Deep Image Mining for Diabetic Retinopathy Screening

Gwenolé Quellec[a,*], Katia Charrière[b,a], Yassine Boudi[b,a], Béatrice Cochener[c,a,d], Mathieu Lamard[c,a]

[a]*Inserm, UMR 1101, Brest, F-29200 France*
[b]*Institut Mines-Telecom; Telecom Bretagne, Dpt ITI, Brest, F-29200 France*
[c]*Univ Bretagne Occidentale, Brest, F-29200 France*
[d]*Service d'Ophtalmologie, CHRU Brest, Brest, F-29200 France*

Abstract

Deep learning is quickly becoming the leading methodology for medical image analysis. Given a large medical archive, where each image is associated with a diagnosis, efficient pathology detectors or classifiers can be trained with virtually no expert knowledge about the target pathologies. However, deep learning algorithms, including the popular ConvNets, are black boxes: little is known about the local patterns analyzed by ConvNets to make a decision at the image level. A solution is proposed in this paper to create heatmaps showing which pixels in images play a role in the image-level predictions. In other words, a ConvNet trained for image-level classification can be used to detect lesions as well. A generalization of the backpropagation method is proposed in order to train ConvNets that produce high-quality heatmaps. The proposed solution is applied to diabetic retinopathy (DR) screening in a dataset of almost 90,000 fundus photographs from the 2015 Kaggle Diabetic Retinopathy competition and a private dataset of almost 110,000 photographs (e-ophtha). For the task of detecting referable DR, very good detection performance was achieved: A_z = 0.954 in Kaggle's dataset and A_z = 0.949 in e-ophtha. Performance was also evaluated at the image level and at the lesion level in the DiaretDB1 dataset, where four types of lesions are manually segmented: microaneurysms, hemorrhages, exudates and cotton-wool spots. For the task of detecting images containing these four lesion types, the proposed detector, which was trained to detect referable DR, outperforms recent algorithms trained to detect those lesions specifically, with pixel-level supervision. At the lesion level, the proposed detector outperforms heatmap generation algorithms for ConvNets. This detector is part of the Messidor® system for mobile eye pathology screening. Because it does not rely on expert knowledge or manual segmentation for detecting relevant patterns, the proposed solution is a promising image mining tool, which has the potential to discover new biomarkers in images.

Keywords: deep learning, image mining, diabetic retinopathy screening, lesion detection

1. Introduction

Retinal pathologies are responsible for millions of blindness cases worldwide. The leading causes of blindness are glaucoma (4.5 million cases), age-related macular degeneration (3.5 million cases) and diabetic retinopathy (2 million cases).[1] Early diagnosis is the key to slowing down the progression of these diseases and therefore preventing the occurrence of blindness. In the case of diabetic retinopathy (DR) screening, diabetic patients have their retinas examined regularly: a trained reader searches for the early signs of the pathology in fundus photographs (see Fig. 1) and decides whether the patient should be referred to an ophthalmologist for treatment. In order to reduce the workload of human interpretation, and therefore streamline retinal pathology screening, various image analysis algorithms have been developed over the last few decades. The first solutions were trained to detect lesions (at the pixel level) using manual segmentations (at the pixel level) for supervision

(Winder et al., 2009; Abràmoff et al., 2010): this is what we call computer-aided detection (CADe) algorithms. Based on the detected lesions, other algorithms were trained to detect pathologies (at the image level) (Abràmoff et al., 2010): this is what we call computer-aided diagnosis (CADx) algorithms. In recent years, new algorithms were designed to detect pathologies directly, using diagnoses (at the image level) only for supervision: these algorithms are based on multiple-instance learning (Quellec et al., 2017; Manivannan et al., 2017) or deep learning (LeCun et al., 2015). Because manual segmentations are not needed, such algorithms can be trained with much larger datasets, such as anonymized archives of examination records. The next challenge is to detect lesions using diagnoses only for supervision. Besides access to large training datasets, such an approach would allow discovery of new biomarkers in images, since algorithms are not limited by the subjectivity of manual segmentations. A few multiple-instance learning algorithms, supervised at the image level, can already detect lesions (Melendez et al., 2015; Quellec et al., 2016b). However, to our knowledge, no deep learning algorithm was designed to solve this task. The primary objective of this study is to find a way to detect lesions, or other biomarkers of DR, using deep learning

---

*LaTIM - Bâtiment 1 - CHRU Morvan - 2, Av. Foch
29609 Brest CEDEX - FRANCE
Tel.: +33 2 98 01 81 29 / Fax: +33 2 98 01 81 24
Email address: gwenole.quellec@inserm.fr (Gwenolé Quellec)
[1]www.who.int/blindness/causes/priority

algorithms supervised at the image level. A secondary objective is to use these local detections to improve DR detection at the image level.

In 2015, a machine learning competition was organized with the goal to design an automated system for grading the severity of diabetic retinopathy (DR) in images.[2] Images were trained and evaluated at the image level in a publicly-available dataset of almost 90,000 images provided by EyePACS (Cuadros and Bresnick, 2009), a free platform for DR screening. The performance criterion was the inter-rater agreement between the automated predictions and the predictions of human readers. As usual in recent pattern recognition competitions (Russakovsky et al., 2015), the top-ranking solutions all relied on deep learning. More precisely, they relied on ensembles of ConvNets. ConvNets are artificial neural networks where each neuron only processes one portion of the input image (LeCun et al., 2015). The main building-block of ConvNets are convolutional layers. In those layers, input images are convolved with multiple filters inside a sliding window (tens or hundreds of 3×3 to 5×5 multichannel filters, typically). After nonlinear post-processing, one activation map is obtained per filter. Those activation maps can be further processed by another convolutional layer, or can be nonlinearly down-sampled by a pooling layer. After several convolutional and pooling layers (10 to 30 layers, typically), ConvNets usually end with dense layers, which produce image-level predictions. Overall, ConvNets usually have a few million free parameters. The inter-rater agreement achieved by solutions of the challenge was clearly at the level of the inter-rater agreement among human readers (Barriga et al., 2014). However, many clinicians would not trust a black box, like a ConvNet (not to mention an ensemble of ConvNets), if their patient's health and their liability are at stake. Therefore, we are looking for a solution that jointly detects referable DR at the image level and detects biomarkers of this disease at the pixel level. Because of their good performance, solutions of the Kaggle DR challenge were reused and modified to also provide pixel-level detections. The proposed modifications rely on ConvNet visualization techniques. The resulting solution is part of the Messidor® system for eye pathology screening,[3] which integrates a mobile non-mydriatic retinograph and algorithms for automated or computer-aided diagnosis.

The remaining of this paper is organized as follows. Section 2 reviews the state of the art from an application point of view, namely deep learning for retinal image analysis, and from a methodological point of view, namely visualizing what ConvNets are learning. Section 3 describes the proposed lesion detection criterion. This criterion is improved by a novel optimization process in section 4. Section 5 presents experiments in three retinal image datasets (Kaggle, DiaretDB1 and e-ophtha). We end with a discussion and conclusions in section 6.

---

2. State of the Art

2.1. Deep Learning for Retinal Image Analysis

Deep learning was recently applied to various tasks related to retinal image analysis. For landmark segmentation and lesion detection, it was applied at a pixel level. For pathology detection, it was applied at an image level.

At a pixel level, a few algorithms were proposed for segmenting retinal vessels (Maji et al., 2015, 2016; Li et al., 2016) and the optic disc (Lim et al., 2015; Srivastava et al., 2015); others were proposed for detecting microaneurysms (Haloi, 2015), hemorrhages (van Grinsven et al., 2016) and exudates (Prentasic and Loncaric, 2015), three lesions related to DR; another one was proposed for detecting various lesions (including hemorrhages and exudates) and normal anatomical structures (Abràmoff et al., 2016). First, Maji et al. (2015, 2016) use an ensemble of ConvNets to classify each pixel as 'part of a vessel' or 'not part of a vessel'. Similarly, Lim et al. (2015) use a ConvNet to classify each pixel as 'part of the optic cup', 'part of the optic disc minus the optic cup' or 'not part of the optic disc'. Alternatively, Srivastava et al. (2015) use a network composed of (unsupervised) stacked auto-encoders followed by a supervised layer to classify each pixel as 'part of the optic disc' or 'not part of the optic disc'. For lesion detection, Haloi (2015), van Grinsven et al. (2016), and Prentasic and Loncaric (2015) use a ConvNet to classify pixels as 'part of a target lesion (a microaneurysm, a hemorrhage, or an exudate, respectively)' or 'not part of a target lesion'. Abràmoff et al. (2016) also use ConvNets to detect lesions or normal anatomical structures. In those seven algorithms, each pixel is classified through the analysis of a squared region centered on the pixel. In contrast, full images are analyzed in Li et al. (2016): stacked auto-encoders trained on 'fundus photograph' / 'vessel segmentation map' pairs are used to generate vessel probability maps the size of fundus photographs. All those algorithms require manual segmentations of training images for supervision.

At an image level, algorithms were proposed for detecting glaucoma (Chen et al., 2015a,b), age-related macular degeneration (AMD) (Burlina et al., 2016) and retinopathy of prematurity (ROP) (Worrall et al., 2016). While Lim et al. (2015) detect glaucomatous patients using the standard cup-to-disc ratio, derived from their deep-learning-based segmentations of the optic disc and cup, Chen et al. (2015a,b) directly classify an image as 'glaucomatous' or 'non-glaucomatous' through the analysis of a large region of interest centered on the optic disc, using one or two ConvNets. To detect AMD, Burlina et al. (2016) use the OverFeat features, derived from a ConvNet trained on the very large, general-purpose ImageNet dataset: these features are used to train a linear support-vector machine (SVM). To detect ROP, Worrall et al. (2016) fine-tuned the GoogLeNet network, also pre-trained on ImageNet. Finally, Arunkumar and Karthigaikumar (2017) proposed an algorithm for differentiating multiple pathologies: AMD, DR, macula pucker, retinoblastoma, retinal detachment and retinitis pigmentosa. Similarly to the AMD detector, image features are extracted with a ConvNet and a multi-class SVM is used to differentiate the various pathologies.
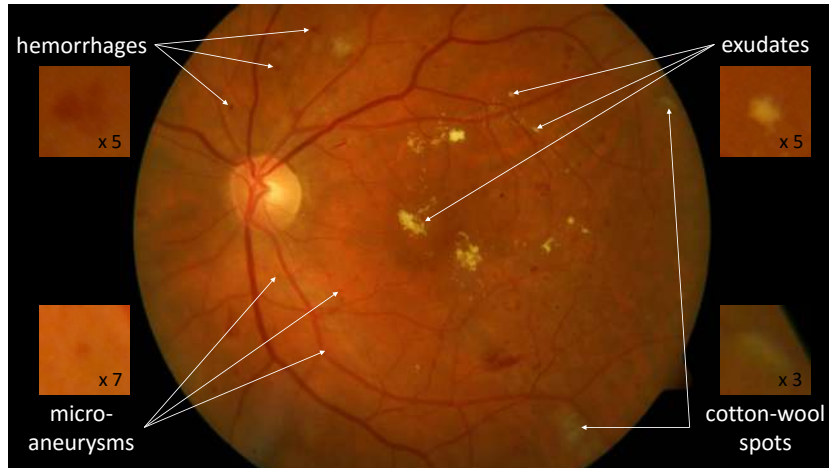
Figure 1: Most common lesions caused by diabetic retinopathy (image015 from the DiaretDB1 dataset with magnified lesions).

Finally, Colas et al. (2016) and Gulshan et al. (2016) also use deep learning techniques for detecting referable DR. In Colas et al. (2016), lesions are first detected (at the pixel level) and then DR severity is graded (at the image level). In Gulshan et al. (2016), the presence of referable DR is detected at the image level using an ensemble of ConvNets.

2.2. *Visualizing what ConvNets are Learning*

Because ConvNets are black boxes, many solutions have been proposed to visualize what they have learned. The earliest solutions consisted in visualizing the trained filters or intermediate activation maps. Given the large number of convolutional units in a ConvNet, it is hard to find out from a visual inspection which pattern each of them is looking for. One way to address this issue is to find which image, inside a large dataset, maximally activates each convolutional unit (Girshick et al., 2014) or to generate an artificial image that maximally activates it (Yosinski et al., 2015). Besides understanding the role of each convolutional unit, an additional question arises when training ConvNet at the image level: which regions or pixels, inside the image, play a role in the image-level prediction? A simple solution was first proposed by Zeiler and Fergus (2014): portions of the image are successively masked out with a sliding window and the image-level responses are analyzed: if a relevant image area is masked out, image-level recognition performance should decrease. This approach has several limitations: 1) from a computational point of view, images need to be processed many times and, more importantly, 2) redundant objects will not be detected. Typically, if a medical image contains several similar lesions, masking a single lesion out may not affect the diagnosis.

In our field, Worrall et al. (2016) proposed a visualization technique for ROP detection. This technique takes advantage of one specificity of GoogLeNet: activation values from all spatial locations in the deepest activation maps are averaged and the resulting average vector (one value per map) is processed with a softmax classifier. By removing the global average operator and applying the softmax classifier to each spatial location, relevant regions can be highlighted. The GoogLeNet network was modified to increase the definition of this visualization map from 7×7 pixels to 31×31 pixels, but this solution cannot provide pixel-level information, which can be a limitation when small lesions are involved.

In contrast, a set of methods was proposed to quantify how much each pixel impacts the image-level prediction, while analyzing the full image: the deconvolution method (Zeiler and Fergus, 2014), the sensitivity analysis (Simonyan et al., 2014) and layer-wise relevance propagation (Bach et al., 2015). These methods allow a visualization in terms of a heatmap the size of the input image. These algorithms have in common that the image only needs to be processed twice: the image data is propagated forward through the network and gradients of the image-level predictions, or similar quantities, are propagated backwards. The simplest solution (Simonyan et al., 2014), for instance, computes the partial derivative of the image-level predictions with respect to the value of each pixel: the backpropagated quantities are partial derivatives of the image-level predictions. The most advanced solution (Bach et al., 2015) forces the backpropagated quantities to be preserved between neurons of two adjacent layers. A detailed comparison can be found in Samek et al. (2016).

For the joint detection of referable DR and DR lesions, we need a solution which can deal with multiple occurrences of the same lesion, unlike Zeiler and Fergus (2014)'s solution, and which can deal with small lesions like microaneurysms, unlike Worrall et al. (2016)'s solution. The above pixel-level visualization techniques are more relevant to our task. However, we will show that the heatmaps they produce contain artifacts caused by the architecture of ConvNets. We propose to reduce those artifacts through a joint optimization of the ConvNet predictions and of the produced heatmaps. Among those three algorithms, sensitivity analysis (Simonyan et al., 2014) is the only criterion that can be easily differentiated, and which is therefore compatible with the proposed optimization: we decided to base our solution on this criterion. To go from visualizations designed to help understand what ConvNets are learning to visualiza-

tions useful for computer-aided diagnosis, the quality of the produced heatmaps needs to be improved, as presented hereafter.

3. Heatmap Generation

3.1. Notations

Let L denote the number of layers in a ConvNet. Let $D^{(l)}$, $l = 0, ..., L$, denote the data flowing from layer l to layer $l + 1$: $D^{(0)}$ denotes the input data, $D^{(l)}$, $1 \le l \le L - 1$, is composed of activation maps and $D^{(L)}$ contains the image-level predictions. For faster computations, ConvNets usually process multiple images simultaneously, so $D^{(0)}$ is generally a mini-batch of N images. $D^{(l)}$ is organized as a fourth-order tensor with dimensions $N \times W_l \times H_l \times C_l$, where $W_l \times H_l$ is the size of the activation maps produced by layer l (or the size of the input images if $l = 0$) and C_l is the number of activation maps per image (or the number of color channels if $l = 0$). In dense layers, such as the prediction layer, each map contains a single value ($W_l = H_l = 1$): in that case, $D^{(l)}_{n,1,1,c}$ is written $D^{(l)}_{n,c}$ for short.

3.2. Sensitivity Criterion

The sensitivity criterion assesses the contribution of one color channel of one pixel, namely $D^{(0)}_{n,x,y,c}$, to the ConvNet's prediction that the image belongs to some class d, namely $D^{(L)}_{n,d}$. This criterion is defined as the absolute value of the partial derivative of $D^{(L)}_{n,d}$ with respect to $D^{(0)}_{n,x,y,c}$, which can be computed according to the chain rule of derivation:

$$\frac{\partial D^{(L)}_{n,d}}{\partial D^{(l-1)}} = \frac{\partial D^{(L)}_{n,d}}{\partial D^{(l)}} \frac{\partial D^{(l)}}{\partial D^{(l-1)}}, \quad l = L - 1, ..., 1. \quad (1)$$

Here, we focus on the 'referable DR' class ($d = $'+'). If we denote by $f_n : \mathbb{R}^{N \times W_0 \times H_0 \times C_0} \to \mathbb{R}$ the ConvNet's prediction that image n belongs to the 'referable DR' class, the overall contribution $\omega_{n,x,y}$ of pixel $D^{(0)}_{n,x,y}$ can be summarized as follows (Simonyan et al., 2014):

$$\omega_{n,x,y} = \left\| \left(\frac{\partial D^{(L)}_{n,+}}{\partial D^{(0)}_{n,x,y,c}} \right)_{c \in \{r,g,b\}} \right\|_q = \left\| \left(\frac{\partial f_n(D^{(0)})}{\partial D^{(0)}_{n,x,y,c}} \right)_{c \in \{r,g,b\}} \right\|_q , \quad (2)$$

where $\|.\|_q, q \in \mathbb{N}$, denotes the q-norm; Simonyan et al. (2014) used $q = \infty$.

3.3. Interpretation

The sensitivity criterion indicates which local changes would modify the network predictions. In the context of DR screening, this can be interpreted as follows. Let us assume that a fundus image is from a referable DR patient and classified as such by the ConvNet. In that case, any lesion should be associated with nonzero ω values, as removing the lesion might downgrade the diagnosis and enhancing the lesion would consolidate it. Now, let us assume that the image is not from a referable DR patient and classified as such by the ConvNet. In that case, subtle microaneurysms in mild nonproliferative DR patients, which are not referable yet, should be associated with nonzero ω values.

Indeed, more pronounced microaneurysms would possibly upgrade the diagnosis to referable DR. So, in both cases, lesions should be detected by the ω criterion.

Although this criterion has interesting features for lesion detection, it also has a few drawbacks: two types of artifacts may appear, so we had to modify this criterion accordingly.

3.4. Disguise Artifacts

The first limitation of the sensitivity criterion is that it does not reveal directly whether a pixel contains evidence for or against the prediction made by a ConvNet: it simply gives, for every pixel, a direction in RGB-space in which the prediction increases or decreases (Samek et al., 2016). In particular, nonzero ω values may also be associated with lesion confounders: dust on the camera's lens resembling microaneurysms, specular reflections resembling exudates or cottonwool spots, etc. Indeed, modifying a confounder could make it resemble a true lesion even more. Typically, changing one or two color channels only would give it a more compatible color. Nonzero ω values may also be associated with healthy tissue surrounding a lesion: changing its color could make the lesion look bigger. We will refer to these phenomena as 'disguise artifacts'.

3.5. Hue-Constrained Sensitivity Criterion

The solution we propose to reduce those artifacts is to constrain the allowed local changes. Specifically, we force these changes to preserve the hue, so that sensitivity analysis essentially focuses on pattern enhancements or attenuations. Hue preservation is ensured by forcing all three color components of a pixel to be multiplied by the same factor. So, instead of computing the sensitivity of each color component independently and combining them afterwards (following Equation (2)), a single sensitivity value $\pi_{n,x,y}$ is computed per pixel in a single operation, as described hereafter. Given the input tensor $D^{(0)}$ with dimensions $N \times W_0 \times H_0 \times C_0$, a binary tensor m with dimensions $N \times W_0 \times H_0 \times 1$ is defined. The proposed hue-constrained sensitivity criterion is given by:

$$\pi_{n,x,y} = \left| \frac{\partial f_n(m \circ D^{(0)})}{\partial m_{n,x,y}} \right| , \quad (3)$$

where tensor m is filled with ones and where '$\circ$' denotes the entrywise tensor multiplication, which implies that $m \circ D^{(0)} = D^{(0)}$. Following the usual convention, the fact that the fourth dimension of m is 1 implies that all color components of a pixel in $D^{(0)}$ are multiplied by the same tensor element in m, which ensures the desired hue preservation property.

3.6. Drafting Artifacts

The second and most important limitation of the sensitivity criterion is that confounders in the vicinity of true lesions tend to be amplified. This effect, illustrated in Fig. 2, is due to downsampling, which occurs in pooling or convolutional layers with a stride greater than one (see Appendix A). Indeed, according to the chain rule of derivation [see Equation (1)], the gradient
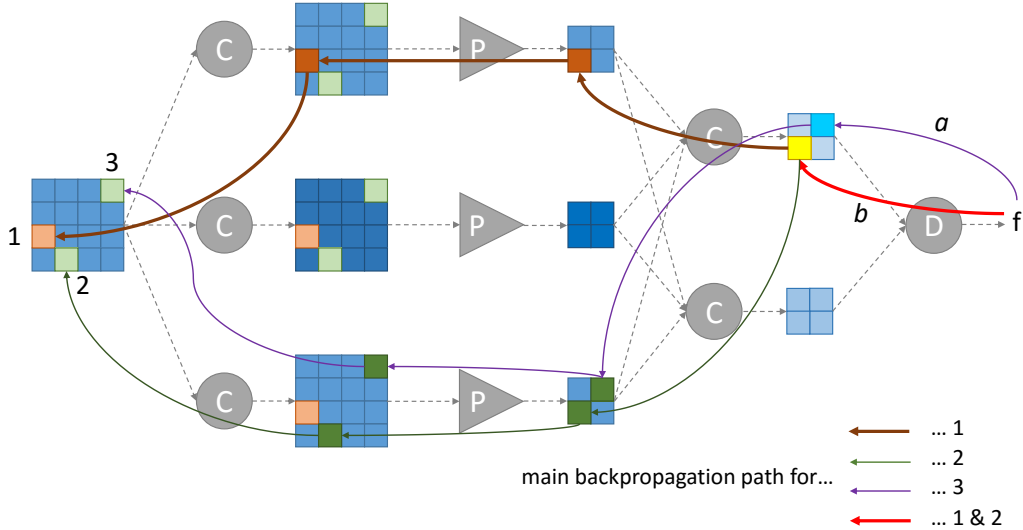
Figure 2: Illustrating drafting artifacts: one lesion ('1', in red) and two confounders ('2' and '3', in green) are processed by a simplistic ConvNet. Convolution operations (stride = 1) and max-pooling operations (size = 2 × 2, stride = 2) are indicated by letters 'C' and 'P', respectively; dense layers are indicated by letter 'D'. Colors in intermediate feature maps represent the contribution of each input pattern (red + green → yellow, green + blue → cyan). Because pattern '1' is a lesion, partial derivatives associated with edges along its backpropagation paths (such as edge 'b') tend to be larger (compared to edge 'a' in particular). Because they are neighbors, patterns '1' and '2' share the first edge ('b') along their main backpropagation path. So, even though patterns '2' and '3' are identical, the sensitivity of output 'f' with respect to '3', obtained by multiplying partial derivatives along its backpropagation paths, is smaller than the sensitivity with respect to '2', which benefits from a drafting effect from '1'.

of $f_n(D^{(0)})$ with respect to $D^{(0)}$ is obtained by multiplying the following partial derivative tensors: $\frac{\partial D^{(1)}}{\partial D^{(0)}}$, ..., $\frac{\partial D^{(L)}}{\partial D^{(L-1)}}$. Because of down-sampling, these partial derivative tensors are of decreasing sizes. As a result, a true lesion and confounders in its vicinity share common terms in the expression of their influence on $f_n(D^{(0)})$. These terms tend to be large because of the true lesion, so the perceived influence of the confounders is artificially boosted. We will refer to those amplified false alarms as 'drafting artifacts'. It should be noted that those drafting artifacts occur in all algorithms based on backpropagation, including the deconvolution method (Zeiler and Fergus, 2014) and layer-wise relevance propagation (Bach et al., 2015).

3.7. Reducing Drafting Artifacts

A brute-force solution for reducing those artifacts would be to 1) compute the π tensor according to Equation (3), 2) record the maximal π_{n,x^*,y^*} values and 3) set the corresponding m_{n,x^*,y^*} values to zero. Then, the π tensor should be computed again using the modified m tensor, in order to record the next largest $\pi_{n,x^{**},y^{**}}$ values, without the drafting influence of the (n, x^*, y^*) pixels. And so on until the influence of each pixel has been recorded independently from its more influential neighbors. However, the complexity of this solution clearly is prohibitive. Instead, we propose an indirect solution which reduces drafting artifacts while training the ConvNet, so that we do not have to deal with them explicitly afterwards.

4. Heatmap Optimization

4.1. Training a ConvNet with the Backpropagation Method

The parameters of a ConvNet (namely weights and biases — see Appendix A) are generally optimized with the backpropa-

gation method. This method progressively calculates the gradient of a loss function $\mathcal{L}_L$ with respect to each parameter tensor ρ, using the chain rule of derivation:

$$\frac{\partial \mathcal{L}_L}{\partial D^{(l-1)}} = \frac{\partial \mathcal{L}_L}{\partial D^{(l)}} \frac{\partial D^{(l)}}{\partial D^{(l-1)}} , \tag{4}$$

$$\frac{\partial \mathcal{L}_L}{\partial \rho} = \frac{\partial \mathcal{L}_L}{\partial D^{(l)}} \frac{\partial D^{(l)}}{\partial \rho} . \tag{5}$$

Those gradients are then used by an optimizer to update the parameters, in an attempt to minimize the loss function. Loss function $\mathcal{L}_L$ typically quantifies the classification or regression error, based on image-level predictions computed in $D^{(L)}$. To improve generalization, regularization terms are usually added to the loss function: they typically aim at minimizing the L1- or L2-norm of the filter weights.

4.2. Sparsity-Enhanced Sensitivity Criterion

In order to reduce drafting artifacts, we propose to include an additional regularization term $\mathcal{L}_0$ to the total loss function $\mathcal{L}$. The aim of $\mathcal{L}_0$ is to maximize the sparsity of ω or π. By forcing the ConvNet to reduce the number of nonzero pixels in ω or π, while maintaining its classification or regression accuracy, the ConvNet has to modify its parameters in such a way that true lesions and confounders in their vicinity share as little large terms as possible in the expression of their influence on $f_n(D^{(0)})$. In other words, the ConvNet is forced to build more discriminative filters: filters that better separate true lesions from confounders. Following Tibshirani (1996), the sparsity of ω or π is maximized through L1-norm minimization, rather than L0-norm minimization, which is NP-hard.
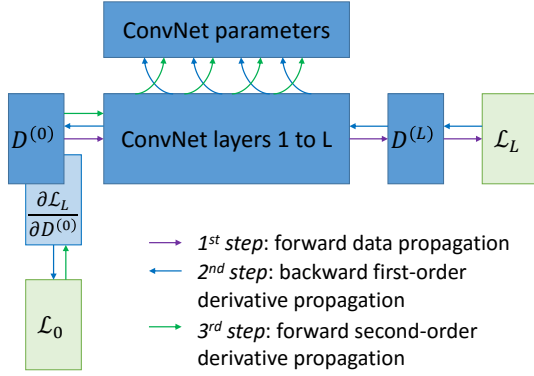
Figure 3: Backward-forward propagation method.

Because those heatmaps depend on backpropagated quantities, the network parameters cannot be optimized using the usual backpropagation method, so a different ConvNet training method had to be proposed: while standard training algorithms propagate image intensities through the ConvNet and backpropagate the gradients of the optimization criterion, the proposed training algorithm involves a third pass on the ConvNet to propagate second-order derivatives forward. This new training procedure can be obtained through simple adaptations of deep learning libraries.

4.3. Backward-Forward Propagation Method

We introduce a novel method for optimizing the parameters of a ConvNet when the loss function to minimize is of the form:

$$
\begin{cases}
\mathcal{L} & = & \mathcal{L}_L + \mathcal{L}_0 \\
\mathcal{L}_0 & = & g\left(\dfrac{\partial \mathcal{L}_L}{\partial D^{(0)}}\right)
\end{cases} \quad . \tag{6}
$$

In order to enhance the sparsity of ω maps, function g is set to the L1-norm, multiplied by some factor ν. The sparsity of π maps is enhanced similarly: in this case, the role of the input data tensor is played by tensor m [see Equation (3)].

The proposed algorithm, illustrated in Fig. 3, updates each parameter tensor ρ during one training iteration, as follows:

1. The data is propagated forward through the network, from $D^{(0)}$ to $D^{(L)}$, in order to compute loss function $\mathcal{L}_L$.

2. The gradient of $\mathcal{L}_L$ is propagated backward through the network, from $D^{(L)}$ to $D^{(0)}$, using Equations (4) and (5). The goal is to compute $\frac{\partial \mathcal{L}_L}{\partial \rho}$, for each parameter tensor ρ, and also to compute $\mathcal{L}_0$.

3. The gradient of $\mathcal{L}_0$ is propagated forward through the first-order derivative of the network, from $D^{(0)}$ to $D^{(L)}$, in order to compute $\frac{\partial \mathcal{L}_0}{\partial \rho}$, for each parameter tensor ρ.

4. Each parameter tensor ρ is updated proportionally to $\frac{\partial \mathcal{L}_L}{\partial \rho} + \frac{\partial \mathcal{L}_0}{\partial \rho}$.

The proposed backward-forward propagation method can be implemented simply using deep learning libraries with built-in automatic differentiation, such as TensorFlow[4]. The main difficulty is to provide a forward second-order derivative function for each operator in the network, in order to perform step 3, while deep learning libraries only provide backward first-order derivatives. Appendix B provides the forward second-order derivatives for operators used in the following experiments.

The proposed heatmap generation and optimization solution is now evaluated in the context of diabetic retinopathy screening, using ConvNets specifically designed for this task.

5. Experiments

5.1. Baseline ConvNet

This section introduces the ConvNets that we used in our experiments. These ConvNets produce predictions at the image level; based on modifications described in sections 3 and 4, they also produce predictions at the pixel level. Successful solutions based on ConvNets were submitted to the 2015 Diabetic Retinopathy competition and the source code of the best solutions is publicly available. Rather than reinventing the wheel, we used the best of these solutions to set up our baseline ConvNets.

5.1.1. Image Preprocessing and Data Augmentation

Image preprocessing was adapted from the min-pooling solution,[5] by B. Graham, which ranked first in the Kaggle Diabetic Retinopathy competition. Let I denote the input image. The width of the camera's field of view in I is estimated and I is resized to normalize that width to 512 pixels. The background I_b of the resulting I_r image is then estimated by a large Gaussian filter in each color channel (standard deviation: 8.5 pixels). A normalized image is defined as $I_n = 4(I_r - I_b)$. Finally, because the camera's field of view usually contains illumination artifacts around its edges, the field of view is eroded by 5% in I_n. Following all the top ranking solutions in the competition, data augmentation is performed during training. Before feeding a preprocessed image to the network, the image is randomly rotated (range: $[0°, 360°]$), translated (range: [-10 px, 10 px]), scaled (range: $[85\%, 115\%]$), horizontally flipped and its contrast is modified (multiplicative factor range: $[60\%, 167\%]$); different transformation parameters are generated at each epoch. The resulting image is resized and cropped to 448×448 pixels.

5.1.2. Network Structures

The network structures used in this study were adapted from the o_O solution,[6] by M. Antony and S. Brüggemann, which ranked second in the Kaggle Diabetic Retinopathy competition. This solution was selected since it relies on networks composed

---

[4]https://www.tensorflow.org — one line implementation of the $\mathcal{L}_0$ cost function, if g is the L1-norm: tf.add_to_collection('losses',tf. reduce_sum(tf.abs(tf.gradients(L_L, [D_l])[0])))

[5]https://www.kaggle.com/c/diabetic-retinopathy-detection/ forums/t/15801/competition-report-min-pooling-and-thank-you

[6]https://www.kaggle.com/c/diabetic-retinopathy-detection/ forums/t/15617/team-o-o-solution-summary

exclusively of basic processing units implemented in all deep learning libraries. This property does not apply to the min-pooling solution, in particular, which relies on specialized operations such as fractional max pooling (Graham, 2014).

The o_O solution relies on two networks, namely 'net A' and 'net B', applied to images of size 448 × 448 pixels. Their structure is described in table 1. It also relies on two sub-networks of 'net A' and 'net B' applied to smaller images (224 × 224 pixels and 112 × 112 pixels). All convolutional and dense layers use untied biases and leaky rectifiers as activation functions (see Appendix A). The last dense layer with a single unit is used for regression, to predict the image label. The designers of o_O noticed that 'net B' alone works almost as well as the ensemble, so we studied 'net B' in more detail.

To show the generality of the approach, the popular AlexNet structure was also evaluated (Krizhevsky et al., 2012). Unlike 'net A' and 'net B', AlexNet processes images of size 224 × 224 pixels, so images had to be downsampled by a factor of 2. Downsampling was done dynamically, using a mean pooling operator with a stride of 2 and a window size of 2 × 2 pixels (see Appendix A), in order to produce heatmaps with 448 × 448 pixels.

5.1.3. Network Training

Following o_O, networks are trained to minimize the mean squared error between image labels and predictions. Additionally, L2 regularization with factor 0.0005 is applied to filter weights in all convolutional and dense layers. We use very leaky rectifiers ($\alpha = 0.33$) instead of leaky rectifiers ($\alpha = 0.01$) in o_O. This allows us to train all layers simultaneously, using the Adam optimizer (Kingma and Ba, 2015). Antony and Brüggemann tried a similar strategy, but with the optimizer they used, namely the Nesterov momentum optimizer (Nesterov, 1983), it did not work well. A learning rate of 0.0001 was used initially. Following common practice, we manually decreased the learning rate by a factor of 10 when performance in the validation set stopped increasing.

5.2. Implementation Details

The proposed algorithms were implemented in C++ and Python using OpenCV[7] for image preprocessing and data augmentation, and TensorFlow for network training and inference. Forward second-order derivatives were implemented in Python when possible; that of MaxPool was implemented in C++. One GPU card was used: a GeForce GTX 1070 by Nvidia. Training and testing were performed using mini-batches of $N = 36$ images, in accordance with the memory capacity of the GPU card (7.92 GB). Each ConvNet was trained with 350,000 mini-batches, i.e. with 350,000 $N = 12.6$ million images generated through data augmentation.

5.3. Datasets

Three datasets were used in this study: the 'Kaggle Diabetic Retinopathy' dataset, used for training and testing at the image level, 'DiaretDB1', for testing at the lesion level and at the image level and also for improving performance at the image level, and finally 'e-ophtha', for testing at the image level.

5.3.1. Kaggle Diabetic Retinopathy Dataset

The first dataset consists of 88,702 color fundus photographs from 44,351 patients: one photograph per eye.[8] Images were captured with various digital fundus cameras, in multiple primary care sites throughout California and elsewhere. Their definitions range from 433 x 289 pixels to 5184 x 3456 pixels (median definition: 3888 x 2592 pixels). Those images were then uploaded to EyePACS, a free platform for DR screening (Cuadros and Bresnick, 2009). For each eye, DR severity was graded by a human reader according to the ETDRS scale (Wilkinson et al., 2003): 'absence of DR', 'mild non-proliferative DR (NPDR)', 'moderate NPDR', 'severe NPDR' and 'proliferative DR (PDR)'. The dataset was split into a training set (35,126 images from 17,563 patients) and a test set (53,576 images from 26,788 patients): those two sets are referred to as 'Kaggle-train' and 'Kaggle-test', respectively. Networks were trained on 80 % of the Kaggle-train dataset (the first 28,100 images) and validated on the remaining 20 % (the last 7,024 images).

For the purpose of this study, about DR screening, severity grades were grouped into two categories: nonreferable DR (absence of DR or mild NPDR) versus referable DR (moderate NPDR or more). The prevalence of referable DR was 19.6 % in Kaggle-train and 19.2 % in Kaggle-test.

5.3.2. DiaretDB1 Dataset

The second dataset consists of 89 color fundus photographs collected at the Kuopio University Hospital, in Finland (Kauppi et al., 2007). Images were captured with the same fundus camera, a ZEISS FF450*plus* digital camera with a 50 degree field-of-view. Images all have a definition of 1500 x 1152 pixels. Independent markings were obtained for each image from four medical experts. The experts were asked to manually delineate the areas containing microaneurysms (or 'small red dots'), hemorrhages, hard exudates and cotton wool spots (or 'soft exudates') and to report their confidence (< 50 %, ≥ 50 %, 100 %) for each segmented lesion. Based on these annotations, only five images in the dataset are considered normal: none of the experts suspect these images to contain any lesions.

Given a target lesion type, Kauppi et al. (2007) proposed a standardized procedure to evaluate the performance of a lesion detector, at the image level, in the DiaretDB1 dataset. In this purpose, one probability map was constructed per image: this map was obtained by averaging, at the pixel level, confidences from all four experts for the target lesion type. If and only if this map contains at least one pixel with an average confidence level above 75 %, then the image is considered to contain the target lesion. Based on this criterion, a receiver-operating characteristic (ROC) curve can be constructed for the lesion detector.

---

id	layer type	activation maps	net 'A'			net 'B'		
			window size	window stride	output tensor size	window size	window stride	output tensor size
1	Input				448 x 448			448 x 448
2	Conv	32	5 x 5	2	224 x 224	4 x 4	2	224 x 224
3	Conv	32	3 x 3	1	224 x 224	4 x 4	1	225 x 225
4	MaxPool	∅	3 x 3	2	111 x 111	3 x 3	2	112 x 112
5	Conv	64	3 x 3	2	56 x 56	4 x 4	2	56 x 56
6	Conv	64	3 x 3	1	56 x 56	4 x 4	1	57 x 57
7	Conv	64	3 x 3	1	56 x 56	4 x 4	1	56 x 56
8	MaxPool	∅	3 x 3	2	27 x 27	3 x 3	2	27 x 27
9	Conv	128	3 x 3	1	27 x 27	4 x 4	1	28 x 28
10	Conv	128	3 x 3	1	27 x 27	4 x 4	1	27 x 27
11	Conv	128	3 x 3	1	27 x 27	4 x 4	1	28 x 28
12	MaxPool	∅	3 x 3	2	13 x 13	3 x 3	2	13 x 13
13	Conv	256	3 x 3	1	13 x 13	4 x 4	1	14 x 14
14	Conv	256	3 x 3	1	13 x 13	4 x 4	1	13 x 13
15	Conv	256	3 x 3	1	13 x 13	4 x 4	1	14 x 14
16	MaxPool	∅	3 x 3	2	6 x 6	3 x 3	2	6 x 6
17	Conv	512	3 x 3	1	6 x 6	4 x 4	1	5 x 5
18	Conv	512	3 x 3	1	6 x 6	∅	∅	∅
19	RMSPool	∅	3 x 3	3	2 x 2	3 x 3	2	2 x 2
20	Dropout	∅						
21	Dense	1024						
22	Maxout	512						
23	Dropout	∅						
24	Dense	1024						
25	Maxout	512						
26	Dense	1						

Table 1: Network Structure (see Appendix A) — ConvNets from the o_O solution.

5.3.3. e-ophtha Dataset

The third dataset consists of 107,799 photographs from 25,702 examination records: generally two photographs per eye, i.e. four photographs per examination record. These photographs were collected in the OPHDIAT screening network in the Paris area (Erginay et al., 2008). Images were captured either with a CR-DGi retinograph (Canon, Tokyo) or with a TRC-NW6S (Topcon, Tokyo) retinograph. Their definitions range from 1440×960 to 2544×1696 pixels. Up to 27 contextual fields were included in each record. This includes 9 demographic information fields (age, gender, weight, etc.) and 18 information fields related to diabetes. Each examination record was analyzed later on by one ophthalmologist, out of 11 participating ophthalmologists, in Lariboisière Hospital (Paris). The ophthalmologist graded DR severity in both eyes. It should be noted that the association between photographs and eyes is unknown. Therefore, the task we address in this study is to detect whether or not the patient has referable DR in at least one of his or her eyes. The dataset was split into a training set of 12,849 examination records ('e-ophtha-train') and a test set of 12,853 records ('e-ophtha-test'), described in Quellec et al. (2016c).

5.4. Visualization Artifacts

Artifacts from various visualization algorithms are illustrated in Fig. 4. First, it can be seen that the original sensitivity criterion is inadequate to finely detect lesions. Sensitivity maps seem to indicate that if lesions grew in size, the diagnosis would be consolidated. The hue constraint prevents the lesion detections from growing. Second, it can be seen that, due to the drafting effect, blood vessels (lesion confounders) in the vicinity of lesions are detected, both in the hue-constrained sensitivity maps and in the layer-wise relevance propagation maps. The resulting false detections are not necessarily connected to the true detection, so they cannot be removed easily through image post-processing techniques: they have to be removed beforehand, hence the proposed approach.

5.5. Image- and Pixel-Level Performance of ConvNets

Figure 5 reports the performance of 'net B' at different checkpoints stored during the training process. The hue-constrained sensitivity criterion is used, with or without sparsity maximization. Performance at the image level was assessed using a ROC analysis in the validation subset of Kaggle-train ('Kaggle-validation' for short), as well as in Kaggle-test: the area A_z under the ROC curve is used as performance metric. Performance at the lesion level was assessed using a free-response ROC (FROC) analysis in the DiaretDB1 dataset. FROC curves are usually not bounded along the x-axis (the number false positives per image): we used as performance metric the area A_z under the FROC curve for $0 \leq x \leq \mu$, divided by μ (with $\mu = 10$). Performance is computed for each lesion type independently, and an average performance metric $\bar{A}_z$ is also computed. The influence of ν, the factor of the $\mathcal{L}_0$ cost function, on the performance at the image level (A_z in the Kaggle validation set) and at the lesion level ($\bar{A}_z$ in the DiaretDB1 dataset), is given in table 2.

To assess detection performance for each lesion type, a ten-fold cross-validation was performed. For each fold,
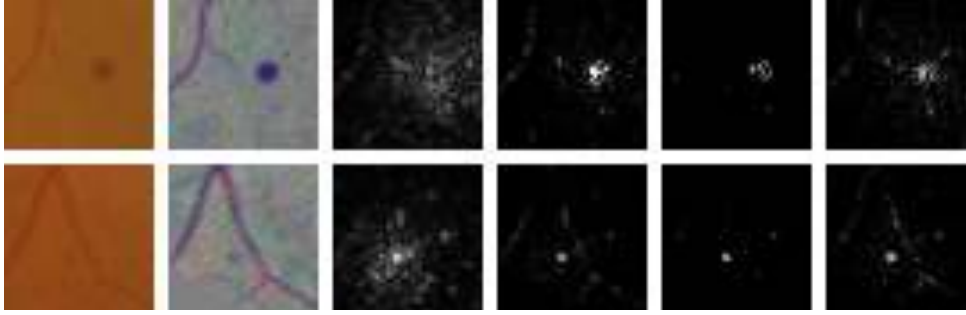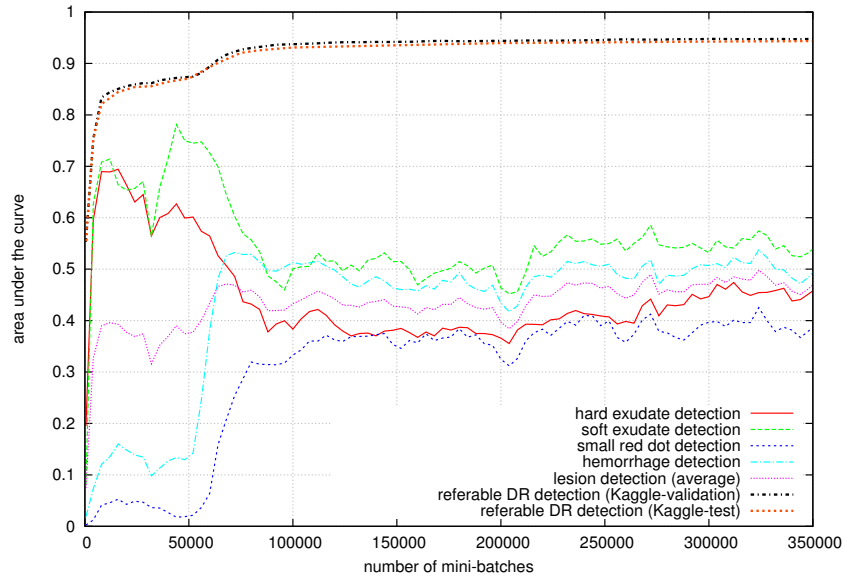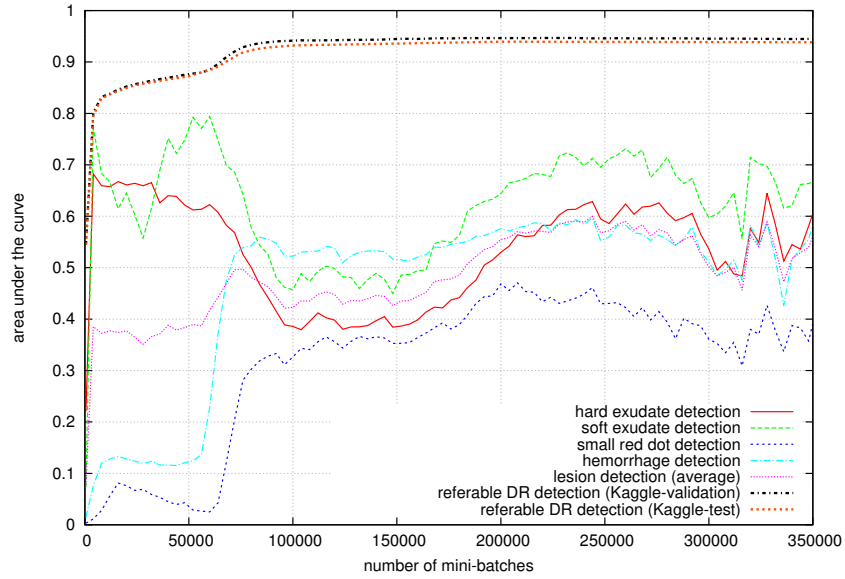
Figure 4: Artifacts from various visualization algorithms using 'net B'. From left to right: the original image, the preprocessed image, the original sensitivity map, the hue-constrained sensitivity map, the hue-constrained and sparsity-enhanced sensitivity map, and the layer-wise relevance propagation map.

(a) Hue-constrained sensitivity criterion — factor of the $\mathcal{L}_0$ cost function: $\nu = 0$

(b) Hue-constrained and sparsity-enhanced sensitivity criterion — factor of the $\mathcal{L}_0$ cost function: $\nu = 10^{-3}$

Figure 5: ROC analysis for 'net B' in Kaggle-validation and Kaggle-test — FROC analysis in DiaretDB1.

9

ν	image level A_z	lesion level $\bar{A}_z$
0	0.948	0.499
10^{-4}	0.948	0.543
10^{-3}	**0.947**	**0.600**
10^{-2}	0.546	0.375

Table 2: Influence of ν, the factor of the $\mathcal{L}_0$ cost function, on the performance of 'net B' — the retained solution is in bold.

1. the optimal checkpoint, as well as the optimal ν value when applicable ($\nu \in \{10^{-4}, 10^{-3}, 10^{-2}\}$), were found using 90% of DiaretDB1 images,

2. lesion probabilities were computed for the remaining 10% using the optimal checkpoint and ν value.

FROC curves are reported in Fig. 6 for 'net B'; areas under the limited FROC curve for all three ConvNets are summarized in Table 3. It appears that the number of false alarms is rather large, particularly in the case of microaneurysm detection. The reason is that human experts primarily segmented the most obvious lesions, while screening algorithms need to focus on the most subtle lesions as well. In other words, many true lesions are counted as false alarms. Of course, this comment also applies to competing automatic solutions. To show the value of our detections, the proposed solution was compared in Fig. 7 to results reported in the literature, following the DiaretDB1 standardized procedure (see section 5.3.2). Most authors reported a single (sensitivity, specificity) pair: this is what we reported in Fig. 7. Some authors reported ROC curves; in that case, we also reported a single (sensitivity, specificity) pair: the one closest to the (sensitivity = 1, specificity = 1) coordinate. Note that all competing solutions (Kauppi et al., 2007; Yang et al., 2013; Franklin and Rajan, 2014; Kumar et al., 2014; Bharali et al., 2015; Mane et al., 2015; Dai et al., 2016) are trained at the lesion level, while ours is trained (in Kaggle-train) at the image level.

5.6. Ensemble Learning

As commonly done in machine learning competitions (Russakovsky et al., 2015), an ensemble of ConvNets was used to boost performance. As less commonly done, all ConvNets in the ensemble originate from the same network, but with parameter values obtained at different checkpoints during the learning process. This way, training the ensemble is not significantly more computationally intensive than training a single ConvNet. As shown in Fig. 5, individual lesion types are not optimally detected after the same number of iterations. So, the idea was to export parameter values from the ConvNet when:

1. hard exudates were optimally detected (network $\mathcal{N}_{HE}$ — iteration 4,000 for 'net B'),

2. soft exudates were optimally detected (network $\mathcal{N}_{SE}$ — iteration 60,000 for 'net B'),

3. small red dots were optimally detected (network $\mathcal{N}_{SRD}$ — iteration 208,000 for 'net B'),

4. hemorrhages were optimally detected (network $\mathcal{N}_H$ — iteration 244,000 for 'net B'),

5. lesions were optimally detected on average (network $\mathcal{N}_{AVG}$ — iteration 328,000 for 'net B'),

6. referable DR was optimally detected (network $\mathcal{N}_{RDR}$ — iteration 224,000 for 'net B').

In the Kaggle DR dataset, network predictions were computed for the current eye, but also for the contralateral eye, so $6 \times 2 = 12$ features were fed to the ensemble classifier per eye. In e-ophtha, network predictions were computed for all images in the current examination record. In regular examination records, consisting of four images, the four predictions computed at each checkpoint were stored in ascending order. In other records, the two lowest and the two largest predictions were stored. These $6 \times 4 = 24$ predictions were fed to the ensemble classifier, with up to 27 contextual features (see section 5.3.3). Following the min-pooling solution, a random forest was used to build the ensembles. These ensembles were trained in the full Kaggle-train and e-ophtha-train datasets, respectively.

Random forest parameters, the number of trees n_T and the maximum depth of each tree d_T, were optimized by 5-fold cross-validation in Kaggle-train and e-ophtha-train: $n_T = 500$ in Kaggle DR dataset and 300 in e-ophtha, $d_T = 25$ in both datasets. ROC curves in Kaggle-test and e-ophtha-test are reported in Fig. 8 for 'net B'. Areas under the ROC curves for all three ConvNets are summarized in Table 4. It is important to notice that using such an ensemble of ConvNets does not necessarily turn the solution into a black box. Indeed, the heatmaps associated with the above ConvNets all have the same size. So, to support decision for a given patient, the heatmaps associated with all the networks involved in the decision process can be blended (one blended heatmap per eye). Furthermore, each heatmap can be weighted by the importance of the associated ConvNet in the decision process.

The performance of the proposed pixel-level detector, using 'net B', is illustrated in Fig. 9 and 10 on two images from independent datasets. The first image comes from the publicly-available Messidor dataset[9]. The second image comes from a private dataset acquired with a low-cost handheld retinograph (Quellec et al., 2016a).

6. Discussion and Conclusions

A solution was proposed in this paper for the automatic detection of referable diabetic retinopathy (DR) and for the automatic detection of lesions related to DR. Unlike competing solutions, this lesion detector is trained using image-level labels only for supervision. The solution relies on ConvNets trained to detect referable DR at the image level. Using a modified sensitivity analysis, the pixels playing a role in the image-level predictions are detected: a heatmap the size of the image is obtained. In order to improve the quality of the heatmaps (attenuate artifacts), we proposed to enhance the sparsity of the heatmaps while training the ConvNets. Because those heatmaps depend on backpropagated quantities, the network parameters
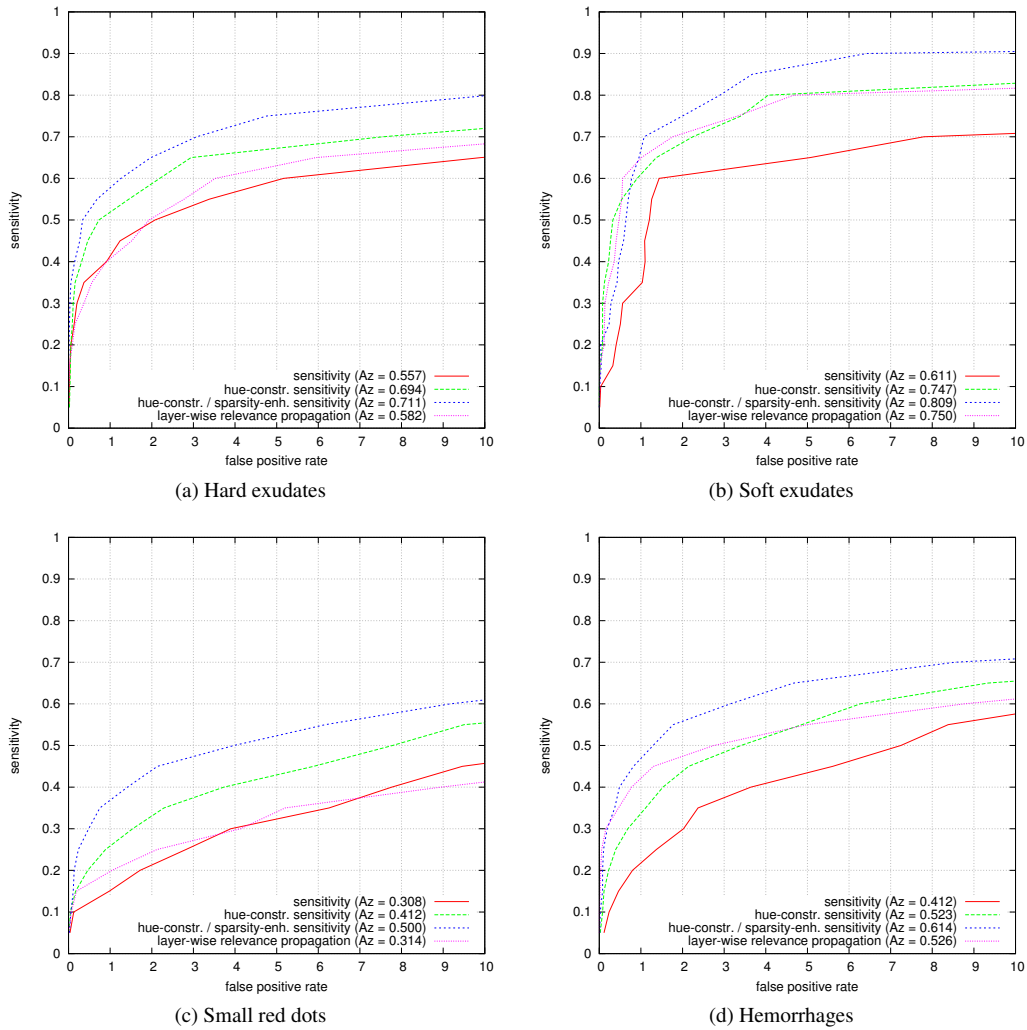
---

[9] http://www.adcis.net/en/Download-Third-Party/Messidor.html

10

(a) Hard exudates

(b) Soft exudates

(c) Small red dots

(d) Hemorrhages

Figure 6: Lesion detection performance in DiaretDB1 for 'net B' at the lesion level.

criterion	ConvNet	hard exudates	soft exudates	small red dots	hemorrhages
sensitivity	net A	0.552	0.624	0.316	0.449
	net B	0.557	0.611	0.308	0.412
	AlexNet	0.474	0.642	0.028	0.274
hue-constrained sensitivity	net A	0.677	0.754	0.393	0.533
	net B	0.694	0.747	0.412	0.523
	AlexNet	0.516	0.742	0.032	0.300
hue-constrained sparsity-enhanced sensitivity	net A	**0.735**	0.802	0.482	0.578
	net B	0.711	**0.809**	**0.500**	**0.614**
	AlexNet	0.618	0.783	0.032	0.320
layer-wise relevance propagation	net A	0.600	0.743	0.315	0.519
	net B	0.582	0.750	0.314	0.526
	AlexNet	0.499	0.747	0.027	0.274

Table 3: Lesion detection performance in DiaretDB1 at the lesion level — area under the FROC curve for $0 \leq x \leq 10$, divided by 10. The best ConvNet/criterion couples are in bold.

(a) Hard exudates

(b) Soft exudates

(c) Small red dots

(d) Hemorrhages

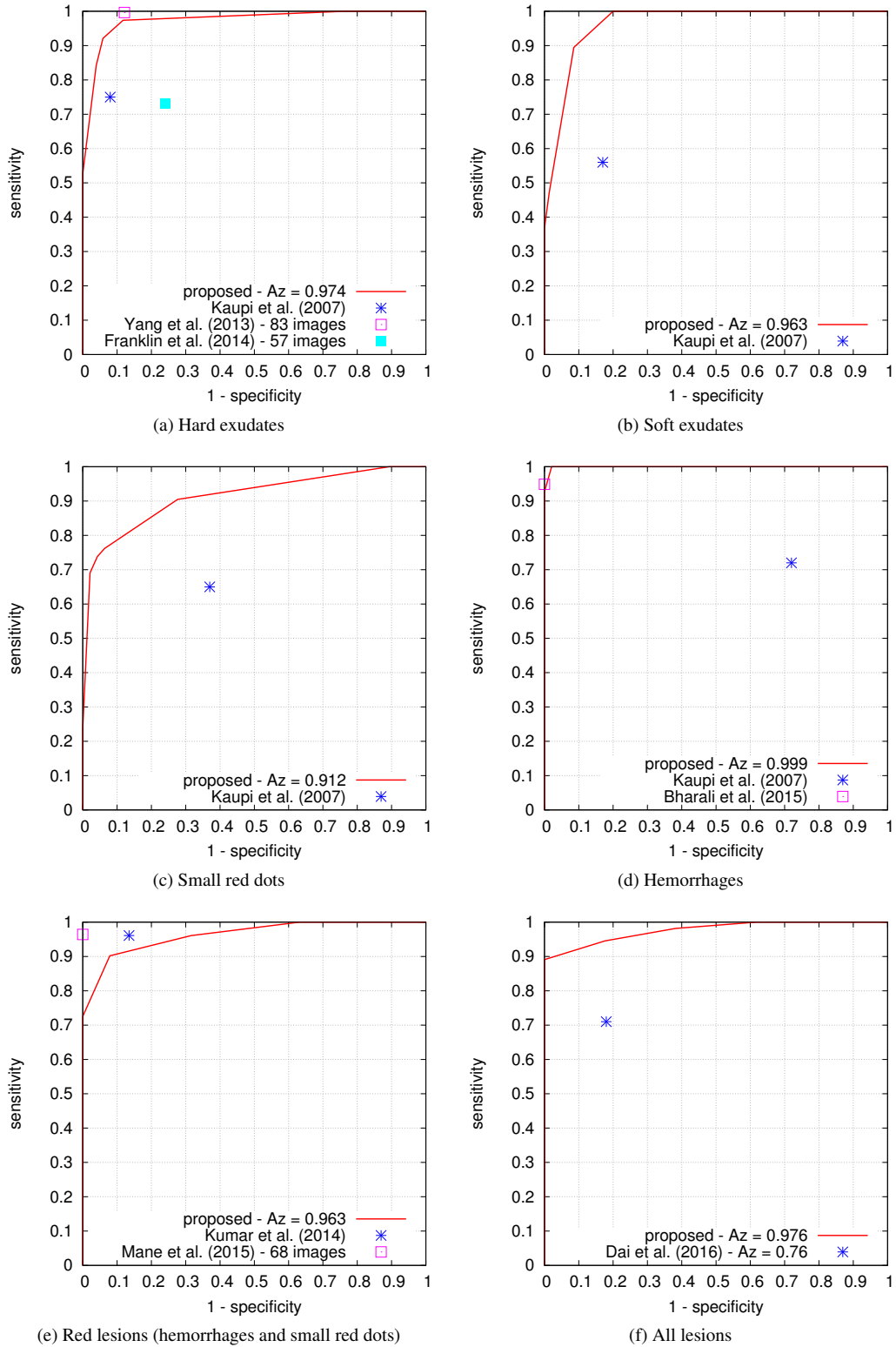(e) Red lesions (hemorrhages and small red dots)

(f) All lesions

Figure 7: Lesion detection performance for 'net B' in DiaretDB1, at the image level, following the standardized procedure.
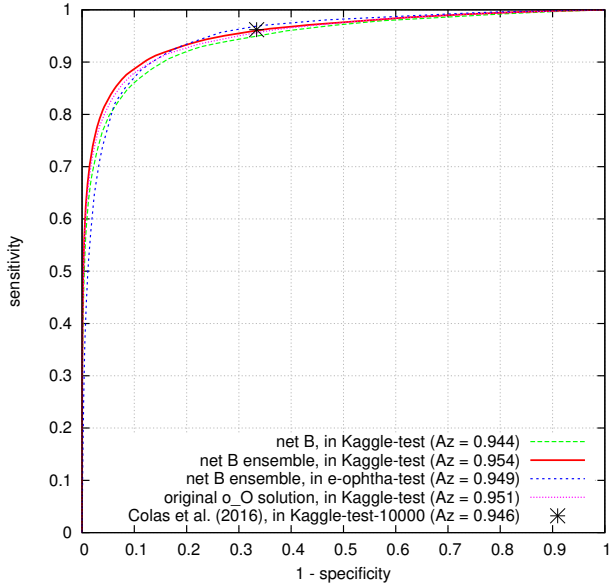
Figure 8: Referable diabetic retinopathy detection in Kaggle-test and e-ophtha-test using 'net B'. Kaggle-test-10000 denotes a subset of 10,000 images from Kaggle-test.

ensemble	ConvNet	Kaggle-test	e-ophtha-test
no	net A	0.940	∅
	net B	0.944	∅
	AlexNet	0.900	∅
yes	net A	0.952	0.948
	net B	0.954	0.949
	AlexNet	0.908	0.907
	all	0.955	0.949

Table 4: Referable diabetic retinopathy detection in Kaggle-test and e-ophtha-test — area under the ROC curve.

cannot be optimized using the usual backpropagation method, so a different ConvNet training method was proposed.

Three ConvNets were trained to detect referable DR in the Kaggle-train dataset, using the proposed heatmap optimization procedure. Then, we evaluated how well those ConvNets could detect lesions in the DiaretDB1 dataset, without retraining them. For lesion detection at the image level, they outperformed previous algorithms, which were explicitly trained to detect the target lesions, with pixel-level supervision (see Fig. 7). This superiority was observed for all lesions or groups of lesions, with the exception of 'red lesions'. Experiments were also performed at the lesion level: for all lesion types, the proposed algorithm was found to outperform recent heatmap generation algorithms (see Table 3). As illustrated in two examples (see Fig. 9 and 10), the produced heatmaps are of very good quality. In particular, the false alarms detected on the vessels, in the vicinity of true lesions in the unoptimized heatmaps (π_0 maps), are strongly reduced with sparsity maximization (π_{HE}, π_{RDR}, $\bar{\pi}$). These experiments validate the relevance of image-level supervision for lesion detectors, but stress the need to optimize the heatmaps, as proposed in this paper. Note that detection performance is not affected much by image quality:

very good detections are produced in the blurry image obtained with a low-cost, handheld retinograph (see Fig. 10). This is a very important feature, which opens the way to automated mobile screening. However, it can be observed that the 'AlexNet' architecture, which achieves moderate DR detection results, also achieves poor detection results at the lesion level, even after heatmap optimization (see Table 3): to ensure good detection performance at the lesion level, the proposed optimization framework should be applied to ConvNet architectures that achieve good image-level performance.

Besides analyzing the pixel-level performance of the final ConvNets, we also analyzed the pixel-level performance while the ConvNets are being trained (see Fig. 5 for the 'net B' ConvNet). It turned out to be quite instructive. By analyzing performance at the image level alone (the area under the ROC curve in the Kaggle validation set), all we can see is that 1) performance quickly reaches a satisfactorily level ($A_z \simeq 0.85$), then 2) slowly increases for many iterations, 3) experiences a second leap to reach a very high level ($A_z \simeq 0.93$) and finally 4) reaches its optimal state ($A_z \simeq 0.95$) very slowly. By analyzing the heatmaps, we understand that the ConvNet very quickly learns to detect exudates and cotton-wool spots (or 'soft exudates'). The second performance leap is observed when the ConvNet learns to detect hemorrhages. The final fine-tuning stage correlates with the progressive discovery of microaneurysms (or 'small red dots') by the ConvNet. Interestingly, lesions were discovered in the same order regardless of the ConvNet structure ('net A', 'net B' or AlexNet). The order in which lesions are discovered by ConvNets makes sense: the most obvious lesions (the largest and the most contrasted) are detected first and then the ConvNet discovers that more subtle lesions (which are more difficult to capture) are in fact more useful to make a diagnosis. By the way, for 'net A' and 'net B', we observe that the detection performance of bright lesions (exudates and cotton-wool spots) decreases when red lesions (hemorrhages and microaneurysms) are discovered: red lesions are indeed usually enough for detecting referable DR. This behavior is not observed for AlexNet: the reason probably is that red lesions are not detected well enough (see Table 3), due to the low resolution of input images (224×224 pixels). The main difference between the two plots of Fig. 5 (with or without sparsity maximization) is observed in the latest stages of training. As the artifacts are removed by enhancing the sparsity of the heatmaps, the detection performance at the pixel level increases for every lesion type. We hypothesized that maximizing the sparsity of the heatmaps would also speed up the training process, by reducing the search space. However, we did not observe such a behavior.

Performance at the image level is also very high, which was to be expected since we used efficient ConvNets from a recent machine learning competition as starting points: a performance of $A_z = 0.9542$ was achieved in Kaggle-test by the proposed framework using 'net B' (95 % confidence interval with DeLong et al. (1988)'s method: [0.9515, 0.9569]). In particular, the proposed solution outperforms the system by Colas et al. (2016) ($A_z = 0.946$). This good performance derives in part from the above observations at the pixel level, which ex-
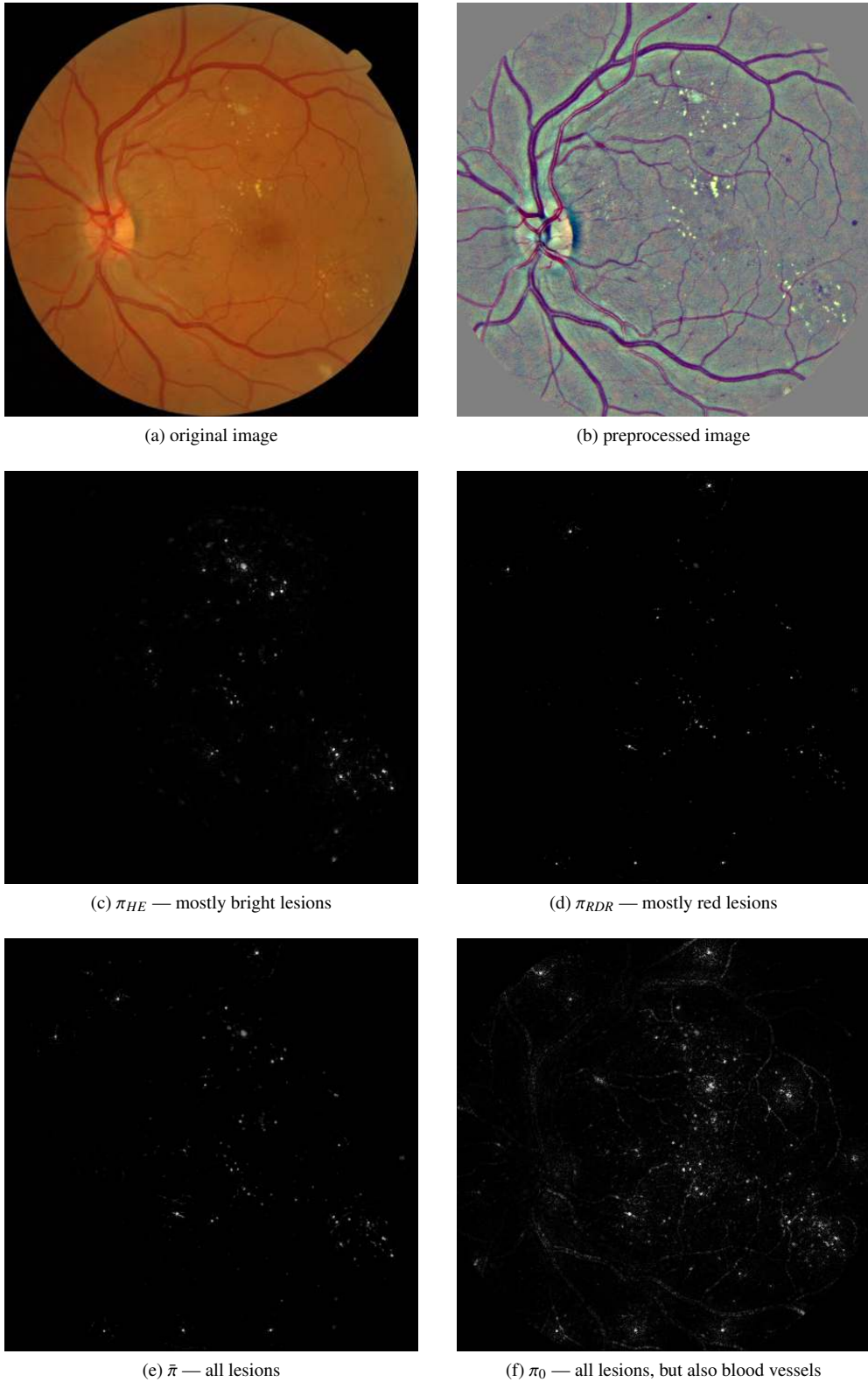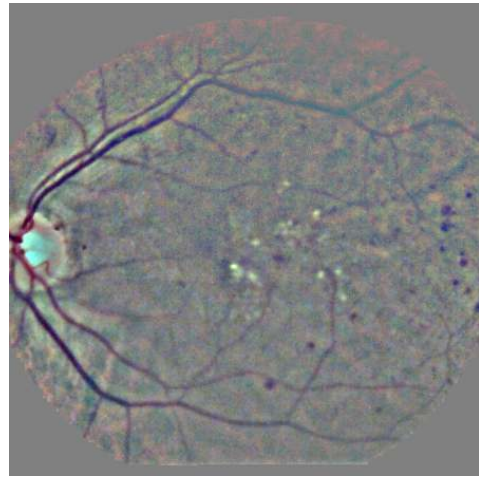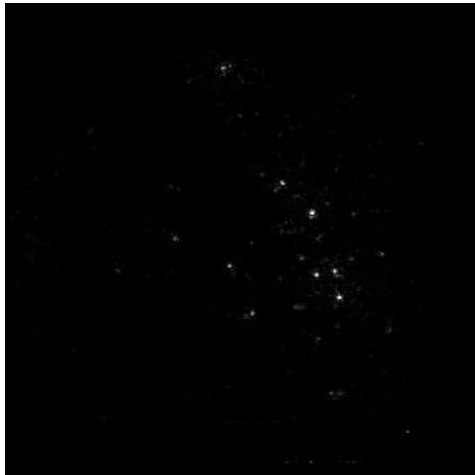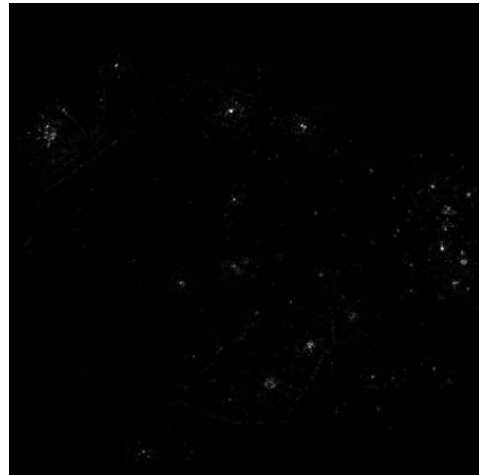
(a) original image

(b) preprocessed image

(c) π_{HE} — mostly bright lesions

(d) π_{RDR} — mostly red lesions

(e) $\bar{\pi}$ — all lesions

(f) π_0 — all lesions, but also blood vessels

Figure 9: Examples of heatmaps from a fundus image of typical quality (image 20051020_57622_0100_PP.png from Messidor). π_0 was obtained without sparsity enhancement. π_{HE} and π_{RDR} were obtained with sparsity enhancement, using networks $\mathcal{N}_{HE}$ and $\mathcal{N}_{RDR}$, respectively. $\bar{\pi}$ was obtained by averaging six π maps (including π_{HE} and π_{RDR}), where each π map is weighted according to the network's importance in the random forest.
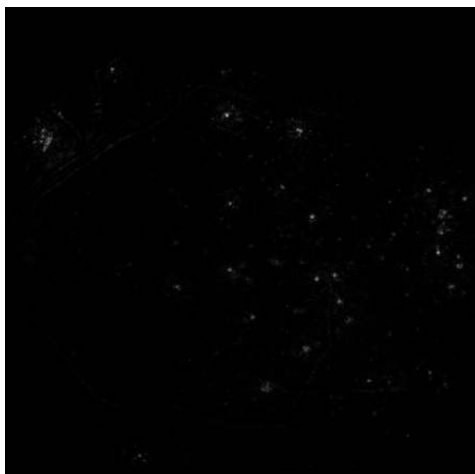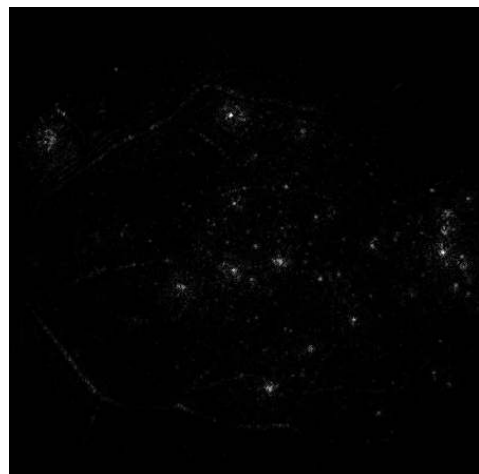
(a) original image

(b) preprocessed image

(c) π_{HE} — mostly bright lesions

(d) π_{RDR} — mostly red lesions

(e) $\bar{\pi}$ — all lesions

(f) π_0 — all lesions, but also blood vessels

Figure 10: Examples of heatmaps from a fundus image acquired with a low-cost handheld retinograph: the Horus DEC 200 (MiiS, Hsinchu, Taiwan). This image was acquired at Brest University Hospital in 2015 (Quellec et al., 2016a).

plains that the proposed solution also outperforms our main baseline solution, namely o_O (A_z = 0.9512). The performance of the ensemble was also very high in e-ophtha-test: A_z = 0.9490 (95 % confidence interval: [0.9459, 0.9521]), as opposed to A_z = 0.8440 for our previous solution, based in part on multiple-instance learning (Quellec et al., 2016c). The proposed ensemble strategy was extended to multiple network structures, but it did not increase performance significantly (see Table 4): because it increases complexity for a very limited benefit, we do not recommend it. It should be noted that Gulshan et al. (2016) recently reported higher performance (up to A_z = 0.991) in smaller datasets (less than 5,000 patients) with a much better ground truth (seven or eight grades per image, as opposed to one in this study); that system does not provide pixel-level information. Abràmoff et al. (2016) also reported higher performance (A_z = 0.980) in a smaller dataset (874 patients) with a better ground truth (three grades per image) for a system relying on pixel-level information for training.

In this study, we focused on detecting referable DR for several reasons. First, this is the most useful classification in the context of DR screening: it determines whether or not the patient needs to be seen in person by an ophthalmologist. Second, the manual segmentations used to evaluate performance at the pixel level do not contain the most advanced signs of DR, such as neovascularizations, so additional manual segmentations would be needed for a relevant evaluation. Third, it allows comparison with the state of the art (Colas et al., 2016). However, unlike most multiple-instance learning algorithms, the proposed solution is not restricted to binary decision problems. The proposed solution is also general in the sense that it could be applied advantageously to all problems governed by a very limited number of relevant pixels. This class of problems was traditionally solved by multiple-instance Learning algorithms, but the proposed solution allows these problems to be solved by ConvNets as well, with enhanced performance. Finally, because the proposed framework is trained to detect relevant patterns in images using image-level labels only for supervision, it can be regarded as a general image mining tool, hence the paper's title. In particular, this framework has the potential to discover new biomarkers in images, which we will try to demonstrate in future works.

The solution presented in this paper for learning to detect referable DR and DR lesions does not require neither expert knowledge nor expert segmentations: it only requires referral decisions stored in examination records. However, expert segmentations (from DiaretDB1) helped us fine-tune the system and improve its performance further. Also, Gulshan et al. (2016) showed that increasing the number of grades per training image, by soliciting multiple experts, significantly improved the performance of their deep learning system. In other words, deep learning solutions will always benefit from clinicians for training, and also of course for assessing their predictions.

In conclusion, a general framework was proposed for solving multiple-instance problems with ConvNets and the relevance of this framework was demonstrated in the context of diabetic retinopathy screening.

References

Abràmoff, M. D., Garvin, M. K., Sonka, M., 2010. Retinal imaging and image analysis. IEEE Rev Biomed Eng 3, 169–208.

Abràmoff, M. D., Lou, Y., Erginay, A., Clarida, W., Amelon, R., Folk, J. C., Niemeijer, M., Oct. 2016. Improved automated detection of diabetic retinopathy on a publicly available dataset through integration of deep learning. Invest Ophthalmol Vis Sci 57 (13), 5200–5206.

Arunkumar, R., Karthigaikumar, P., Feb. 2017. Multi-retinal disease classification by reduced deep learning features. Neural Comput Appl 28 (2), 329–334.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., Samek, W., Jul. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS One 10 (7).

Barriga, E. S., McGrew, E., VanNess, R., Zamora, G., Nemeth, S. C., Bauman, W., Soliz, P., Apr. 2014. Assessing agreement between dilated indirect stereoscopic exam and digital non-mydriatic retinal photography for the evaluation of diabetic retinopathy. In: Proc ARVO. Vol. 55. Orlando, FL, USA, pp. 5335–5335.

Bharali, P., Medhi, J., Nirmala, S., 2015. Detection of hemorrhages in diabetic retinopathy analysis using color fundus images. In: Proc IEEE ReTIS. Kolkata, India, pp. 237–242.

Burlina, P., Freund, D., Joshi, N., Wolfson, Y., Bressler, N., Apr. 2016. Detection of age-related macular degeneration via deep learning. In: Proc ISBI. Prague, Czech Republic, pp. 184–188.

Chen, X., Xu, Y., Wong, D., Wong, T., Liu, J., Aug. 2015a. Glaucoma detection based on deep convolutional neural network. In: Proc IEEE EMBC. Milan, Italy, pp. 715–718.

Chen, X., Xu, Y., Yan, S., Wong, D., Wong, T., Liu, J., Oct. 2015b. Automatic feature learning for glaucoma detection based on deep learning. In: Proc MICCAI. Munich, Germany, pp. 669–677.

Colas, E., Besse, A., Orgogozo, A., Schmauch, B., Meric, N., Besse, E., Oct. 2016. Deep learning approach for diabetic retinopathy screening. In: Froc EVER. Nice, France.

Cuadros, J., Bresnick, G., May 2009. EyePACS: an adaptable telemedicine system for diabetic retinopathy screening. J Diabetes Sci Technol 3 (3), 509–516.

Dai, B., Bu, W., Wang, K., Wu, X., 2016. Fundus lesion detection based on visual attention model. Commun Comput Inform Sci 623, 384–394.

DeLong, E. R., DeLong, D. M., Clarke-Pearson, D. L., Sep. 1988. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. Biometrics 44 (3), 837–845.

Erginay, A., Chabouis, A., Viens-Bitker, C., Robert, N., Lecleire-Collet, A., Massin, P., Jun 2008. OPHDIAT: quality-assurance programme plan and performance of the network. Diabetes Metab 34 (3), 235–42.

Franklin, S., Rajan, S., 2014. Diagnosis of diabetic retinopathy by employing image processing technique to detect exudates in retinal images. IET Image Process 8 (10), 601–609.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proc IEEE CVPR. Washington, DC, USA, pp. 580–587.

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y., Jun. 2013. Maxout networks. In: Proc ICML. Vol. 28. Atlanta, GA, USA, pp. 1319–1327.

Graham, B., Dec. 2014. Fractional max-pooling. Tech. Rep. arXiv:1412.6071 [cs].

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., Kim, R., Raman, R., Nelson, P. C., Mega, J. L., Webster, D. R., Nov. 2016. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. JAMA.

Haloi, M., May 2015. Improved microaneurysm detection using deep neural networks. Tech. Rep. arXiv:1505.04424 [cs].

He, K., Zhang, X., Ren, S., Sun, J., Feb. 2015. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. Tech. Rep. arXiv:1502.01852 [cs.CV].

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. R., Jul. 2012. Improving neural networks by preventing co-adaptation of feature detectors. Tech. Rep. arXiv:1207.0580 [cs].

Kauppi, T., Kalesnykiene, V., Kamarainen, J.-K., Lensu, L., Sorri, I., Raninen, A., Voutilainen, R., Pietilä, J., Kälviäinen, H., Uusitalo, H., 2007. The DI-ARETDB1 diabetic retinopathy database and evaluation protocol. In: Proc BMVC. Warwik, UK.

Kingma, D., Ba, J., May 2015. Adam: a method for stochastic optimization. In: Proc ICLR. San Diego, CA, USA.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. ImageNet classification with deep convolutional neural networks. In: Proc Adv Neural Inform Process Syst. Vol. 25. Granada, Spain, pp. 1097–1105.

Kumar, P., Kumar, R., Sathar, A., Sahasranamam, V., 2014. Automatic detection of red lesions in digital color retinal images. In: Proc IC3I. Mysore, India, pp. 1148–1153.

LeCun, Y., Bengio, Y., Hinton, G., May 2015. Deep learning. Nature 521 (7553), 436–444.

Li, Q., Feng, B., Xie, L., Liang, P., Zhang, H., Wang, T., Jan. 2016. A cross-modality learning approach for vessel segmentation in retinal images. IEEE Trans Med Imaging 35 (1), 109–118.

Lim, G., Cheng, Y., Hsu, W., Lee, M., Nov. 2015. Integrated optic disc and cup segmentation with deep learning. In: Proc ICTAI. Vietri sul Mare, Italy, pp. 162–169.

Maji, D., Santara, A., Ghosh, S., Sheet, D., Mitra, P., Aug. 2015. Deep neural network and random forest hybrid architecture for learning to detect retinal vessels in fundus images. In: Proc IEEE EMBC. Milan, Italy, pp. 3029–3032.

Maji, D., Santara, A., Mitra, P., Sheet, D., Mar. 2016. Ensemble of deep convolutional neural networks for learning to detect retinal vessels in fundus images. Tech. Rep. arXiv:1603.04833 [cs, stat].

Mane, V., Kawadiwale, R., Jadhav, D., 2015. Detection of red lesions in diabetic retinopathy affected fundus images. In: Proc IEEE IACC. Bangalore, India, pp. 56–60.

Manivannan, S., Cobb, C., Burgess, S., Trucco, E., 2017. Sub-category classifiers for multiple-instance learning and its application to retinal nerve fiber layer visibility classification. IEEE Trans Med Imaging (in press).

Melendez, J., van Ginneken, B., Maduskar, P., Philipsen, R. H. H. M., Reither, K., Breuninger, M., Adetifa, I. M. O., Maane, R., Ayles, H., Sánchez, C. I., Jan. 2015. A novel multiple-instance learning-based approach to computer-aided detection of tuberculosis on chest X-rays. IEEE Trans Med Imaging 34 (1), 179–192.

Nesterov, Y., 1983. A method of solving a convex programming problem with convergence rate O(1/sqr(k)). Soviet Math Doklady 27, 372–376.

Nielsen, M. A., 2015. How the backpropagation algorithm works. In: Neural networks and deep learning. Determination Press, Ch. 2.

Prentasic, P., Loncaric, S., Sep. 2015. Detection of exudates in fundus photographs using convolutional neural networks. In: Proc ISPA. Zabreb, Croatia, pp. 188–192.

Quellec, G., Bazin, L., Cazuguel, G., Delafoy, I., Cochener, B., Lamard, M., Apr. 2016a. Suitability of a low-cost, handheld, nonmydriatic retinograph for diabetic retinopathy diagnosis. Transl Vis Sci Technol 5 (2), 16.

Quellec, G., Cazuguel, G., Cochener, B., Lamard, M., 2017. Multiple-instance learning for medical image and video analysis. IEEE Rev Biomed Eng (in press).

Quellec, G., Lamard, M., Cozic, M., Coatrieux, G., Cazuguel, G., Jul. 2016b. Multiple-instance learning for anomaly detection in digital mammography. IEEE Trans Med Imaging 35 (7), 1604–1614.

Quellec, G., Lamard, M., Erginay, A., Chabouis, A., Massin, P., Cochener, B., Cazuguel, G., Apr. 2016c. Automatic detection of referral patients due to retinal pathologies through data mining. Med Image Anal 29, 47–64.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., Fei-Fei, L., Apr. 2015. ImageNet large scale visual recognition challenge. Int J Comput Vis 115 (3), 211–252.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., Müller, K. R., 2016. Evaluating the visualization of what a deep neural network has learned.

IEEE Trans Neural Netw Learn Syst (in press).

Simonyan, K., Vedaldi, A., Zisserman, A., Apr. 2014. Deep inside convolutional networks: visualising image classification models and saliency maps. In: ICLR Workshop. Calgary, Canada.

Srivastava, R., Cheng, J., Wong, D., Liu, J., Apr. 2015. Using deep learning for robustness to parapapillary atrophy in optic disc segmentation. In: Proc ISBI. New York, NY, USA, pp. 768–771.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. J Royal Statist Soc B 58 (1), 267–288.

van Grinsven, M., van Ginneken, B., Hoyng, C., Theelen, T., Sanchez, C., Feb. 2016. Fast convolutional neural network training using selective data sampling: Application to hemorrhage detection in color fundus images. IEEE Trans Med Imaging.

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

Winder, R. J., Morrow, P. J., McRitchie, I. N., Bailie, J. R., Hart, P. M., Dec. 2009. Algorithms for digital image processing in diabetic retinopathy. Comput Med Imaging Graph 33 (8), 608–622.

Worrall, D. E., Wilson, C. M., Brostow, G. J., Oct. 2016. Automated retinopathy of prematurity case detection with convolutional neural networks. In: Proc MICCAI Deep Learning Workshop. Lecture Notes in Computer Science. pp. 68–76.

Yang, N., Lu, H.-C., Fang, G.-L., Yang, G., 2013. An effective framework for automatic segmentation of hard exudates in fundus images. J Circuit Syst Comp 22 (1).

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H., Jul. 2015. Understanding neural networks through deep visualization. In: Proc ICML DL Works. Lille, France.

Zeiler, M. D., Fergus, R., Sep. 2014. Visualizing and understanding convolutional networks. In: Proc ECCV. Zurich, Switzerland, pp. 818–833.

Appendix A. Popular ConvNet Operators

This appendix describes the operators used by the ConvNets evaluated in this paper. Their backward first-order derivatives and their forward second-order derivatives are given in Appendix B.

Appendix A.1. Convolutional Layer (Conv)

Let $w_l \times h_l$ denote the size of the sliding window and let s_l denote its stride, i.e. the offset between two sliding window locations. The value of $D^{(l)}$ in the cth activation map of the nth image, at the (x, y) coordinate, is obtained by a cross-correlation product between $D^{(l-1)}$, inside a sliding window centered on $(s_l x, s_l y)$, and a tensor $\Omega^{(l)}$ of C_l filters with dimensions $w_l \times h_l \times C_{l-1}$ each. A bias $b_{x,y,c}^{(l)}$ is added to the product and a nonlinear activation function a_l is applied to the sum:

$$D_{n,x,y,c}^{(l)} = a_l \left(\sum_{u=-\frac{w_l}{2}}^{w_l-\frac{w_l}{2}} \sum_{v=-\frac{h_l}{2}}^{h_l-\frac{h_l}{2}} \sum_{d=1}^{C_{l-1}} \Omega_{u,v,d,c}^{(l)} D_{n,s_l x+u, s_l y+v, d}^{(l-1)} + b_{x,y,c}^{(l)} \right). \tag{A.1}$$

Biases are generally tied, meaning that $b_{x,y,c}^{(l)} = b_c^{(l)}$, otherwise they are said to be untied.

Appendix A.2. Activation Functions

Currently, the most popular activation functions are rectifiers and leaky rectifiers (He et al., 2015), which are much less computationally intensive than the traditional sigmoid function for instance. They can be expressed as follows:

$$r_\alpha(x) = \max(\alpha x, x), \forall x \in \mathbb{R}, \tag{A.2}$$

where $\alpha \in [0, 1[$ is usually small: $\alpha = 0.01$, typically, or $\alpha = 0$ in the original rectifier function. Leaky rectifiers with large α values (e.g. $\alpha = 0.33$) are called 'very leaky rectifiers'.

Appendix A.3. Dense or Fully-Connected Layers (Dense)

Dense layers are a special case of convolutional layers where $w_l = W_{l-1}$ and $h_l = H_{l-1}$, so the dimensions of $D^{(l)}$ are $N \times 1 \times 1 \times C_l$, where C_l is the number of neurons / filters in layer l.

Appendix A.4. Pooling Layer (MaxPool, MeanPool or RMSPool)

Pooling layers also rely on a $w_l \times h_l$ sliding window with a stride of s_l. Those layers replace the content of the sliding window in the input data tensor by a single value in the output tensor. With a stride greater than 1, a down-sampling operation is performed. Typical pooling operators are the maximum, the mean and the root mean square (RMS). Let $\mathcal{N}_l(x, y) = \{(s_l x + u, s_l y + v)$ s.t. $u = -\frac{w_l}{2}, ..., w_l - \frac{w_l}{2}, v = -\frac{h_l}{2}, ..., h_l - \frac{h_l}{2}\}$ denote the neighborhood of pixel location (x, y) inside $D^{(l-1)}$. A MaxPool layer computes:

$$D_{n,x,y,c}^{(l)} = \max_{(u,v)\in\mathcal{N}_l(x,y)} D_{n,u,v,c}^{(l-1)} \qquad (A.3)$$

and a MeanPool layer computes:

$$D_{n,x,y,c}^{(l)} = \frac{1}{w_l \times h_l} \sum_{(u,v)\in\mathcal{N}_l(x,y)} D_{n,u,v,c}^{(l-1)} \quad . \qquad (A.4)$$

Note that the number of activation maps is unchanged by those operators: $C_l = C_{l-1}$. Finally, an RMSPool layer simply derives from a MeanPool layer as follows:

$$RMSPool\left(D^{(l)}\right) = \sqrt{MeanPool\left(D^{(l)2}\right)} \quad . \qquad (A.5)$$

Appendix A.5. Dropout and Maxout Layers

Dropout is a popular regularization technique (Hinton et al., 2012). During each training iteration, a random selection of filters from layer $l + 1$ (one filter in p) are 'dropped': their input and output connections are temporarily removed. The goal is to train multiple models, where each model is a 'thinned' version of the ConvNet. For improved performance, a Maxout layer can be placed in position $l + 2$: this operator simply returns the maximum output among subsets of p filters from layer $l + 1$ (Goodfellow et al., 2013).

Appendix B. Forward Second-Order Derivatives of Popular ConvNet Operators

Appendix B.1. Forward Second-Order Derivatives for Cross-Correlation

We first discuss the main building-block of ConvNets, namely cross-correlation between data $D^{(l-1)}$ and filter weights $\Omega^{(l)}$, used by convolutional and dense layers [see Equation

(A.1)]. Bias addition, which is trivial, is not discussed and activation is discussed in the following section. Because cross-correlation has two inputs ($D^{(l-1)}$ and $\Omega^{(l)}$), two backward first-order derivative functions need to be computed:

$$\left(\frac{\partial \mathcal{L}_L}{\partial D^{(l-1)}}\right)_{n,x,y,d} = \sum_{u=-\frac{w_l}{2}}^{w_l-\frac{w_l}{2}} \sum_{v=-\frac{h_l}{2}}^{h_l-\frac{h_l}{2}} \sum_{c=1}^{C_l} \Omega_{x,y,d,c}^{(l)} \left(\frac{\partial \mathcal{L}_L}{\partial D^{(l)}}\right)_{n,s_lx-u,s_ly-v,c} , \qquad (B.1)$$

$$\left(\frac{\partial \mathcal{L}_L}{\partial \Omega^{(l)}}\right)_{x,y,d,c} = \sum_{u=-\frac{w_l}{2}}^{w_l-\frac{w_l}{2}} \sum_{v=-\frac{h_l}{2}}^{h_l-\frac{h_l}{2}} \sum_{n=1}^{N} \left(\frac{\partial \mathcal{L}_L}{\partial D^{(l)}}\right)_{n,x,y,c} D_{n,s_lx+u,s_ly+v,d}^{(l-1)} . \qquad (B.2)$$

These equations derive from the chain rule of derivation [see Equation (5)] and the differentiation of a cross-correlation product (Nielsen, 2015). We can see that the cross-correlation between $\Omega^{(l)}$ and $D^{(l-1)}$ in the forward transform [see Equation (A.1)] becomes an actual convolution product between $\Omega^{(l)}$ and $\frac{\partial \mathcal{L}_L}{\partial D^{(l)}}$ in the backward first-order derivative function [see Equation (B.1)]. As expected, one can verify that it becomes a cross-correlation product again, between $\Omega^{(l)}$ and $\frac{\partial \mathcal{L}_0}{\partial D^{(l-1)}}$, in the forward second-order derivative function:

$$\left(\frac{\partial \mathcal{L}_0}{\partial D^{(l)}}\right)_{n,x,y,c} = \sum_{u=-\frac{w_l}{2}}^{w_l-\frac{w_l}{2}} \sum_{v=-\frac{h_l}{2}}^{h_l-\frac{h_l}{2}} \sum_{d=1}^{C_{l-1}} \Omega_{u,v,d,c}^{(l)} \left(\frac{\partial \mathcal{L}_0}{\partial D^{(l-1)}}\right)_{n,s_lx+u,s_ly+v,d} . \qquad (B.3)$$

The gradient of the loss function with respect to $\Omega^{(l)}$ is a cross-correlation product between $\frac{\partial \mathcal{L}_L}{\partial D^{(l)}}$, playing the role of a filter, and the input data [see Equation (B.2)]. One can verify that the same applies during the forward pass, where $\frac{\partial \mathcal{L}_0}{\partial D^{(l-1)}}$ plays the role of the input data:

$$\left(\frac{\partial \mathcal{L}_0}{\partial \Omega^{(l)}}\right)_{x,y,d,c} = \sum_{u=-\frac{w_l}{2}}^{w_l-\frac{w_l}{2}} \sum_{v=-\frac{h_l}{2}}^{h_l-\frac{h_l}{2}} \sum_{n=1}^{N} \left(\frac{\partial \mathcal{L}_L}{\partial D^{(l)}}\right)_{n,x,y,c} \left(\frac{\partial \mathcal{L}_0}{\partial D^{(l-1)}}\right)_{n,s_lx+u,s_ly+v,d}. \qquad (B.4)$$

Appendix B.2. Forward Second-Order Derivatives for the Leaky Rectifier

Let $y = r_\alpha(x) = \max(\alpha x, x)$ with $\alpha \in [0, 1[$ [see Equation (A.2)]. Because r_α simply is a piecewise linear operator, the backward first-order derivatives for r_α are given by:

$$\frac{\partial \mathcal{L}_L}{\partial x} = \begin{cases} 0 & \text{if } \alpha = 0 \text{ and } x < 0 \\ \frac{1}{\alpha}\frac{\partial \mathcal{L}_L}{\partial y} & \text{if } \alpha > 0 \text{ and } x < 0 \\ \frac{\partial \mathcal{L}_L}{\partial y} & \text{if } x \geq 0 \end{cases} \qquad (B.5)$$

and the forward second-order derivatives are given by:

$$\frac{\partial \mathcal{L}_0}{\partial y} = \begin{cases} \alpha\frac{\partial \mathcal{L}_0}{\partial x} & \text{if } x < 0 \\ \frac{\partial \mathcal{L}_0}{\partial x} & \text{if } x \geq 0 \end{cases} \quad . \qquad (B.6)$$

Note that the test for choosing the multiplicative factor (1 or α) is always triggered by the operator's input (x). The same applies to the MaxPool operator below.

*Appendix B.3. Forward Second-Order Derivatives for Max-
Pool*

The backward first-order derivatives for MaxPool [see Equation (A.3)] are given by:

$$\left(\frac{\partial \mathcal{L}_L}{\partial D^{(l-1)}}\right)_{n,x,y,c} = \sum_{\substack{u,v \\ \text{s.t. } D^{(l)}_{n,u,v,c}=D^{(l-1)}_{n,x,y,c}}} \left(\frac{\partial \mathcal{L}_L}{\partial D^{(l)}}\right)_{n,u,v,d} . \qquad (B.7)$$

This means that the errors are backpropagated to the winning neuron inside each sliding window location. One can verify that the forward second-order derivatives are given by:

$$\left(\frac{\partial \mathcal{L}_0}{\partial D^{(l)}}\right)_{n,x,y,c} = \left(\frac{\partial \mathcal{L}_0}{\partial D^{(l-1)}}\right)_{\text{argmax}\left(D^{(l)}_{n,x,y,c}\right)} , \qquad (B.8)$$

where $\text{argmax}\left(D^{(l)}_{n,x,y,c}\right)$ returns the index of the winning neuron inside the corresponding sliding window in $D^{(l-1)}$.

*Appendix B.4. Forward Second-Order Derivatives for Mean-
Pool*

The backward first-order derivatives for MeanPool [see Equation (A.4)] are given by:

$$\left(\frac{\partial \mathcal{L}_L}{\partial D^{(l-1)}}\right)_{n,x,y,c} = \frac{1}{w_l \times h_l} \sum_{\substack{u,v \\ \text{s.t. } (x,y)\in\mathcal{N}_l(u,v)}} \left(\frac{\partial \mathcal{L}_L}{\partial D^{(l)}}\right)_{n,u,v,c} . \qquad (B.9)$$

This means that, during backpropagation, the errors are equally distributed to all neurons inside each sliding window location. The forward second-order derivatives are a special case of Equation (B.3), where W_l is a mean filter; it is given by:

$$\left(\frac{\partial \mathcal{L}_0}{\partial D^{(l)}}\right)_{n,x,y,c} = \frac{1}{w_l \times h_l} \sum_{(u,v)\in\mathcal{N}_l(x,y)} \left(\frac{\partial \mathcal{L}_0}{\partial D^{(l-1)}}\right)_{n,u,v,c} . \qquad (B.10)$$

In other words, the forward second-order derivative function for MeanPool is MeanPool itself.

*Appendix B.5. Forward Second-Order Derivatives for
Dropout and Maxout*

Dropout does not need to be addressed specifically as it simply alters the network temporarily: the above first-order and second-order derivatives are simply computed in the thinned network. As for Maxout, it is addressed similarly to the other maximum-based operators (leaky rectifiers and MaxPool).