applied sciences

MDPI

Article

Detecting Diabetic Retinopathy Using Embedded Computer Vision

Parshva Vora and Sudhir Shrestha *

Intelligent Systems Laboratory, Department of Engineering Science, Sonoma State University,
Rohnert Park, CA 94928, USA; vorap@sonoma.edu
* Correspondence: sudhir.shrestha@sonoma.edu; Tel.: +1-707-664-4385

Received: 4 September 2020; Accepted: 13 October 2020; Published: 17 October 2020

check for updates

Featured Application: A low-cost solution for detecting diabetic retinopathy for the medically underserved population who do not have access to regular eye exams and are at increased risk of diabetes-related blindness.

Abstract: Diabetic retinopathy is one of the leading causes of vision loss in the United States and other countries around the world. People who have diabetic retinopathy may not have symptoms until the condition becomes severe, which may eventually lead to vision loss. Thus, the medically underserved populations are at an increased risk of diabetic retinopathy-related blindness. In this paper, we present development efforts on an embedded vision algorithm that can classify healthy versus diabetic retinopathic images. Convolution neural network and a k-fold cross-validation process were used. We used 88,000 labeled high-resolution retina images obtained from the publicly available Kaggle/EyePacs database. The trained algorithm was able to detect diabetic retinopathy with up to 76% accuracy. Although the accuracy needs to be further improved, the presented results represent a significant step forward in the direction of detecting diabetic retinopathy using embedded computer vision. This technology has the potential of being able to detect diabetic retinopathy without having to see an eye specialist in remote and medically underserved locations, which can have significant implications in reducing diabetes-related vision losses.

Keywords: diabetic retinopathy; embedded computer vision; low-cost solution for under-served populations; convolution neural network; k-fold cross-validation

1. Introduction

Diabetic retinopathy (DR) is caused by damage to the blood vessels in the tissue at the back of the eye (retina) causing vision impairment and blindness. Uncontrolled blood sugar is a major risk factor. Since diabetic retinopathy lacks early symptoms, it is very difficult to detect the diseases at an early stage [1]. The 2020 diabetes report from the Center for Disease Control (CDC) found that the US diabetic population rose to 9.4% or 3.3 million people [2]. Another 88 million people, more than 1 out of 3 adults, have pre-diabetes. When left untreated, pre-diabetes can lead to type-2 diabetes within five years. The diabetic population worldwide was more than 250 million in 2017 [3]. This number is projected to rise to around 629 million by 2045 [4]. An estimated one-third of the diabetic population has diabetic retinopathy symptoms, with a significant portion of them being vision-threatening [5]. Currently, 7.7 million Americans are impacted by this disease, and it is expected to rise to 11.3 million by 2030 [6]. One of the major contributors of increased diabetic retinopathy-related vision loss is the lack of access to the medical care needed to catch the disease at an early stage.

The current diabetic retinopathy detection method involves a dilated eye exam. In this exam, eye-dilating drops placed inside a patient's eye widen eye pupils and allow doctors to see the eye blood vessels [7,8]. A special dye is injected, and pictures are taken as the dye circulates through blood vessels. The images are used to further examine the blood vessels and catch any damaged veins or fluid leaks. These eye exams are very effective; however, for patients without health insurance, it costs $200 or more in the US and is often unavailable in remote or developing parts of the world [9,10].

The detection of diabetic retinopathy using computer vision has recently been considered as a possible alternative to a visual analysis by a clinician. One such example is the Kaggle competition on diabetic retinopathy, in which more than 600 teams participated [11]. Similarly, Lam et al. present an automated detection of diabetic retinopathy using TensorFlow and GoogLeNet in [12]. Different aspects of pre-processing to feature extraction are presented. In [13], Wu et al. presents the classification of diabetic retinopathy using convolution neural network. Using a self-gated soft-attention mechanism and pre-trained coarse network, the presented work performs four-class hierarchical classification based on the severity of the disease. The detection of diabetic retinopathy using deep learning is also presented in [14–17]. In another approach, hardware-based solutions have also been reported [18]. With a digital signal processing kit developed by Texas Instrument, the article reports the possibility of using hardware for detecting diabetic retinopathy. These studies were conducted using high-resolution retina images taken with a fundus camera. Efforts to detect retinopathy from images taken with a low-cost camera have also been reported [19].

This manuscript furthers the reported work through a solution toward detecting diabetes retinopathy using embedded computer vision for implementation on a cost-effective portable NVIDIA Jetson TX2 hardware [20] and on-device real-time classification without the need for an Internet connection. We present training and testing of the Caffe and Keras model using healthy and diabetic retinopathy images. The training and testing set consists of 83,000 high-resolution retina images labeled by experts. The Caffe model was trained and tested on NVIDIA-embedded hardware.

2. Materials and Methods

2.1. Training and Testing Dataset

Diabetic retinopathy is generally divided into five groups: Normal, Mild DR, Moderate DR, Severe DR, and Proliferative DR (Table 1). Diseases start with small changes in blood vessels, which are designated as Mild DR. At this stage, a complete recovery is possible. If proper care is not taken, in a few years, it will progress to Moderate DR, where leakage in blood vessels may begin. Then, the diseases progress further to Severe and Proliferative DR and may lead to complete vision loss.

Table 1. Various stages of diabetic retinopathy [21].

Time Frame in Years	**0**	**3–5**	**5–10**	**10–15**	**>15**
Stages of DR	Normal	Mild DR: Stage-1 non-proliferate	Moderate DR: Stage-2 non-proliferate	Severe DR: Stage-3 non-proliferate	Proliferate DR: Stage-4 proliferate
Changes in Retina	No retinopathy	A few small bulges in the blood vessels.	A few small bulges in the blood vessels. Spots of blood leakage. Deposits of cholesterol.	Larger spots of blood leakages. Irregular beading in veins. Growth of new blood vessels at the optic disc. Blockage of blood vessels.	Beading in veins. Growth of new blood vessels elsewhere in the retina. Clouding of vision. Complete vision loss.

To predict DR with a higher accuracy using a machine learning algorithm, a large amount of training data are needed. The data need to come from reliable sources with accurate labels. We used the Kaggle dataset, which was provided by EyePacs [22]. EyePacs screened more than 750,000 patients and collected 5 million retina images [22]. The Kaggle dataset contains 35,126 images for training and another 53,594 for testing. The size of the images ranged from 360 KB to 2 MB. However, only a few images were below 500 KB. The Kaggle dataset is one of the largest datasets of diabetic retinopathy images currently available. Table 2 below shows the number of images in each DR category among the training and testing datasets. Figure 1 shows samples images in different classes representing different stages of DR. The Kagle database provided images of all DR categories in one folder and a comma separate value (CSV) file with a description of each image category. For training and testing, the images need to be separated and placed in separate folders. A script was written to separate the images based on CSV labels, which is shown below. Next, the images were cropped using the Otsu method to isolate the main features [23,24]. Furthermore, images were normalized and contrast adjusted using a filtering algorithm. Data augmentation was also performed to improve the diversity of data. Different operations of padding, cropping, and flipping were also performed.

```
while IFS = ',' read -r f1 f2
do
imageEye = $(echo $f2 | cut -d '_' -f 2)
if [$imageEye == 'left' -a $f2 == '1' ]
then
echo 'done left'
else
echo 'done right'
fi
done < "file_name"
```

Table 2. The number of images in each diabetic retinopathy (DR) category for training and testing.

DR Category/No. of Images	Training		Testing	
	Left Eye	Right Eye	Left Eye	Right Eye
Normal (No DR)	12,871	12,939	19,717	19,816
Mild DR	1212	1231	1905	1875
Moderate DR	2702	2590	3957	3904
Severe DR	425	448	613	601
Proliferate DR	353	355	596	610

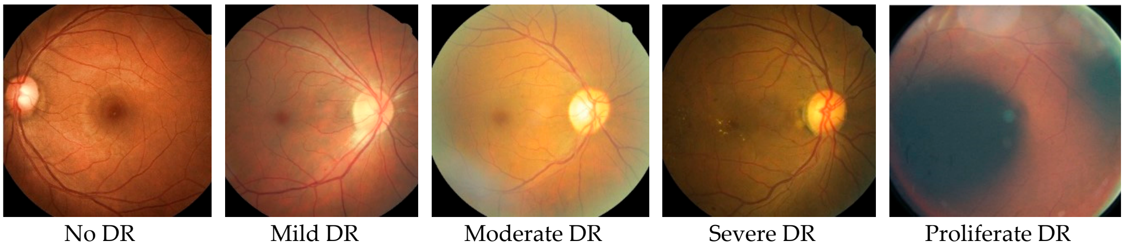

Figure 1. Sample images from the five different image classes showing various DR stages: No DR, Mild DR, Moderate DR, Severe DR, and Proliferate DR. As the disease progresses, blood vessels expand, slowly distorting vision over time.

2.2. Convolution Neural Network

A simple neural network with a set of input, multiple hidden layers, and an output layer was used. Each hidden layer is similar to a neuron with a set of inputs and weighted outputs, which produces output with the help of an activation function [25–29]. Convolution neural network (CNN) learns such features on the patterns from the input training images when those patterns occur repeatedly in the input data. In diabetic retinopathy, those features are distortions in blood vessels, with macula occurring in the retina. When these features occur repeatedly and CNN sufficiently learns to classify them, they form a model. Then, this model is tested with a new dataset for accuracy.

A filter or convolution kernel is a matrix that is convolved with the input image to detect specific features. Consider an example where the input image is 4 × 4. A 3 × 3 filter is applied to the image, resulting in an output. This process is repeated by moving the filter by one column, generating a second output. The initial choice of the filter size is random and is optimized based on the accuracy. The learning rate determines the step size at each iteration. It is important to have an optimum learning rate because a too-small learning rate will slow down the convergence, and a too-high learning rate may make a learning jump over minima. To determine an optimum learning rate, the system is trained with different learning rates, typically starting from 0.001, and a rate that results in the highest accuracy is chosen. Activation functions are mathematical equations that determine the output of a neural network [29–31]. Without activation functions, the output of the neural network is linear. An epoch is when one scans the complete dataset in one pass through the neural network in either a forward or backward direction. The number of epochs impacts the performance of a system. If a dataset is very large and the number of epochs is chosen incorrectly, the time to train the system increases. Thus, it is important to choose an optimum epoch value. A hidden layer is an intermediate layer between the input and output layers, where a set of weighted inputs produces an output through an activation function [32].

The CNN architecture consists of convolution and pooling layers. The convolution layer is a core layer of a neural network and performs convolution operation (point-wise or depth-wise). The pooling layer is used to reduce the spatial size of convolved features. There are two types of pooling layers: max pooling and average pooling. A fully connected layer, following the pooling layer, holds composite and aggregated information and is used to predict the output as either healthy or unhealthy.

2.3. Comparison of Various Tools

GoogLeNet is a 22 layers deep CNN architecture developed by Google. In 2014, GoogleNet won a competition for visual recognition with a 6.7% error rate. The network was trained on 1000 object categories. It can be retrained to perform new tasks using transfer learning [33]. TensorFlow is an open-source library and platform to develop end-to-end solutions for machine learning. It provides an inbuilt application programming interface (API) that supports different machine learning algorithms. We initially used TensorFlow as one of the platforms for faster development [34]. We implemented our project on two platforms, Keras and NVIDIA Jetson. Keras helps to integrate lower-level deep learning languages such as TensorFlow.

2.4. Nvidia Jetson TX2 and Nvidia Digits

We chose NVIDIA Jetson TX2 as our hardware platform because one of our primary goals was to process the data locally and in real time so that it can be used to classify retina images without the need for an Internet connection [20]. The features provided by the NVIDIA Jetson enables producing classification results in a reasonable timeframe for a practical application and allows for future on-device retraining and improvement of the model. This is important, as our aim is to enable use of the device at remote locations where an Internet connection may not be available or reliable. However, it must be noted that it is possible to implement the presented model on a microcontroller and CPU-based hardware platform. The performance and efficiency of Jetson are equivalent to an Intel Xeon E5. In a direct comparison of Jetson with Intel Xeon Server in running a deep learning inference model based on a GoogleNet deep learning image recognition network, Jetson was able to process 290 images per second compared to 231 by the Intel Xeon Server with a 128 batch size [35]. With a few modifications, and following the instruction provided by NVIDIA, we installed, set up, and ran an NVIDIA deep learning GPU training system (DIGITS), including installing Nvidia drivers, installing Docker, and setting up and starting a DIGITS Container [36]. NVIDIA DIGITS is an open-source software provided to design, train, and visualize deep neural networks for image classification, segmentation, and object detection using Caffe, Torch, and TensorFlow. Docker provides an easy tool to create, deploy, and run applications using containers [37]. Caffe is a deep learning framework that provides processing speed and expressive architecture for developing machine learning models.

Once the setup was complete, the training data were imported to DIGITS, classification was selected based on desired output labels of healthy or unhealthy. Lightning memory-mapped database (LMDB), a memory-mapped file, was chosen as a database backend for its faster input/output performance over a large dataset. Next, an inference model was created using the solver options shown in Table 3 below. Once the model is trained, it outputs epochs and accuracy. The trained model can be tested with testing images.

Table 3. Solver options used to optimize hyperparameters.

Solver Option	Sample Value
Training Epochs	30
Snapshot Interval (in epochs)	1
Validation Interval (in epochs)	1
Batch Size	32
Solver Option	Stochastic gradient descent (SGD)
Base Learning Rate	0.01

2.5. Training and Testing Procedure

The training and testing were conducted with Caffe and Keras models. Caffe was chosen because of its direct support in NVIDIA DIGITS. Keras was chosen because it allowed easy optimization to improve system accuracy. For the Caffe GoogleNet model in NVIDIA DIGITS, the LMDB format was used. In the Keras model, hierarchical data format version 5 (HDF5) was used. HDF5 provides a simple format to read/write over a large dataset. It is easy to add data to a dataset without creating copies. In addition, it supports different languages such as R, C, and Fortran, making it easier to use across various platforms. Once the dataset was processed and converted into the respective file format, it was processed separately with Caffe and Keras models. Once both the models were trained using training images, they were tested with a testing dataset of 53,594 images. The K-fold cross-validation method was used in this study. In k-fold cross-validation, the complete dataset is randomly divided into k folds or groups. One of the groups is used for testing and the remaining groups are used for validation. This process is repeated for all the groups until the complete dataset is trained [38]. In each iteration of k-fold cross validation, the data were spilt into a training and testing dataset, with 30% for testing and 70% for training. We tested for k = 5, 10, and 12. We found that k = 5 was optimal.

3. Results and Discussion

Results from data preprocessing, hyperparameter optimization, and test results are presented and discussed in this section.

3.1. Data Preprocessing

The images were processed in two formats, LMDB in NVIDIA Jetson and HDF5 in Google Engine. The use of RAM, number of CPUs and cores, and processing time were compared. The results are shown in Table 4. Time taken to convert data in the HDF5 file format was comparable to LMDB.

Table 4. Data preprocessing resource comparison between lightning memory-mapped database (LMDB) and hierarchical data format version 5 (HDF5).

Resources	LMDB [NVIDIA Jetson TX2]	HDF5 [Google Engine]
RAM	8 GB	16 GB
CPU	2	1
Cores	4	1
Processing Time	2 Hours 37 Minutes	2 Hours 33 Minutes

3.2. Hyperparameters Optimization

Techniques implemented during the training process were discussed in the previous section. In addition to those techniques, hyperparameters were optimized to improve accuracy. The initial selections of the hyperparameters in Table 3 were optimized through an iterative process to improve accuracy. The optimized parameters are shown in Table 5.

Table 5. Optimized hyperparameters.

Hyperparameters	Tested Value Being Used
Training Epochs	100
Random Seed	100
Batch Size	32
Solver Options	Gradient Descent
Base Learning Rate	0.001

3.3. Accuracy Rate for Inference Model

The trained Caffe and Keras models were separately tested with 53,594 test images that included No DR, Mild DR, Moderate DR, Severe DR, and Proliferate DR. Then, the predictions were aggregated to separate as Healthy and Unhealthy. A No DR result is Healthy, whereas a Mild DR to Proliferate DR is Unhealthy. For example, for a Mild DR test image, a No DR result is counted as Healthy classification and Moderate DR is counted as an Unhealthy classification. We chose these two classifications because of our end intention for the envisioned device, which is to give the healthcare professional conducting the test and the patient one of two specific types of feedback: “The eye appears to be healthy. Please consult an eye specialist if there are any concerns,” or “A visit to an eye specialist is recommended. The system detects diabetic retinopathy symptoms.” In addition, because of the higher risk of a false negative result, specificity is very important for this application.

3.3.1. Caffe Model

The classification results with the trained Caffe model developed using NVIDIA Jetson are shown in Table 6 below. As discussed above, results were aggregated to two classes: Healthy, which means a patient does not have diabetic retinopathy (No DR), and Unhealthy, which implies that a patient has diabetic retinopathy (Mild, Moderate, Severe, or Proliferate DR). The input image types to the model are indicated in the table. As discussed previously, the diabetic retinopathy generally progresses from mild to proliferating over time. The result shows that the model accurately predicts healthy images 98% (specificity or true negative) of the time. The best result for images with DR was observed for the Proliferate DR. The model accurately classified 40% (sensitivity or true positive) of images as unhealthy. The least accurate results were with Mild DR images. The models inaccurately classified 97% of Mild DR images as healthy. We observed that the sensitivity increases as the severity of the disease increases. The combined sensitivity for severe and proliferate DR was 35%. The overall accuracy of the model is 75.6%. We can see that it is highly influenced by the higher specificity value of the model and a large number of No DR images compared to other classes.

Table 6. Results from the Caffe model.

Test Images		Classification	
Type	**No. of Images**	**Healthy**	**Unhealthy**
No DR	39,533	98%	2%
Mild DR	3780	97%	3%
Moderate DR	7861	90%	10%
Severe DR	1214	70%	30%
Proliferate DR	1206	60%	40%

3.3.2. Keras Model

Table 7 below shows the classification results with the Keras model. Similar to the case of the Caffe model, we aggregated the classification results into Healthy (No DR) and Unhealthy (Mild to Proliferate DR). The input image types to the model are indicated in the table. The model accurately predicted healthy images 97% of the time (specificity or true negative). The best result for images with DR was again observed for the Proliferate DR. The model accurately classified 55% of images as unhealthy (sensitivity or true positive). The least accurate results were with Mild and Moderate DR images. The models inaccurately classified 97% of Mild and Moderate DR images as Healthy. In addition, with the Keras model, the sensitivity increased as the severity of the disease increased. The combined sensitivity for Severe and Proliferate DR was 48%. The overall accuracy of this model was 76.7%. In addition, in this case, we observe that the accuracy is influenced by the higher specificity value and a large number of No DR images compared to other classes. For moderate to proliferate DR, the sensitivity for the Keras model increased significantly compared to the Caffe model. For Proliferate DR images, the sensitivity for the Keras model was 55% compared to 40% for the Caffe model.

Table 7. Results from the Keras model.

Test Images		Classification	
Type	**No. of Images**	**Healthy**	**Unhealthy**
No DR	39,533	97%	3%
Mild DR	3780	97%	3%
Moderate DR	7861	80%	20%
Severe DR	1214	59%	41%
Proliferate DR	1206	45%	55%

The least accurate results both for Caffe and Keras models were with Mild and Moderate DR images. We believe this is due to the fact that Mild and Moderate DR are the initial stages of retinopathy, and they are less distinguishable from healthy (No DR) compared to Severe and Proliferate DR. The accuracy of the presented models need to be further improved for practical application. Methods that may be used to improve the accuracy include removing low-quality images, using the local spectrum analysis method [39], or enhancing local contrast using contrast limited adaptive histogram [40]. The accuracy can also be improved by optimizing the presented model or considering alternative models such as recurrent neural network (RNN). The ability of RNN to accumulate features or memorize from past inputs can be advantageous in DR image classification [41,42].

4. Conclusions

We presented the training and testing of the Caffe and Keras model using retina images of healthy and at various stages of diabetic retinopathy. A total of 35,000 high-resolution images labeled by experts were used for training of the models. Another 53,000 images also labeled by experts were used for testing of the trained models. The classification results were aggregated as healthy and unhealthy. The unhealthy images included mild to proliferate diabetic retinopathy. The Caffe model classified healthy and unhealthy images with 75.6% accuracy and 98% specificity. The sensitivities for severe and proliferate diabetic retinopathy images were 30% and 40%, respectively. The Keras model classified healthy and unhealthy images with 76.7% accuracy and 97% specificity. However, the sensitivities for severe and proliferate diabetic retinopathy images significantly improved to 41% and 55%, respectively. The future goal of this research is to develop low-cost hardware capable of on-site real-time classification of retina images. As a result of the nature of the intended application, the specificity (true negative) needs to be close to 100% to reduce the risks associated with a wrong classification result. In addition, to reduce unnecessary tests and associated costs, the sensitivity of the system has to be further improved for a practical application. This can be achieved by further optimizing the presented model or considering alternative models such as recurrent neural network. The Caffe model was trained and tested on an NVIDIA embedded hardware. The Keras model has shown more promise and will be implemented on the same hardware, and the accuracy will be further improved. Although the accuracy needs to be further improved, the presented results represent a significant step forward in the direction of detecting diabetic retinopathy using embedded computer vision for implementation in portable low-cost hardware. This technology has the potential of being able to detect diabetic retinopathy without having to see an eye specialist in remote and medically underserved locations, which can have significant implications in reducing diabetes-related vision losses in the future.

Author Contributions: Both P.V. and S.S. contributed to writing and editing the paper, research design and results analysis, experimental work was done by P.V. under the supervision of S.S., and S.S. was responsible for project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank the support and resources provided by the Intelligent Systems Lab and the Department of Engineering Science at Sonoma State University.

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

References

1. Mayo Clinic. Diabetic Retinopathy. Available online: https://www.mayoclinic.org/diseases-conditions/diabetic-retinopathy/symptoms-causes/syc-20371611 (accessed on 20 June 2020).
2. *National Diabetes Statistics Report: Estimates of Diabetes and Its Burden in the United States, Centers for Diabetes Control and Prevention*; Department of Health and Human Services: Atlanta, GA, USA, 2020.
3. Timmons, J. Diabetes: Facts, Statistics, and You. Available online: https://www.healthline.com/health/diabetes/facts-statistics-infographic (accessed on 20 June 2020).
4. Cheloni, R.; Gandolfi, S.A.; Signorelli, C.; Odone, A. Global Prevalence of Diabetic Retinopathy: Protocol for a Systematic Review and Meta-Analysis. *BMJ Open* **2019**, *9*. [CrossRef] [PubMed]
5. Lee, R.; Wong, T.Y.; Sabanayagam, C. Epidemiology of Diabetic Retinopathy, Diabetic Macular Edema and Related Vision Loss. *Eye Vis.* **2015**, *2*, 17. [CrossRef] [PubMed]
6. Genentech. Retinal Diseases Fact Sheet. Genentech: Breakthrough Science. One Moment, One Day, One Person at a Time. Available online: https://www.gene.com/stories/retinal-diseases-fact-sheet (accessed on 20 June 2020).
7. Mayo Clinic. Diabetic Retinopathy: Overview. Available online: https://g.co/kgs/WTnvDF (accessed on 20 June 2020).
8. Mayo Clinic. Diabetic Retinopathy: Diagnosis. Available online: https://www.mayoclinic.org/diseases-conditions/diabetic-retinopathy/diagnosis-treatment/drc-20371617 (accessed on 20 June 2020).
9. Getting an Eye Exam Without Insurance: What to Expect (Costs and More), Nvision. Available online: https://www.nvisioncenters.com/insurance/eye-exam/ (accessed on 10 July 2020).
10. How Much Does an Intravenous Fluorescein Angiography Cost Near Me? MDsave. Available online: https://www.mdsave.com/procedures/intravenous-fluorescein-angiography/d482fbcc (accessed on 10 July 2020).
11. Diabetic Retinopathy Detection, Identify Signs of Diabetic Retinopathy in Eye Images, Kaggle. Available online: https://www.kaggle.com/c/diabetic-retinopathy-detection (accessed on 18 January 2020).

12. Lam, C.; Yi, D.; Guo, M.; Lindsey, T. Automated Detection of Diabetic Retinopathy Using Deep Learning. AMIA Joint Summits on Translational Science Proceedings, 18 May 2018. Available online: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5961805/ (accessed on 18 January 2020).
13. Wu, Z.; Shi, G.; Chen, Y.; Shi, F.; Chen, X.; Coatrieux, G.; Yang, J.; Luo, L.; Li, S. Coarse-to-fine classification for diabetic retinopathy grading using convolutional neural network. *Artif. Intell. Med.* **2020**, *108*, 101936. [CrossRef] [PubMed]
14. Gargeya, R.; Leng, T. Automated Identification of Diabetic Retinopathy Using Deep Learning. *Ophthalmology* **2017**, *124*, 962–969. [CrossRef] [PubMed]
15. Gulshan, V.; Peng, L.; Coram, M.; Stumpe, M.C.; Wu, D.; Narayanaswamy, A.; Venugopalan, S.; Widner, K.; Madams, T.; Cuadros, J.; et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA Innov. Healthcare Deliv.* **2016**, *316*, 2402–2410. [CrossRef] [PubMed]
16. Alfian, G.; Syafrudin, M.; Fitriyani, N.L.; Anshari, M.; Stasa, P.; Svub, J.; Rhee, J. Deep Neural Network for Predicting Diabetic Retinopathy from Risk Factors. *Mathematics* **2020**, *8*, 1620. [CrossRef]
17. Ting, D.S.W.; Cheung, C.Y.-L.; Lim, G. Development and Validation of a Deep Learning System for Diabetic Retinopathy and Related Eye Diseases Using Retinal Images from Multiethnic Populations with Diabetes. *JAMA Orig. Investig.* **2017**, *318*, 2211–2223. [CrossRef] [PubMed]
18. Datta, N.S.; Banerjee, R.; Dutta, H.S.; Mukhopadhyay, S. Hardware Based Analysis on Automate Early Detection of Diabetic-Retinopathy. *Proc. Technol.* **2012**, *4*, 256–260. [CrossRef]
19. Zalewski, S. Earlier Detection of Diabetic Retinopathy with Smartphone AI, Health News, Medical Breakthroughs & Research for Health Professionals, 29 April 2019. Available online: https://labblog.uofmhealth.org/health-tech/earlier-detection-of-diabetic-retinopathy-smartphone-ai (accessed on 18 January 2020).
20. NVIDIA Jetson TX2. Available online: https://developer.nvidia.com/embedded/jetson-tx2-developer-kit (accessed on 18 January 2020).
21. Diabetic Retinopathy. American Optometric Association. Available online: https://www.aoa.org/healthy-eyes/eye-and-vision-conditions/diabetic-retinopathy?sso=y (accessed on 18 January 2020).
22. News, EyePACS, 9 November 2018. Available online: http://www.eyepacs.com/blog/news (accessed on 18 January 2020).
23. Wolff, C. Using Otsu's Methods to Generate Data for Training of Deep Learning Image Segmentation Models, SCE Developer, 17 May 2018. Available online: https://devblogs.microsoft.com/cse/2018/05/17/using-otsus-method-generate-data-training-deep-learning-image-segmentation-models/ (accessed on 10 June 2019).
24. Omer, A.M.; Elfadil, M. Preprocessing of Digital Mammogram Image on Otsu's Threshold. *Am. Sci. Res. J. Eng. Technol. Sci.* **2017**, *37*, 220–229.
25. Raycad, Convolutional Neural Network (CNN), Medium. Available online: https://medium.com/@raycad.seedotech/convolutional-neural-network-cnn-8d1908c010ab (accessed on 10 June 2019).
26. Amidi, A.; Amidi, S. Convolutional Neural Networks Cheat Sheet. Stanford University. Available online: https://stanford.edu/~{}shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks (accessed on 15 June 2019).
27. CS231n Convolutional Neural Networks for Visual Recognition, GitHub. Available online: https://cs231n.github.io/convolutional-networks/ (accessed on 16 June 2019).
28. Saha, S. A Comprehensive Guide to Convolutional Neural Networks—The ELI5 Way, Medium, 15 December 2018. Available online: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (accessed on 15 June 2019).
29. Kim, S. A Beginner's Guide to Convolutional Neural Networks (CNNs), Medium, 15 February 2019. Available online: https://towardsdatascience.com/a-beginners-guide-to-convolutional-neural-networks-cnns-14649dbddce8 (accessed on 15 June 2019).
30. Leonel, J. Hyperparameters in Machine/Deep Learning, Medium, 7 April 2019. Available online: https://medium.com/@jorgesleonel/hyperparameters-in-machine-deep-learning-ca69ad10b981 (accessed on 16 June 2019).
31. Geva, 7 Types of Activation Functions in Neural Networks: How to Choose? MissingLink.ai. Available online: https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/ (accessed on 16 June 2019).

32. What Is a Hidden Layer?—Definition from Techopedia, Techopedia.com. Available online: https://www.techopedia.com/definition/33264/hidden-layer-neural-networks (accessed on 16 June 2019).
33. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
34. TensorFlow. Available online: https://www.tensorflow.org/ (accessed on 16 June 2020).
35. Franklin, D. NVIDIA Jetson TX2 Delivers Twice the Intelligence to the Edge, NVIDIA Developer Blog, 7 March 2017. Available online: https://devblogs.nvidia.com/jetson-tx2-delivers-twice-intelligence-edge/ (accessed on 22 September 2020).
36. DIGITS Workflow. Available online: https://github.com/dusty-nv/jetson-inference/blob/master/docs/digits-workflow.md (accessed on 17 June 2020).
37. NVIDIA Docker: GPU Server Application Deployment Made Easy. Available online: https://developer.nvidia.com/blog/nvidia-docker-gpu-server-application-deployment-made-easy/ (accessed on 17 June 2020).
38. Cross-Validation: Evaluating Estimator Performance. Scikit-Learn. Available online: https://scikit-learn.org/stable/modules/cross_validation.html (accessed on 20 September 2020).
39. Zhou, W.; Wu, H.; Wu, C.; Yu, X.; Yi, Y. Automatic Optic Disc Detection in Color Retina Images by Local Feature Spectrum Analysis. *Comput. Math. Methods Med.* **2018**, *2018*, 1942582. [CrossRef] [PubMed]
40. Saalfeld, S. Ehcance Local Ontrast (CLAHE). ImageJ, 1 September 2010. Available online: https://imagej.net/Enhance_Local_Contrast_(CLAHE) (accessed on 1 October 2020).
41. Hesamian, M.H.; Jia, W.; He, X.; Kennedy, P. Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges. *J. Digit. Imaging* **2019**, *32*, 582–596. [CrossRef] [PubMed]
42. Alom, M.Z.; Sasan, M.; Yakopcic, C.; Taha, T.M.; Asari, V.K. Recurrent Residual Convolution Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation. *J. Med. Imaging* **2019**, *6*. [CrossRef]

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.