Rafał Grzeszczuk

APPROACH TO CLASSIFYING DATA WITH HIGHLY LOCALIZED UNMARKED FEATURES USING NEURAL NETWORKS

Abstract *To face the increasing demand of quality healthcare, cutting-edge automation technology is being applied in demanding areas such 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 a saliency map in the amplification of features. Unlike previous efforts, this approach does not use any prior information about feature localization. We present an experimental study based on the Diabetic Retinopathy classification problem, in which our method has shown to achieve an 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* 35,120 *images of various qualities, inconsistent resolutions, and aspect ratios.*

Keywords classification, neural networks, medical image analysis

Citation Computer Science 20(3) 2019: 329–342

1. Introduction

Medical image analysis is a complex field of applied computer vision. Due to the ramifications of an 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 the early stages of a disease (when treatment is easier).

It is often the case 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, usually requiring a trained human expert such as an experienced physician to make the assessments. Typical problems that fall into this category include (but are not limited to) tumor detection in X-ray and computer tomography scans [13, 16] or (micro-)stroke detection in magnetic resonance imaging scans [10]. In this paper, we focus on diabetic retinopathy, which is often abbreviated "DR" [21] (cf. Fig. 1).

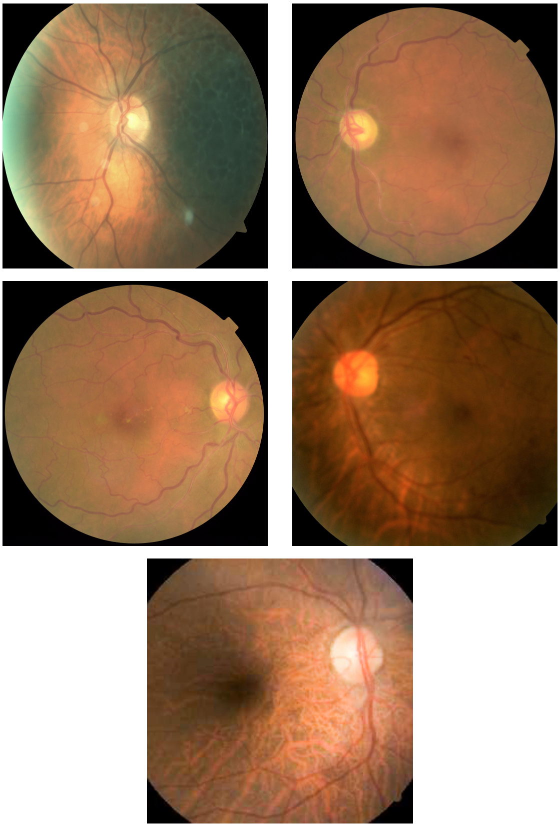

Figure 1. Example fundi images from each class ordered by increasing amounts of retinal deterioration. Image in top-left corner shows healthy retina (Class 0). Bottom image is example of proliferative retinopathy (Class 4)

The retina is a tissue that is responsible for converting visual stimuli into chemical signals that are subsequently transmitted to and interpreted by the visual cortex, which is located in the occipital lobe of the brain [17]. In patients with Type I diabetes, abnormally high concentrations of glucose are found in the blood, resulting in persistent damage to their internal organs. As the diabetes progresses, high blood sugar levels cause the blood vessels to become damaged and perforated, resulting in small hemorrhages that can damage the retina.

This can be detected by a non-invasive medical procedure called ophthalmoscopy; during this procedure, the retina is examined (or photographed) through the pupil with an optical device. Unfortunately, the examination part is a significant bottleneck of the process. Detecting tiny fractures in vessels, micro-aneurysms, and blood clots is a task that is challenging even for a licensed physician. In the case of a scarcity of specialists, this can limit access to medical care. Annotated image analysis for ophthalmology would alleviate this concern.

Most state-of-the-art approaches to medical image analysis rely on the application of neural networks. In general, neural networks are a family of models that approximate some decision function using a combination of linear transformations, which are 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, the input data is first propagated through the network to find out the network's response.

Next, a loss function is used to measure the difference between the ground truth and the predictions. After that, a gradient is calculated and the weights of each layer updated, as the gradient is propagated backwards through the network to find a minimum of the loss function. This is usually achieved by a stochastic optimization algorithm such as SGD with momentum [1].

Related work

Various approaches have been used to address the image classification problem. Aside from the more classical approaches using feed-forward neural networks of limited depths, a vast part of the most successful results was achieved using deep convolutional neural networks (CNNs). Numerous architectures have been tested on benchmarks such as CIFAR-10 [11] or ImageNet [12]. Recent advancements have also made the process of training faster, allowing us to achieve better accuracy [4]. The DR image classification problem has been well-known to the community and approached several times.

Yun introduced a well-performing method that relies on heavily pre-processed input used to train a feed-forward neural network [22]. In 2015, Haloi proposed a compact convolutional neural network model [7], achieving state-of-the-art accuracy in detecting micro-aneurysms using a dataset with pixel-level ground truths. During the same year, a solution to the coexistent problem of blood vessel detection was proposed by [15] using neural networks for digital fundus image segmentation.

Recent efforts by Gulshan *et al.* [6] have resulted in significant progress compared to the previous works. Their solution is based on significant data augmentation and the preprocessing of a large dataset consisting of more than 120,000 fundus images. The ground truth labels in this model contained detailed information about image quality and were reviewed by multiple board-certified ophthalmologists. An interesting approach with comparable results was proposed by Lim *et al.*, who used image segmentation to assess the presence of DR in images [14]. All of these contributions assumed prior knowledge about exact feature localization and often an augmented set of ground truths. This is in contrast to our work, which attempts to learn the features and their localizations merely from the class membership information.

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 experimental results. Section 4 summarizes the results and presents the direction of future work.

2. Classification model

In the approach described in this paper, we use convolutional neural networks. A neural network is a machine-learning model that applies a series of linear and non-linear transformations on input data in order to make predictions. Unlike other types of classifiers (e.g., Kernel SVM), it does not depend on a fixed set of basis functions; it allows the basis functions to be altered during the training to fit the modeled data more accurately. In addition, hidden layers of convolutional neural networks apply the convolution operation using filters learned during the training.

The use of convolution makes the model shift invariant; this trait is valuable in general image processing, as it decouples the feature presence from its location and provides significantly better generalization capabilities. 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 that has been successfully approached by a number of studies using pixel-wise classification [12, 20] or other techniques [18], most of which require label data that contains at least some segmentation information. In this work, no prior localization information is required, yet the model is capable of predicting the approximate location of the features of interest.

2.1. Network architecture

The architecture of our model is based on residual networks [9], which enables the construction of networks with a larger number of hidden layers while avoiding the vanishing gradient problem. The entire model employs a set of networks that are all constructed in a similar way (as shown in Figure 2).

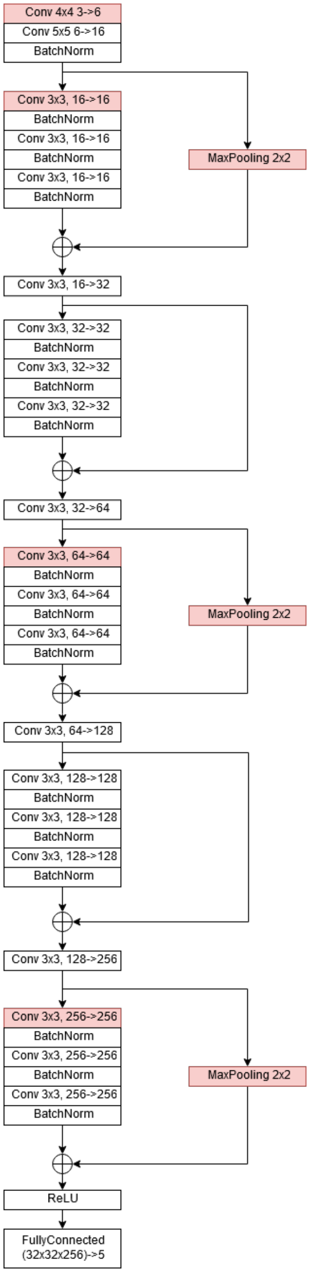

Figure 2. Diagram of base neural network architecture. There are two types of blocks differing by optional dimensionality reduction. If dimensionality is reduced, then skip connection applies max-pooling to match dimensionality. Otherwise, skip-connection is simply identity, as described in [9]. Dimensionality-reducing layers are marked with red background. All layers include zero-padding in order to keep dimensions equal to powers of two. Each convolutional layer is annotated with its window size and input and output dimensionalities

The network consists of an input segment, five residual segments composed sequentially, and a fully connected layer for classification. Each residual segment accepts the output of the preceding one. The output of such segment is the sum of two transformations: one is either identity or max pooling, depending on the desired dimensionality of the 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 back-propagation algorithm is used [19]. In the final prediction network, adaptive spatial pooling layer [8] 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 the desired output tensor dimensionality.

2.2. Feature extraction

First, we train our model by using images downsampled to $3 \times 1024 \times 1024$ pixels in order to distinguish among classes $\{0, 1\}$ (referred to as "healthy") and $\{2, 3, 4\}$ (referred to as "unhealthy"). The 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 so that the brightest pixel has a value of 1 and the darkest pixel has a value of 0.

We apply the method above in a new approach to classification that is based on applying saliency maps to reinforce the information about the locations of important features. This allows us to create a classification model that makes predictions based on the parts of the input image that contains the most important information. This first training step enabled the network to learn strong and prevalent features that could be easily distinguished from the surroundings. In particular, the key idea is to use the information stored in the network's weights to trace the parts of the input image that caused the increase in the probabilities of the positive class. To achieve this, the trained model is modified by changing all of the layers in such a manner that they back-propagate the gradient only for the positive class while the gradient corresponding to the negative 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 increases the probability of image being classified as "unhealthy". The result is called a saliency map [19]. See Figure 3 for an example visualization. After obtaining the saliency map, we perform the max operation on it in order to select the most important area of the input image. The straightforward approach here is to select the point with the highest value. However, one can also consider applying a clustering method that will preserve information about the detections of density in order to lower the significance of random noise. The coordinates of the points with the greatest influence on the positive class are subsequently used to generate 1581 $3 \times 512 \times 512$ pixel-sized crops centered around the corresponding pixels in the full-resolution images from members of the “4” and “5” classes (see Figure 4 for an example).

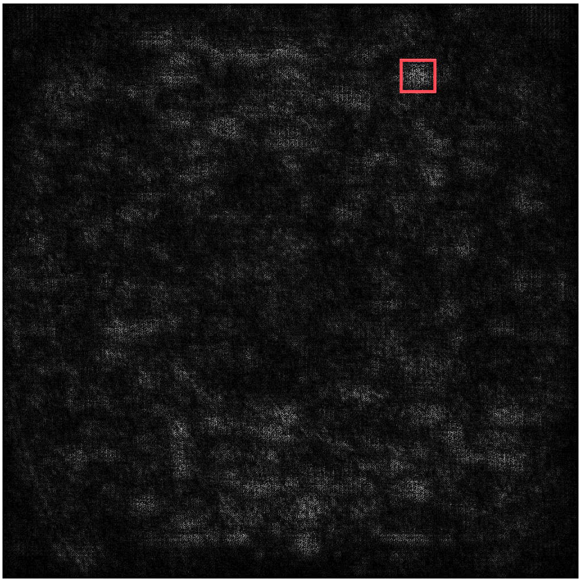

Figure 3. Example of a saliency map for a member of Class 4. The red rectangle denotes the area selected as the center of the most suspicious area of the fundus image

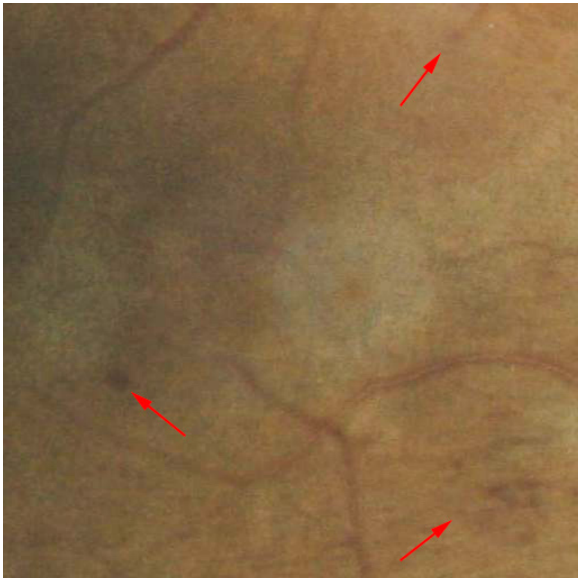

Figure 4. Example of most suspicious area for member of Class 4. Arrows denote features of interest including micro-aneurysm and signs of sub-retinal hemorrhage

If the location of the center causes the crop boundaries to overflow the image, the center is shifted accordingly. This allows us to gather more information about the suspicious area (as opposed to simply padding the overflow). Crops of the images representing the "0" class are acquired by randomly selecting the $3 \times 512 \times 512$ crop windows.

The crops acquired as a result of this process are then labeled as follows: those selected from the images representing the "0" or "1" classes are given a "healthy" label, while the others are labeled "unhealthy." Next, the same network architecture shown in Figure 2 is used to train the prediction model that distinguishes between these two classes. Finally, we create a corresponding network by copying all of the weights from the previous one but replacing the fully-connected layer with a 1×1 convolution. This technical trick enables the network to avoid size-mismatch errors when dealing with input images of variable sizes. Each training image is assessed by the network. We also introduce a new parameter (κ), which determines the percentage of pixels from each image that are not taken into account. Its purpose is to minimize the "border effect", which causes the network to generate unrealistically high probabilities in locations with very high variances. Most of these regions occur around the circular borders of the fundus. After the network processing, a new crop center is selected for each image by the rule shown in Formula 1.

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

where $I_n(x, y)$ returns the intensity of the saliency map for the (x, y) pixel in the original input image.

Next, each center is used to generate crops in the same manner as in the previous step; these are then used to create one of the two final prediction models. We experiment with two approaches to make our predictions. The first one is to train the final CNN with the 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 convolutional window was applied). However, this proved to be ineffective. The majority of the image fragments were assigned such probabilities of being members 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 first approach to final image classification

```
crops_step1 = [], crops_step2 = []
net:=train_network(training_set)
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_step1.addNew(crop)
net_crops_1:=train_network(crops_step1)
for image in fullsize_training_set:
        response:=net_crops_1.forward_pass(image)
        map:=saliency_map(response)
        map:=apply_kappa_rule(map)
        (cx,cy):=argmax(map)
        full_image:=get_matching_full_image(image)
        crop:=create_crop(full_image,cx,cy)
        crops_step2.addNew(crop)
net_crops_2:=train_network(crops_step2)
for image in fullsize_validation_set:
        response:=net_crops_2.forward_pass(image)
        probabilities:=histogram(response)
        return prediction(histogram)
```

To tackle the issues mentioned above, we decided to apply the following steps:

1. Train the CNN constructed as shown in Figure 2 with downsampled images.
2. Process the full-size image with the network previously trained on downsampled images modified by applying an adaptive spatial pooling layer before the fully connected layer. This allows the network to output a single vector of probabilities for the entire image rather than a set of overlapping segments.
3. Obtain a 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 training crops (Line 10 of Listing 2).

The final decision thresholds were set with the aid of a two-class support vector machine [3]. The entire process is presented in detail in Listing 2. Processing the entire pipeline lasts for approximately 70 hours.

Listing 2. Pseudocode for second approach relying on saliency map

```
 1  crops = []
 2  net:=train_network(training_set)
 3  for image in downsampled_training_set:
 4          response:=net.forward_pass(image)
 5          map:=saliency_map(response)
 6          map:=apply_kappa_rule(map)
 7          (cx,cy):=argmax(map)
 8          crop:=create_crop(full_image,cx,cy)
 9          crops.addNew(crop)
10  net_crops:=train_network(crops)
11  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

We evaluated our model on a test set of 1349 images randomly sampled from the provided data set. These images were chosen prior to the training procedure and were not included in the training set. All experiments were run on a single Nvidia Tesla K40 graphics processing unit with 12GB of memory. The data was stored on a solid-state disk drive in a compressed image format. A number of metrics were analyzed in order to assess each model's fitness. We also computed a confusion matrix for each evaluated model. The evaluation criteria are based on calculating the percentage of correctly predicted classes. However, this is not enough to determine which of two given models was giving more accurate responses in multi-class cases, as 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 is necessary here. Since the strong similarities between adjacent classes diminish the significance of the confusion between them, we also used supplementary metrics. This counts the percentage of test images assigned to the correct classes or to a class different by one grade. The third measure (used mostly for training purpose yet still yielding valuable information) was Cohen's quadratic weighed kappa [2].

Due to the need for the clinical assessment and treatment of each patient with a detected DR, it is a legitimate choice to limit the prediction to just two classes: one representing Stages 0 and 1 (labeled "healthy") and another representing Stages 2, 3, and 4 (labeled "unhealthy"). This also simplified the analysis of the results and comparison of the different models. Each image was assigned to one of these two classes by the network; in the end, a final accuracy percentage score was calculated for each class.

Dataset

We used a dataset consisting of 35,127 fundus images randomly selected from a publicly available dataset supplied by EyePACS for a machine-learning contest [5]. The samples were acquired during independent series of clinical examinations using Centervue DRS, Topcon NW, Optovue iCam, and Canon CR/DGi retina-imaging cameras (all of which provide a 45° field of view). The examinations were carried out between May 2015 and October 2015 on outpatients of various ages and sexes present for DR screening (some of which being examined more than once within the collection period). Some of the fundus images were taken after applying a pupil-dilation substance. Despite the fact that it could interfere with the image brightness, this fact was not reflected in the data labels. The images were labeled 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 that can indicate the earliest stage of the disease or other conditions and impose the need for further diagnostics, "2" means a moderate presence of features, "3" means an abundant presence of features, and "4" means severe proliferative DR. The dataset distribution over the classes is presented in Table 1.

Table 1

Distribution of dataset elements

Class index	Share in dataset [%]
0	73
1	7
2	15
3	3
4	2

4. Results and discussion

In this section, we present the three models that we thoroughly evaluated. 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 demonstrated by the results in Table 2).

Table 2

Comparison of per-class prediction accuracy of three models

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

The saliency map approach allowed us to reach a greater than 82% accuracy and a 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 arguably had a non-negligible impact on the final results. Despite this, a clear improvement is visible over the naive approach, which is reflected in both an 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 the false-positive rate.

Table 3
Comparison of AUC scores of three models

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

What is more important, after a manual examination of randomly selected crops created during the training process, a vast majority of them turned out to be centered around some sort of anomaly of the retina. However, due to the lack of proper labeling of the dataset, it was impossible to measure this objectively. The exact per-class prediction results are presented below in Table 4.

Table 4
Confusion matrix of 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

The most important observation is that no image belonging to Class 4 was classified as "healthy"; the 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 strongly localized features and proposing a classification border. However, predictions for some classes would need to be marked as needing human assessment in real-world applications. Finally, it should be noted that there is no direct comparison made to other works on the same dataset in this chapter. It would be difficult to do so because of the huge differences in the approach and desired outcomes 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.

5. Conclusions

We have shown that it is possible to use neural networks with saliency maps to improve the classification of data with highly localized features without any prior knowledge about their localization. The results showed a significant improvement as compared 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, these results cannot be directly compared. There is space for improvement in this method; further research is necessary to explore additional possibilities. It is especially interesting to explore how the models behave when trained on full-resolution input images. However, this will require significantly more computational power.

References

[1] Bottou L.: Stochastic Learning. In: Bousquet O., von Luxburg U., Rätsch G. (eds.), *Advanced Lectures on Machine Learning. ML 2003. Lecture Notes in Computer Science*, vol. 3176, Springer, Berlin–Heidelberg, pp. 146–168, 2004.

[2] Cohen J.: A Coefficient of Agreement for Nominal Scales, *Educational and Psychological Measurement*, vol. 20(1), pp. 37–46, 1960.

[3] Cortes C., Vapnik V.: Support Vector Networks, *Machine Learning*, vol. 20(3), pp. 273–297, 1995.

[4] Devi B.A., Rajasekaran M.P.: Performance Evaluation of MRI Pancreas Image Classification Using Artificial Neural Network (ANN). In: Satapathy S., Bhateja V., Das S. (eds.), *Smart Intelligent Computing and Applications. Smart Innovation, Systems and Technologies*, vol. 104 , Springer, pp. 671–681, 2019.

[5] EyePACS: Public diabetic retinopathy dataset, 2015 `www.kaggle.com/c/diabetic-retinopathy-detection`.

[6] Gulshan V., Peng L., Coram M., et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs, *JAMA*, vol. 316(22), pp. 2402–2410, 2016. `https://doi.org/10.1001/jama.2016.17216`.

[7] Haloi M.: Improved Microaneurysm Detection Using Deep Neural Networks, *arXiv preprint arXiv:1505.04424*, 2015.

[8] He K., Zhang X., Ren S., Sun J.: Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds.), *Computer Vision – ECCV 2014*. Lecture Notes in Computer Science, vol. 8691, Springer, Cham, pp. 346–361, 2014.

[9] He K., Zhang X., Ren S., Sun J.: Deep Residual Learning for Image Recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[10] Kasabov N.K.: NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data, *Neural Networks*, vol. 52, pp. 62–76, 2014.

[11] Krizhevsky A., Nair V., Hinton G.: The CIFAR-10 dataset, 2014. `https://www.cs.toronto.edu/~kriz/cifar.html`.

[12] Krizhevsky A., Sutskever I., Hinton G.E.: ImageNet Classification with Deep Convolutional Neural Networks. In: *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[13] Kuruvilla J., Gunavathi K.: Lung cancer classification using neural networks for CT images, *Computer Methods and Programs in Biomedicine*, vol. 113(1), pp. 202–209, 2014.

[14] Lim G., Lee M.L., Hsu W., Wong T.Y.: Transformed Representations for Convolutional Neural Networks in Diabetic Retinopathy Screening. In: *AAAI Workshop: Modern Artificial Intelligence for Health Analytics*, 2014.

[15] Melinščak M., Prentašić P., Lončarić S.: Retinal Vessel Segmentation Using Deep Neural Networks. In: *VISAPP 2015 (10th International Conference on Computer Vision Theory and Applications)*, 2015.

[16] Othman M.F., Basri M.A.M.: Probabilistic Neural Network for Brain Tumor Classification. In: *2011 Second International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, pp. 136–138, IEEE, 2011.

[17] Rodieck R.W.: The Vertebrate Retina: Principles of Structure and Function, Freeman, 1973.

[18] Ronneberger O., Fischer P., Brox T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, vol. 9351, Springer, Cham, pp. 234–241, 2015. `https://doi.org/10.1007/978-3-319-24574-4_28`.

[19] Simonyan K., Vedaldi A., Zisserman A.: Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, *arXiv preprint arXiv:1312.6034*, 2013.

[20] Tompson J., Goroshin R., Jain A., LeCun Y., Bregler C.: Efficient Object Localization Using Convolutional Networks. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[21] Wilkinson C., Ferris F.L., Klein R.E., Lee P.P., Agardh C.D., Davis M., Dills D., Kampik A., Pararajasegaram R., Verdaguer J.T.: Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales, *Ophthalmology*, vol. 110(9), pp. 1677–1682, 2003.

[22] Yun W.L., Acharya U.R., Venkatesh Y.V., Chee C., Min L.C., Ng E.: Identification of different stages of diabetic retinopathy using retinal optical images, *Information Sciences*, vol. 178(1), pp. 106–121, 2008. `https://doi.org/10.1016/j.ins.2007.07.020`.

Affiliations

Rafał Grzeszczuk

AGH University of Science and Technology, Department of Computer Science, Krakow, Poland, ORCID ID: https://orcid.org/0000-0002-0736-9500

Received: 27.06.2019
Revised: 14.08.2019
Accepted: 14.08.2019