameter. The number of trainable parameters in the network is shown as a reference, although this value is clearly the same as for the Table 1 for the same network architecture.

We can clearly appreciate the importance of RRP for CDR prediction, as alternatives like the base case don't look too bad from other perspectives but they are not able to predict the radius and, thus, the CDR correctly in a significant portion of the cases.

In Fig. 8 we can see the best and worst prediction using the 4/72/Y/2.0. After consulting with several ophthalmologists, we believe that, in many cases, discrepancies produced by the larger networks correspond to very difficult cases and they are very similar to the discrepancies found when the same images are analyzed by human experts. In Fig. 9 we can see that even the networks like 4/72/Y/2.0 with over 44 million parameters do not significantly overfit the data.

Some of the most sophisticated models presented in Table 1 fail to get good RRP for the cup case. As an example, the 7/64/Y/1.3 architecture, with over 14 million parameters, only obtains a RRP of 73. Thus, we introduce in Table 2 new architectures for cup detection, but try to keep the number of parameters to a reasonable level. Although we have made trials with very different architectures, in general we got interesting results both from wide architectures with high increment ratios and few layers, and from wide and deep architectures with low increment ratios.

An important aspect is illustrated in Fig. 10. Sometimes the networks predict images that, although they are not too bad when measured using Dice or even RRP, are considered as very bad predictions by ophthalmologists.

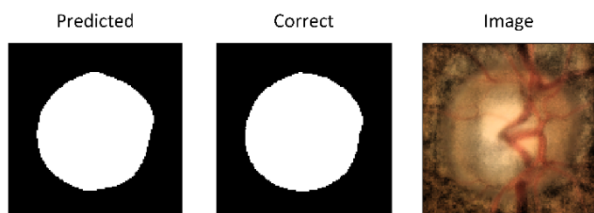
Our objective is to find an architecture where:

- The worst image Dice is above 55, and the test image subjective quality is acceptable for an expert ophthalmologist.
- At least 75% of the images predict the cup ratio with an error smaller than 10%, i.e.  $RRP > 75$ .
- The number of parameters is under 6M.

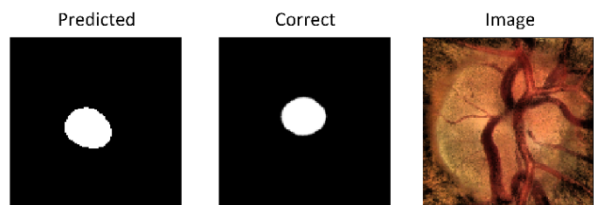
**TABLE 3.** Comparison with existing methods in the literature. Results are percentage dice coefficients.

Author	Method	Cup Drishti	Disc Drishti	Cup RIM-ONE	Disc RIM-ONE	Disc DRIONS	Disc Multi	Cup Multi
Zilly et al. [27]	Ensemble learning based CNN (DL)	87	97	-	-	-		
Al-Bander [3]	Fully convolutional DenseNet	83	95	69	90	94		
Sevastopolsky [2]	Modified U-Net CNN	-	-	82	94	94		
Shankaranarayana et al. [28]	Fully convolutional and adversarial network Generalized U-Net-Multi-dataset train/validation	-	-	94	98	-		
Our work							94	94

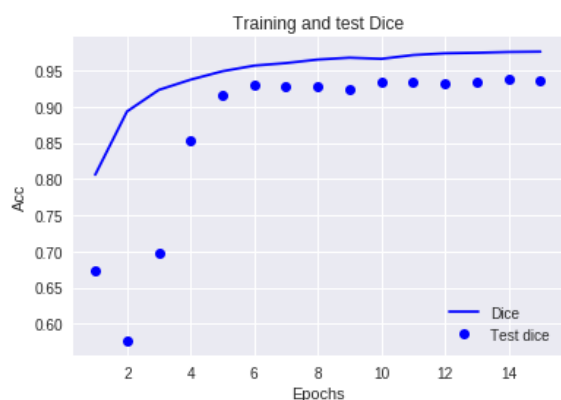
Best case: Image #02  
Dice coefficient: 0.9908



Worst case: Image #56  
Dice coefficient: 0.5706

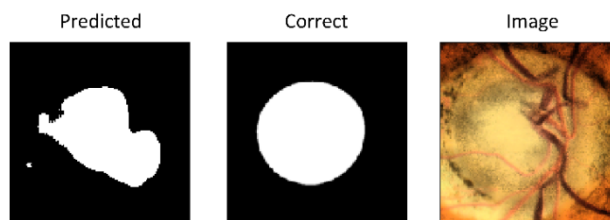


**FIGURE 8.** Best and worst cup segmentation with 4/72/Y/2.0 net.



**FIGURE 9.** Learning curve for 44M parameter 4/72/Y/2.0.

These criteria are met by the 6/72/Y/1.2 and 6/96/Y/1.1 networks. The 5/64/Y/1.3 is quite near the required criteria but, as already shown in Fig. 10, it does not satisfy the subjective quality requirement. Both 6/72/Y/1.2 and 6/96/Y/1.1 produce reasonably good results, but the subjective quality of



**FIGURE 10.** Bad prediction from 5/64/Y/1.3.

the first alternative is slightly better and, thus, we would recommend this choice.

In table 3, we compare our results with those obtained by other researchers. We only include results that are directly related to our proposed solutions, i.e., those works that perform cup and disc segmentation using a deep learning approach. As in the other tables, the results are presented as percentage Dice coefficients. It is important to note that all the other authors train and test with each specific dataset independently. This approach is not suitable for our objective, i.e. providing segmentation as a cloud-based service. In the table, we show that with our training methodology we can get world-class results over the whole group of datasets.

In the case of disc segmentation, we obtain a Dice coefficient of 94%, which is the same as that obtained by [2] for RIM-ONE and DRIONS, and by [3] for DRIONS, but better than the result of [3] for RIM-ONE (90%), and not as good as the results of [27] and [3] for DRISHTI (97% and 95%) and [28] for RIM-ONE (98%). In the case of cup segmentation, our result (94%) is equal to that obtained by [28] for RIM-ONE, and better than the other results from [2], [3] and [27]. This is only a first step, and we should retrain our system when we have further data available from more sources. The high-speed possible with TPU-based training makes this concept feasible in practice.

#### IV. CONCLUSION AND FUTURE WORK

We have been able to show that by using data from different datasets, doing adequate image preprocessing and performing very significant data augmentation (both statically and dynamically), we have been able to perform cup and disc segmentation getting results with a similar performance to

that obtained by other authors using a single dataset for evaluation and testing. This is, at least, a first approximation to the possibility of running this type of segmentation as a service on the cloud.

The use of Pröve's generalized parameterizable recursive U-net model allows to easily train and test any U-Net configuration without having to perform any changes in the code. This allows a much greater flexibility for testing different architectures.

Training on Google Cloud TPUs has allowed to test many different configurations, training them in a time almost independent of the network architecture. To our best knowledge, this is the first time that a U-Net architecture has been tested on TPUs. The speedup obtained with TPUs (2 to 3 times faster than cloud GPU solutions) makes this implementation very attractive for systems like ours, that require periodic retraining.

We have been able to test networks with 4 to 7 contraction and expansion layers, from 32 to 92 input channels, and with layer to layer width ratios from 1.1 to 2.0. The number of parameters has varied from 0.9 to 44 million.

Many U-Net architectures have been proven adequate for disc segmentation. However, only a very small number of alternatives have provided good quality cup segmentation while keeping the number of network parameters at reasonable levels.

The use of the free Google Colaboratory iPython notebook environment together with the public availability of the datasets allows all our experiments to be easily replicable directly from an internet browser without the necessity of powerful local GPU devices.

There are many possibilities for expanding this work in the future related, among other possible topics, to the modifications needed to always produce disc and cup shapes that are acceptable to ophthalmologists, the automation of the architecture configuration parameters and the use of other CNN architectures in parallel to use the produced data for direct Glaucoma detection. It should be noted that in this work we have tried to correctly segment the disc and the cup as this leads to the correct calculation of CDR, that is widely used for glaucoma detection. We have also introduced a new parameter (Radii Ratio parameter- RRP) that is very useful to estimate the accuracy of the CDR.

Although we have used cup and disc segmentation and thus the possible calculation of cup to disc ratio as our objective problem, it is clear that the proposed static and dynamic data augmentation approach, together with the fully parameterizable U-Net and the capabilities of TPU-based training, would allow to use the developed tools, with little modifications, to solve many multidataset binary image segmentation problems.

An important problem with this work has been the experimental nature and lack of documentation of Google's TensorFlow TPU support. In the future, as Keras has become a core component of TensorFlow 2.0, our code should be able to run in this environment with few modifications.

## ACKNOWLEDGMENT

Development in Cloud environment was supported by Google Cloud platform research credit program.

## REFERENCES

- [1] G. Litjens, T. Kooi, B. E. Bejnordi, A. Arindra A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, Dec. 2017. doi: [10.1016/j.media.2017.07.005](https://doi.org/10.1016/j.media.2017.07.005).
- [2] A. Sevastopolsky, "Optic disc and cup segmentation methods for glaucoma detection with modification of U-net convolutional neural network," *Pattern Recognit. Image Anal.*, vol. 27, no. 3, pp. 618–624, Jul. 2017. doi: [10.1134/s1054661817030269](https://doi.org/10.1134/s1054661817030269).
- [3] B. Al-Bander, B. Williams, W. Al-Nuaimy, M. Al-Tae, H. Pratt, and Y. Zheng, "Dense fully convolutional segmentation of the optic disc and cup in colour fundus for glaucoma diagnosis," *Symmetry*, vol. 10, no. 4, p. 87, Mar. 2018. doi: [10.3390/sym10040087](https://doi.org/10.3390/sym10040087).
- [4] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. CVPR*, Salt Lake City, UT, USA, Jun. 2018, pp. 8789–8797. doi: [10.1109/CVPR.2018.00916](https://doi.org/10.1109/CVPR.2018.00916).
- [5] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (Lecture Notes in Computer Science)*, vol. 9351, N. Navab, J. Hornegger, W. Wells and A. Frangi, Eds. Geneva, Switzerland: Springer, 2015, pp. 234–241. doi: [10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28).
- [6] M. Thieme and T. Reina. (2018). *Biomedical Image Segmentation with U-Net. Upsampling versus Transposed Convolution*. Accessed: Apr. 11, 2019. [Online]. Available: [www.intel.ai/biomedical-image-segmentation-u-net#gs.cV253umX](http://www.intel.ai/biomedical-image-segmentation-u-net#gs.cV253umX)
- [7] Sebastian Ruder. (2016). *An Overview of Gradient Descent Optimization Algorithms*. Accessed: Apr. 15, 2019. [Online]. Available: [ruder.io/optimizing-gradient-descent/](http://ruder.io/optimizing-gradient-descent/).
- [8] H. A. Quigley and A. T. Broman, "The number of people with glaucoma worldwide in 2010 and 2020," *Brit. J. Ophthalmol.*, vol. 90, pp. 262–267, Mar. 2006. doi: [10.1136/bjo.2005.081224](https://doi.org/10.1136/bjo.2005.081224).
- [9] J. B. Jonas and A. M. Bron, "Optic disc photography in the diagnosis of glaucoma," in *Glaucoma Medical Diagnosis and Therapy*, 2nd ed., vol. 1, T. M. Shaarawy, M. B. Sherwood, R. A. Hitchings and J.G. Crowston, Eds. New York, NY, USA: Elsevier, 2015, ch. 19, pp. 209–220. doi: [10.1016/b978-0-7020-5193-7.00019-4](https://doi.org/10.1016/b978-0-7020-5193-7.00019-4).
- [10] N. Thakur and M. Juneja, "Survey on segmentation and classification approaches of optic cup and optic disc for diagnosis of glaucoma," *Biomed. Signal Process. Control*, vol. 42, pp. 162–189, Apr. 2018. doi: [10.1016/j.bspc.2018.01.014](https://doi.org/10.1016/j.bspc.2018.01.014).
- [11] A. Aquino, M. Emilio, E. Gegundez-Arias, and D. Marin, "Detecting the optic disc boundary in digital fundus images using morphological, edge detection, and feature extraction techniques," *IEEE Trans. Med. Imag.*, vol. 29, no. 11, pp. 1860–1869, Nov. 2010. doi: [10.1109/tmi.2010.2053042](https://doi.org/10.1109/tmi.2010.2053042).
- [12] D. D. Patil, R. R. Manza, G. C. Bedke, and D. D. Rathod, "Development of primary glaucoma classification technique using optic cup & disc ratio," in *Proc. Int. Conf. Pervasive Comput.*, Pune, India, Jan. 2015, pp. 1–5. doi: [10.1109/PERVASIVE.2015.7087139](https://doi.org/10.1109/PERVASIVE.2015.7087139).
- [13] S. J. Kim, K. J. Cho, S. Oh, "Development of machine learning models for diagnosis of glaucoma," *PLoS ONE*, vol. 12, no. 5, May 2017, Art. no. e0177726. doi: [10.1371/journal.pone.0177726](https://doi.org/10.1371/journal.pone.0177726).
- [14] M. C. V. S. Mary, E. B. Rajasingh, J. K. K. Jacob, D. Anandhi, U. Amato, and S. E. Selvan, "An empirical study on optic disc segmentation using an active contour model," *Biomed. Signal Process. Control*, vol. 18, pp. 19–29, Apr. 2015. doi: [10.1016/j.bspc.2014.11.003](https://doi.org/10.1016/j.bspc.2014.11.003).
- [15] F. Calimeri, A. Marzullo, C. Stamile, and G. Terracina, "Optic disc detection using fine tuned convolutional neural networks," in *Proc. 12th Int. Conf. Signal-Image Technol. Internet-Based Syst.*, Naples, Italy, Dec. 2016, pp. 69–75. doi: [10.1109/SITIS.2016.20](https://doi.org/10.1109/SITIS.2016.20).
- [16] S. Liu, S. L. Graham, A. Schulz, M. Kalloniatis, B. Zangerl, W. Cai, Y. Gao, B. Chua, H. Arvind, J. Grigg, D. Chu, A. Klistorner, and Y. You, "A deep learning-based algorithm identifies glaucomatous discs using monoscopic fundus photographs," *Ophthalmol.*, vol. 1, no. 1, pp. 15–22, Jul./Aug. 2018. doi: [10.1016/j.ogla.2018.04.002](https://doi.org/10.1016/j.ogla.2018.04.002).



[17] J. De Fauw, J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, and H. Askham, "Clinically applicable deep learning for diagnosis and referral in retinal disease," *Nature Med.*, vol. 24, no. 9, pp. 1342–1350, 2018. doi: [10.1038/s41591-018-0107-6](https://doi.org/10.1038/s41591-018-0107-6).

[18] F. Chollet, *Deep Learning with Python*. Shelter Island (Suffolk), NY, USA: Manning, 2018.

[19] Google. (2019). *Cloud TPU System Architecture*. Accessed: Feb. 27, 2019. [Online]. Available: <https://cloud.google.com/tpu/docs/system-architecture>

[20] B. Zoph, E.D. Cubuk, G. Ghiasi, T.-Y. Lin, J. Shlens and Q. V. Le, "Learning data augmentation strategies for object detection," Jun. 2019, *arXiv:1906.11172*. [Online]. Available: <https://arxiv.org/abs/1906.11172>

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: <https://arxiv.org/abs/1704.04861>

[22] F. Fumero, S. Alayón, J. L. Sánchez, J. Sigut, and M. González-Hernández, "RIM-ONE: An open retinal image database for optic nerve evaluation," in *Proc. 24th Int. Symp. Comput.-Based Med. Syst.*, Bristol, U.K., Jun. 2011, pp. 1–6. doi: [10.1109/cbms.2011.5999143](https://doi.org/10.1109/cbms.2011.5999143).

[23] J. Sivaswamy, S. R. Krishnadas, G. D. Joshi, M. Jain, and A. U. S. Tabish, "Drishti-GS: Retinal image dataset for optic nerve head(OH) segmentation," in *Proc. IEEE 11th Int. Symp. Biomed. Imag.*, Beijing, China, May 2014, pp. 53–56. doi: [10.1109/ISBI.2014.6867807](https://doi.org/10.1109/ISBI.2014.6867807).

[24] E. J. Carmona and M. Rincón, J. García-Feijoó, and J. M. Martínez-de-la-Casa, "Identification of the optic nerve head with genetic algorithms," *Artif. Intell. Med.*, vol. 43, no. 3, pp. 243–259, 2008. doi: [10.1016/j.artmed.2008.04.005](https://doi.org/10.1016/j.artmed.2008.04.005).

[25] A. Amidi and S. Amidi. (2019). *A Detailed Example of How to Use Data Generators With Keras*. Accessed: Mar. 26, 2019. [Online]. Available: <http://stanford.edu/shervine/blog/keras-how-to-generate-data-on-the-fly>

[26] Google. (2019). *Colab System Specs*. Accessed: Apr. 11, 2019. [Online]. Available: [https://colab.research.google.com/drive/151805XTDg-dgHb3-AXJCpnWaqRhop\\_2](https://colab.research.google.com/drive/151805XTDg-dgHb3-AXJCpnWaqRhop_2)

[27] J. Zilly, J. M. Buhmann, and D. Mahapatra, "Glaucoma detection using entropy sampling and ensemble learning for automatic optic cup and disc segmentation," *Comput. Med. Imag. Graph.*, vol. 55, pp. 28–41, Jan. 2017. doi: [10.1016/j.compmedimag.2016.07.012](https://doi.org/10.1016/j.compmedimag.2016.07.012).

[28] S. M. Shankaranarayana, K. Ram, K. Mitra, and M. Sivaprakasam, "Joint optic disc and cup segmentation using fully convolutional and adversarial networks," in *Fetal, Infant and Ophthalmic Medical Image Analysis (Lecture Notes in Computer Science)*, vol. 10554, M. Cardoso, Eds. Cham, Switzerland: Springer, 2017, pp. 168–176. doi: [10.1007/978-3-319-67561-9\\_19](https://doi.org/10.1007/978-3-319-67561-9_19).



**JAVIER CIVIT-MASOT** received the degree in biomedical engineering from the Universities of Seville and Malaga, Spain, in 2018. He is currently pursuing the master's degree in intelligent systems for energy and transport with the Higher Polytechnical School, Seville. He is currently with COBER SL, an SME focused on medical instruments and AI-based diagnosis aids. He has also collaborated with the Medical Physics Laboratory, Aristotle University of Thessaloniki, Greece. His current

research interests include deep learning algorithms for diagnostic support, medical instrumentation design, and finite element analysis of intestinal flow.



**FRANCISCO LUNA-PEREJÓN** received the master's degree in computer engineering from the University of Seville, Seville, Spain, in 2019, where he is currently pursuing the Ph.D. degree with the Technical School of Computer Engineering. His Ph.D. thesis is on design and application of algorithms for gait analysis in people with motor disabilities. His current research interest includes handcrafted and machine learning algorithms for diagnostic support and healthcare system design.



**SATURNINO VICENTE-DÍAZ** received the master's degree in computer science and the Ph.D. degree from the University of Seville, Spain, in 1996 and 2001, respectively. Since 2010, he has been an Associate Professor with the University of Seville and also the Vice Dean of the Technical School of Computer Engineering. He has been a Researcher of the Robotics and Computer Technology Group, since 1996. He has authored and coauthored more than 40 articles in refereed

international journals and conferences in the fields of robotics, accessibility, embedded systems, and bioinspired systems. He has participated in more than 20 research projects and contracts, such as the EU projects FLEX, CAVIAR, and CARDIAC.



**JOSÉ MARÍA RODRÍGUEZ CORRAL** received the master's degree in computer engineering and the Ph.D. degree from the University of Seville, Spain, in 1993 and 2002, respectively. From 1993 to 1995, he involved in robot control with the Robotics and Computer Technology Research Group, University of Seville. He was an Associate Lecturer with the Avenida Universidad de Cádiz, Spain, from 1995 to 1998, where he is currently an Associate Professor of computer languages and

systems and also a member of the Applied Robotics Group. He has authored various articles and research reports on computer architecture. His current research interests include engineering education, robotics, and bus emulation.



**ANTÓN CIVIT** received the master's degree in physics (electronics) and the Ph.D. degree from the University of Seville, Spain, in 1984 and 1987, respectively. After working for several months with Hewlett-Packard, he joined the University of Seville, where he is currently a Full Professor of computer architecture and the Director of the Robotics and Computer Technology Research Group, School of Computer Engineering. He has authored various articles and research reports on

neuromorphic engineering, computer architecture, rehabilitation technology, and robotics. He has also been a member of the European Commission eAccessibility Expert Group and responsible for several EU projects in the areas of health, neuromorphic engineering, and eAccessibility. He is currently an Associate Editor of *Neuroscience* (Frontiers). His current research interests include deep learning-based medical diagnosis, medical instrumentation, and robotics.

...