An approach to classification of data with highly localized unmarked features using neural networks

Rafał Grzeszczuk

Department of Computer Science, Electronics and Telecommunication,
AGH University of Science and Technology

Abstract. To face the world's needs, cutting edge automation technology is being applied in such demanding areas as medical imaging. This paper proposes a novel approach to classification problems on datasets with sparse, highly localized features. It is based on the use of saliency map in amplification of features. Unlike previous efforts, this approach does not use any prior information about feature localization. We present an experimental study based on Diabetic Retinopathy classification problem, in which our method has shown to achieve over 20% higher accuracy in solving a two-class Diabetic Retinopathy classification problem than a naive approach based solely on residual neural networks. The dataset consists of 35120 images of various quality, inconsistent resolution and aspect ratio.

1 Introduction

Medical image analysis is a complex field of applied computer vision. Due to the implications of incorrect judgement, the correctness requirements are very high. Given the nature of some medical conditions, the features of interest tend to be subtle, especially in early stages of disease, when treatment is easier. It usually implies that the features will be sparsely distributed over the input image and strongly localized in a few places. This makes the problem difficult to solve and usually requires a trained human expert, such as an experienced physician, to make the judgement. Typical problems that fall into this category include, but are not limited to tumour detection in X-ray and Computer Tomography scans [13, 9] or (micro-)stroke detection in Magnetic Resonance Imaging scans [6]. Analyzed in this paper is diabetic retinopathy, often abbreviated as DR [19].

The retina is a tissue responsible for converting visual stimuli into chemical signals, which are subsequently transmitted to and interpreted by the visual cortex located in the occipital lobe of the brain [14]. In patients with type I diabetes, abnormally high concentrations of glucose are found in blood, causing persistent damage to internal organs. As diabetes progresses, high blood sugar levels cause the blood vessels to become damaged and perforated, resulting in small haemorrhages that can damage the retina. This can be detected by a non-invasive medical procedure called ophtamoloscopy, during which the retina

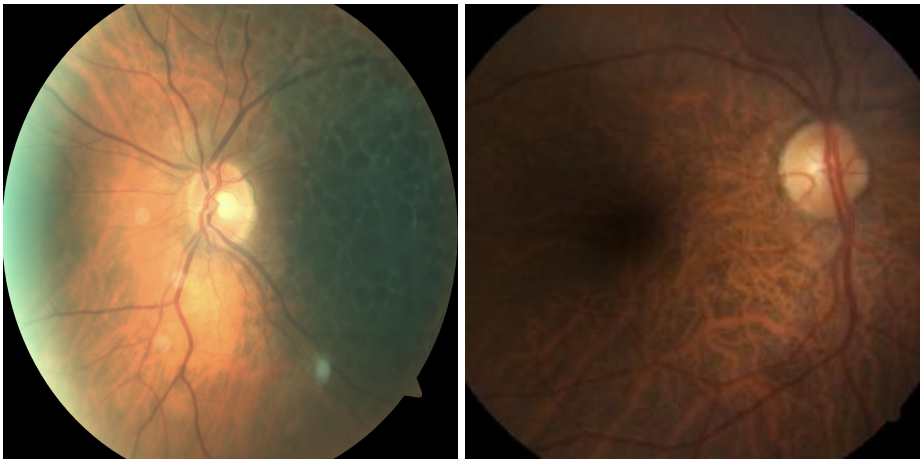

Fig. 1. Example fundi images. The image on the left shows a healthy retina (class 0). The image on the right is an example of proliferative retinopathy (class 4).

is examined (or photographed) through the pupil with an optical device. A significant bottle-neck of the process is the examination part. Detecting the tiny fractures of the vessels, micro-aneurysms and blood clots is a non-trivial task even for a licensed physician. This can limit access to medical care in case of scarcity of specialists.

Various approaches have been used to address this classification problem. Aside from the more classical approaches using feed-forward neural networks of limited depth, a vast part of the most successful results was achieved using deep convolutional neural networks (CNNs). In general, neural networks are a family of models, which approximate a given function using a combination of linear transformations, referred to as layers. A layer is usually represented by a weight matrix, whose dimensions define the number of input and output features. After each layer, a non-linear activation function is usually applied. During the training phase, input data is first propagated through the network to find out the network's response. Next, a loss function is applied to measure the difference between ground-truth and the predictions. After that, a gradient is calculated and weights of each layer are updated accordingly as the gradient is propagated backwards through the network to find a minimum of the loss function. It is usually achieved by a stochastic optimization algorithm, such as SGD with momentum [12]. State-of-the-art CNNs have shown their high ability to solve difficult image classification problems. Numerous architectures have been tested on such benchmarks as CIFAR-10 [7] or ImageNet [8]. Recent advancements have both made the process of training faster and allowed to achieve better accuracy.

The DR problem has been well-known to the community and approached several times. Yun introduced a well-performing method relying on heavily preprocessed input used to train a feed-forward neural network [20]. In 2015, Haloi

proposed a compact convolutional neural network model [3] achieving state-of-the-art accuracy in detecting micro-aneurysms using a dataset with pixel-level ground truths. In the same year a solution to the coexistent problem of blood vessel detection was proposed [11], using neural networks for digital fundus image segmentation. Recent efforts by Gulshan et al. [18] resulted in significant advancement compared to the previous works. Their solution is based on significant data augmentation and preprocessing of a large dataset of over 120000 fundus images. The ground-truth labels contained detailed information about image quality and were reviewed by multiple board-certified ophtamologists. An interesting approach with comparable results was proposed by Lim et al., who used image segmentation to assess the presence of DR in images [10].

Structure of the paper The structure of this work is organized as follows: in section 2 we describe the architecture of the network models used and the processing pipeline. In section 3 we present the experiment design and statistical results. Section 4 summarizes the results and presents the direction of future work.

The dataset We have used a dataset consisting of 35,127 fundus images acquired during independent series of clinical examinations using various devices. The images were labelled as one of the five classes, indexed from 0 to 4. Membership in the "0" class means that the image is free of any features characteristic to DR, "1" means subtle changes, "2" means moderate presence of features, "3" means high presence of features and "4" means severe, proliferative DR. The dataset distribution over classes is presented in Table 1.

Table 1. Distribution of dataset elements

Class index	Share in dataset
0	0.73
1	0.07
2	0.15
3	0.03
4	0.02

2 Classification model

In the approach described in this paper we use convolutional neural networks. Neural network is a model that applies a series of linear (and optionally non-linear) transformations on input data tensors and output predictions as values. Unlike other types of classifiers it does not depend on a fixed set of basis functions like, for instance, kernel SVM, but allows the basis functions to be altered by the training data in order to fit the modelled data more accurately. In addition,

hidden layers of convolutional neural networks apply the convolution operation using filters of size defined as a hyperparameter and the values modified by the fitting algorithm. The use of convolution makes the model shift invariant. This trait is valuable in general image processing as it decouples feature presence from its location and provides significantly better generalization capability. However, sometimes it is important to know the location of the feature, for example when there are multiple objects of interest in one image or its location has valuable information. Image segmentation is a well-known problem and has been approached by a number of studies using pixel-wise classification [8][17] or other techniques [15], most of which require label data that contain at least some segmentation information. In this approach, no prior localization information is required, yet the model is capable of predicting approximate location of the features of interest.

2.1 Network architecture

The network architecture is based on the residual networks [5], which allow to construct architectures consisting of a larger number of hidden layers, while avoiding the vanishing gradient problem. The entire model consists of a set of networks all constructed in a similar way, as shown in Figure 2. The network consists of five segments. Each segment computes a sum of two transformations of the input received from the previous layer. One is either identity or max pooling, depending on desired dimensionality of output. The other is built from three subsequent convolutional layers, each followed by a batch normalization layer. Various modifications are applied to the base network architecture. During the computation of the saliency map, a modified backpropagation algorithm is used [16]. In the final prediction network, a technique called Adaptive Spatial Pooling Layer [4] is used to ensure the consistency of the output tensor dimensionality with minimal computational cost. The layer automatically adjusts the stride and kernel size parameters in order to obtain desired output tensor dimensionality.

2.2 Feature extraction

First, the model is trained using images downsampled to $3 \times 1024 \times 1024$ pixels in order to distinguish between classes {0, 1} (referred to as "healthy") and {2, 3, 4} (referred to as "unhealthy"). Images are preprocessed by subtracting the average image calculated over the entire training set. In addition, each image is normalized on the fly by stretching its histogram, i.e. so that the brightest pixel has value 1 and the darkest pixel has value 0 (assuming that each pixel's brightness is represented by a decimal fraction between 0 and 1). The new approach to classification is based on applying the saliency map to reinforce the information about location of the features and use it to create a classification model that makes predictions based on the parts of the input image containing the most information. This first training step enabled the network to learn strong, prevalent features that could be easily distinguished from the surroundings. The key idea is to use the information stored in the network's weights to trace parts of the

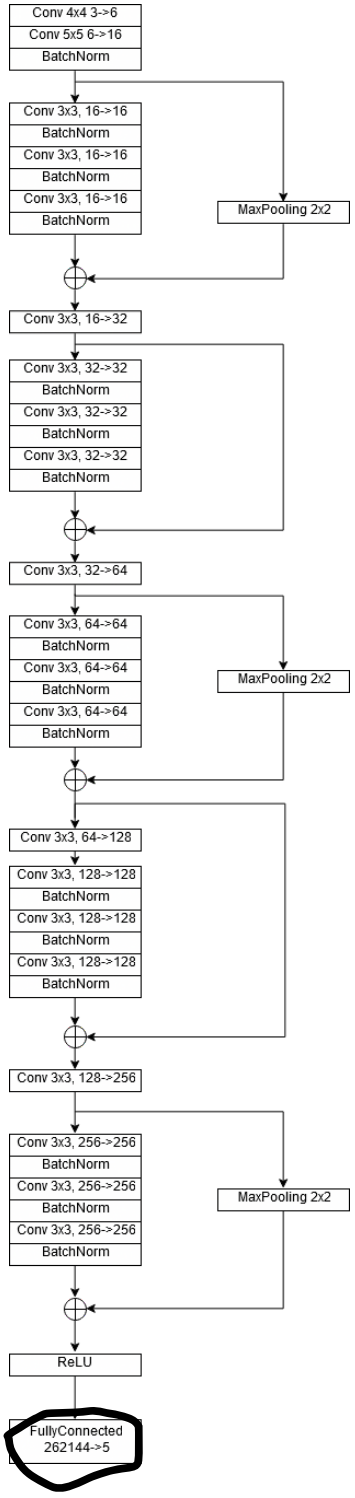

Fig. 2. Base neural network architecture diagram.

input image that caused the increase in probabilities of the "unhealthy" class. To achieve this, the trained model is modified by changing all layers in such a manner that they back-propagate the gradient only for the traced class while the rest is set to 0. The gradient is then propagated from the output layer down to the input layer. After calculating the gradient with respect to the input, an average over all three channels is calculated, resulting in a grayscale image. The intensity of each pixel describes how much it contributes to increase the likelihood of the "unhealthy" class. The obtained result is called a saliency map [16]. Its visualization is presented in Figure 3. After obtaining the saliency map, we

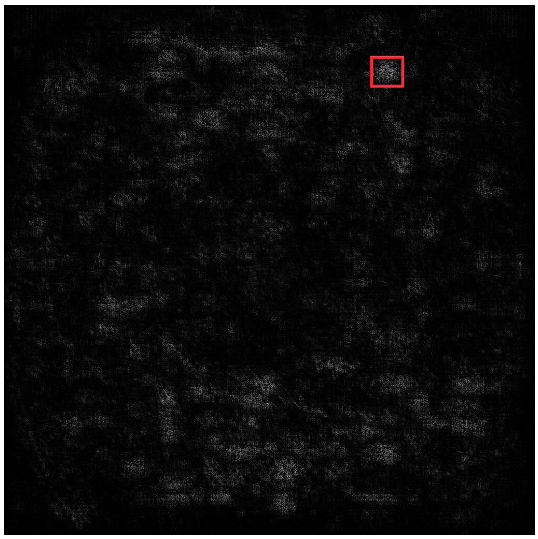

Fig. 3. Example of a saliency map for a member of class 4. The red rectangle denotes the area selected as the centre of the most suspicious area of the fundus image.

propose to perform the max operation on it in order to select the most significant area of the input image. The straightforward approach here is to select the point with the highest value, however one can consider applying a clustering method that will preserve information about feature significance density in order to lower the significance of a random noise. The coordinates of each point for which the maximum influence is indicated are subsequently used to generate 1581 $3 \times 512 \times 512$ pixel-sized crops centred around the corresponding pixels in the full resolution images from members of classes "4" and "5". An example is shown in Figure 4. If the location of the centre causes the crop boundaries to overflow the image, the centre is shifted accordingly. This allows to gather more information about the suspicious area, as opposed to simply padding the overflow. Crops of images representing the "0" class are acquired by randomly selecting the upper-left corner coordinates of the $3 \times 512 \times 512$ window, thus eliminating all corner cases. The crops acquired as a result of this process are

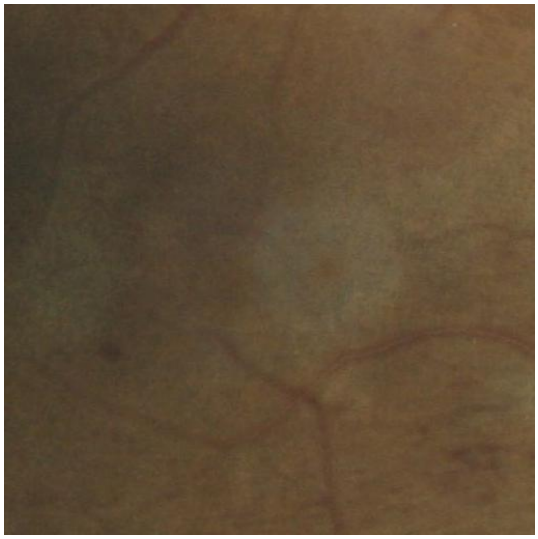

Fig. 4. Example of a most suspicious area for a member of class 4. Arrows denote features of interest including microaneurysm and signs of subretinal haemorrhage.

then labeled as follows: those selected from images representing classes "0" or "1" are given the label "healthy", while the remaining are given the label "unhealthy". Next, the same network architecture as shown in Figure 2 is used to train the prediction model that distinguishes between those two classes. Next, as a part of our contribution, we create a corresponding network by copying all weights from the previous one, but replacing the fully-connected layer to a 1×1 convolution. This technical trick enables the network to avoid size-mismatch errors when dealing with input images of variable sizes. Every training image is assessed by the network. We introduce a new parameter: κ, determining the percentage of pixels from each edge that are not to be taken into account. Its purpose is to minimize the "border effect", which would cause the network to generate unrealistically high probabilities in locations with very high variance. Most of such regions occurred around the circular borders of the fundus. Hence, a new crop centre for each image described by a function $I_n(x, y)$ of size $w \times h$ was determined by the rule shown in Formula 1.

$$(c_x, c_y) = \arg \max_{\substack{\kappa<x<w-\kappa \\ \kappa<y<y-\kappa}} I_n(x, y) \tag{1}$$

Next, every centre is used to generate crops in the same manner as in the previous step, which are then used to create one of the two final prediction models. Two different approaches are used to make predictions. The first one is to train the final convolutional network with crops acquired in the second iteration and apply it to the entire full-size image, resulting in a tensor representing the probabilities of each class for overlapping areas of the image, as if a sliding window was applied.

However, this proved to be ineffective. The majority of image fragments were assigned such probabilities of being a member of the "unhealthy" class that the two classes would not be separable. The process is presented in detail in Listing 1.

Listing 1. Pseudocode for the first approach utilizing the sliding window.

```
1  net:=train_network(training_set)
2  crops_step1 = []
3  crops_step2 = []
4  for image in downsampled_training_set:
5          response:=net.forward_pass(image)
6          map:=saliency_map(response)
7          map:=apply_kappa_rule(map)
8          (cx,cy):=argmax(map)
9          crop:=create_crop(full_image,cx,cy)
10         crops_step1.addNew(crop)
11 net_crops_1:=train_network(crops_step1)
12 for image in fullsize_training_set:
13         response:=net_crops_1.forward_pass(image)
14         map:=saliency_map(response)
15         map:=apply_kappa_rule(map)
16         (cx,cy):=argmax(map)
17         full_image:=get_matching_full_image(image)
18         crop:=create_crop(full_image,cx,cy)
19         crops_step2.addNew(crop)
20 net_crops_2:=train_network(crops_step1)
21 for image in fullsize_validation_set:
22         response:=net_crops_2.forward_pass(image)
23         probabilities:=histogram(response)
24         return prediction(histogram)
```

To tackle the issues mentioned earlier, in our contribution we decided to apply the following steps:

1. Train the CNN constructed as shown in Figure 2 with downsampled images.
2. Forward feed he full size image to the network previously trained on downsampled images, modified by applying an adaptive spatial pooling layer before the fully connected layer. It allows the network to output a single vector of probabilities for the entire image rather than a set of overlapping segments.
3. Obtain the saliency map of the full size image
4. Apply the kappa border rule to the saliency map.
5. Generate the most suspicious crop, as described earlier in this section.
6. Classify the crop using the network trained with the crops in line 10 of Listing 2.

The final decision thresholds were set with the aid of a two-class support vector machine [2]. The entire process is presented in detail in Listing 2.

Listing 2. Pseudocode for the second approach relying on the saliency map.

```
net:=train_network(training_set)
crops = []
for image in downsampled_training_set:
        response:=net.forward_pass(image)
        map:=saliency_map(response)
        map:=apply_kappa_rule(map)
        (cx,cy):=argmax(map)
        crop:=create_crop(full_image,cx,cy)
        crops.addNew(crop)
net_crops:=train_network(crops)
for image in fullsize_validation_set:
        response:=net_crops.forward_pass(image)
        map:=saliency_map(response)
        map:=apply_kappa_rule(map)
        (cx,cy):=argmax(map)
        crop:=create_crop(full_image,cx,cy)
        response:=net_crops.forward_pass(crop)
        probabilities:=histogram(response)
        return prediction(histogram)
```

3 Experiment design

The model was evaluated on a testing set of 1349 images randomly sampled out of the provided data set. Those images were chosen prior to the training procedure and were not included in the training set. All experiments were run on a single graphics processing unit Nvidia Tesla K40 with 12GB of memory. The data was stored on a solid state drive disk. A number of metrics were analysed in order to assess each model's fitness. For each of them a confusion matrix was computed beforehand. The first approach is based on calculating the percentage of correctly predicted classes. However, in multi-class cases this is not enough to determine which of two given models was giving more accurate responses, since it does not distinguish between two cases, with a similar number of false-positive and false-negative errors respectively, of which the latter is much more unwanted. Additional interpretation of the confusion matrix was necessary here. Also, since the strong similarities between adjacent classes diminish the significance of confusion between them, another score was introduced. It describes the percentage of test images to the correct class or to a class different by 1 grade. The third measure, used mostly for training purpose yet still yielding valuable information, was Cohen's quadratic weighed kappa [1].

Due to the need of clinical assessment and treatment of each patient with DR detected it is a legitimate choice to limit the prediction to just two classes: one representing stages 0 and 1, labelled as "healthy", and the other representing stages 2, 3 and 4, labelled as "unhealthy". This also simplified the analysis process of the results and comparison of different models. Each image was assigned

by the network to one of those two classes and in the end, a final accuracy percentage score was shown for each class.

4 Results and Discussion

In this section we present the three models which were subject to thorough evaluation. Along with the predictions of a simple ResNet-based model, we tested the sliding window model and the saliency map based model. The last one proved to be much more accurate in detecting the "unhealthy" class than the naive residual network approach, as presented in Table 2.

Table 2. Comparison of prediction accuracy of the three models.

	ResNet	Sliding Window	Saliency Map
"healthy" class	91.1%	58.5%	69.5%
"unhealthy" class	41.2%	56.6%	82.8%

Table 3. Comparison of AUC scores of the three models.

	ResNet	Sliding Window	Saliency Map
Area under ROC curve	0.71	0.56	0.77

The saliency map approach allowed to reach over 82% accuracy and nearly 70% specificity. It is worth noting that due to hardware restrictions, the images had to be scaled down significantly in the first step of the training process, which supposedly had a non-negligible impact on the final results. Despite that, a clear improvement is visible over the naive approach, which is reflected in both increase in the AUC value shown in Table 3 and a huge reduction of the false-negative rate at the cost of a small increase in false-positive rate. What is more important, after manual examination of randomly selected crops created during the training process, a vast majority of them was centred around some sort of anomaly of the retina. However, due to lack of proper labeling of the dataset, it was impossible to measure this objectively. Exact per-class prediction results are presented below in Table 4. The most important observation is that no image belonging to class 4 was classified as "healthy" and most errors were made by misclassifying members of class 2 as "healthy". This along with the AUC value shows that the model is capable of detecting strong, localized features and propose a classification border. In real-world applications, these questionable results could be marked as needing human assessment. It should be noted that in this chapter there is no direct comparison made to other works on the same dataset. It would be difficult

Table 4. Confusion matrix of the saliency map model.

	"healthy"		"unhealthy"		
	Class 0	Class 1	Class 2	Class 3	Class 4
"healthy"	542	201	319	7	0
"unhealthy"	17	31	64	91	77

to do so because of huge differences in the approach and desired effect of our research. The main aim of our work was to find a way to locate the region with a high feature density rather than to improve the general classification score, yet it does not diminish its usefulness.

5 Conclusions

We have shown that it is possible to use neural networks with saliency maps to improve classification of data with highly localized features without any prior knowledge about their localization. Results have shown a significant improvement in comparison to a standard approach. It should be pointed out that state-of-the-art classification results were achieved using well annotated datasets with feature localization information, unlike the approach proposed in this paper. Because of this difference, those results cannot be compared directly. There is space for improvement in this method and further research is necessary to explore additional possibilities. It is especially worth looking into how the models behave when trained on full-resolution input images in the first step, however, it will require significantly more computational power.

6 Acknowledgements

I would like to express sincere gratitude to my advisor, Marcin Kurdziel, PhD for his expert advice, as well as Wojciech Czech, PhD for his valuable critique.

References

1. Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.
2. Corinna Cortes and Vladimir Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.
3. Mrinal Haloi. Improved microaneurysm detection using deep neural networks. *arXiv preprint arXiv:1505.04424*, 2015.
4. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014.

5. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
6. Nikola K Kasabov. Neucube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. *Neural Networks*, 52:62–76, 2014.
7. Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.
8. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
9. Jinsa Kuruvilla and K Gunavathi. Lung cancer classification using neural networks for ct images. *Computer methods and programs in biomedicine*, 113(1):202–209, 2014.
10. Gilbert Lim, Mong-Li Lee, Wynne Hsu, and Tien Yin Wong. Transformed representations for convolutional neural networks in diabetic retinopathy screening. In *AAAI Workshop: Modern Artificial Intelligence for Health Analytics*, 2014.
11. Martina Melinščak, Pavle Prentašić, and Sven Lončarić. Retinal vessel segmentation using deep neural networks. In *VISAPP 2015 (10th International Conference on Computer Vision Theory and Applications)*, 2015.
12. Yurii Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.
13. Mohd Fauzi Othman and Mohd Ariffanan Mohd Basri. Probabilistic neural network for brain tumor classification. In *Intelligent Systems, Modelling and Simulation (ISMS), 2011 Second International Conference on*, pages 136–138. IEEE, 2011.
14. Robert W Rodieck. The vertebrate retina: principles of structure and function. 1973.
15. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*, pages 234–241. Springer International Publishing, Cham, 2015.
16. Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.
17. Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
18. Gulshan V, Peng L, Coram M, and et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316(22):2402–2410, 2016.
19. CP Wilkinson, Frederick L Ferris, Ronald E Klein, Paul P Lee, Carl David Agardh, Matthew Davis, Diana Dills, Anselm Kampik, R Pararajasegaram, Juan T Verdaguer, et al. Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales. *Ophthalmology*, 110(9):1677–1682, 2003.
20. Wong Li Yun, U. Rajendra Acharya, Y.V. Venkatesh, Caroline Chee, Lim Choo Min, and E.Y.K. Ng. Identification of different stages of diabetic retinopathy using retinal optical images. *Information Sciences*, 178(1):106 – 121, 2008.