Computerized Medical Imaging and Graphics 55 (2017) 28–41

Contents lists available at ScienceDirect

Computerized Medical Imaging and Graphics

journal homepage: www.elsevier.com/locate/compmedimag

Glaucoma detection using entropy sampling and ensemble learning for automatic optic cup and disc segmentation

Julian Zilly[a], Joachim M. Buhmann[b], Dwarikanath Mahapatra[b],*

[a] Department of Mechanical Engineering, ETH Zurich, Switzerland
[b] Department of Computer Science, ETH Zurich, Switzerland

ARTICLE INFO

Article history:
Received 1 February 2016
Received in revised form 16 July 2016
Accepted 29 July 2016

Keywords:
Optic cup
Optic disc
Segmentation
Glaucoma
Ensemble learning
CNN
Boosting

ABSTRACT

We present a novel method to segment retinal images using ensemble learning based convolutional neural network (CNN) architectures. An entropy sampling technique is used to select informative points thus reducing computational complexity while performing superior to uniform sampling. The sampled points are used to design a novel learning framework for convolutional filters based on boosting. Filters are learned in several layers with the output of previous layers serving as the input to the next layer. A softmax logistic classifier is subsequently trained on the output of all learned filters and applied on test images. The output of the classifier is subject to an unsupervised graph cut algorithm followed by a convex hull transformation to obtain the final segmentation. Our proposed algorithm for optic cup and disc segmentation outperforms existing methods on the public DRISHTI-GS data set on several metrics.

1. Introduction

Glaucoma is one of the leading causes of irreversible vision loss in the world, accounting for 12% of such cases. It is estimated that almost 80 million people globally may be affected with glaucoma by 2020 (World Health Organization, 2006). Glaucoma is characterized by damage to the optic nerve through increasing degeneration of the nerve fibers. The progression of the disease is asymptomatic in the early stages but gradually leads to irreversible vision loss. Although there is no known cure, early treatment has been shown to decrease the rate of blindness by around 50% (Michelson et al., 2008). Hence it is essential to have a reliable early detection system for glaucoma onset. This work proposes a computationally efficient method using a ensemble learning based convolutional neural network (CNN) architecture for accurate and robust segmentation of the optic cup (OC) and optic disc (OD) from retinal fundus images. The segmented OC and OD are used to calculate the cup-to-disc ratio (CDR) which is an important indicator of glaucoma progression.

Ophthalmologists use three principal methods to detect onset of glaucoma (Cheng et al., 2013). The first approach is the assessment of increased intraocular pressure inside the eye. However, this is not sensitive enough for early detection and glaucoma can sometimes occur without increased eye pressure. The second approach identifies field of abnormal vision with specialized equipment which makes it unsuitable for a comprehensive screening of glaucoma except in sophisticated medical centers. The third approach is evaluation of damage to the optic nerve. This is most reliable but requires a trained professional, is time-consuming, expensive and highly subjective. Expert assessment may vary depending on experience and training (Cheng et al., 2013). CDR values from segmented optic cup and disc are an important indicator of damage to the optic nerve.

The use of automated diagnostic tools is desirable to minimize subjectivity and make the diagnosis robust and consistent. Color fundus imaging (CFI) has emerged as the preferred procedure for comprehensive large-scale retinal disease screening due to their ease of acquisition and good visibility of retinal structures (Singer et al., 1992). Glaucoma screening methods apply computer algorithms on color fundus retinal images for OD and OC segmentation and calculation of CDR values.

1.1. Prior related work

Automatic CDR measurement involves: (1) optic disc localization and segmentation; and (2) optic cup segmentation. Current state-of-the-art methods for disc segmentation use morphological features (Aquino et al., 2010) and active contours (Joshi et al.,

* Corresponding author.
E-mail address: dwarikanath.mahapatra@inf.ethz.ch (D. Mahapatra).

http://dx.doi.org/10.1016/j.compmedimag.2016.07.012
0895-6111/© 2016 Elsevier Ltd. All rights reserved.

2011). A OC segmentation method is proposed in Joshi et al. (2012) using depth maps computed from relatively displaced sequentially acquired images. Finally, a confidence measure is used to determine the boundary localization. Performance of these methods depends upon initialization and ability to identify weak edges. Chakravarty and Sivaswamy (2014) formulate a Markov Random Field on depth maps extracted from multiple shifted images of the same retina to model the relative depth and discontinuity at the cup boundary. This depth map is subsequently used for optic cup segmentation.

Of late machine learning methods have become popular as they provide a powerful tool for feature classification using learned models. Cheng et al. (2013) formulate a superpixel based classification method to segment the OD and OC. Center surround statistics from the super pixel neighborhood improve performance and a self-assessment reliability score indicates when a given segmentation might be less reliable. Bock et al. (2010) apply glaucoma specific preprocessing (including blood vessel removal), followed by the extraction of different generic features which are compressed using principal component analysis (PCA). A probabilistic two-stage classification scheme then combines these features types into a proposed glaucoma risk index. Mahapatra et al. use a field of experts model (Mahapatra and Buhmann, 2015) and consensus based methods (Mahapatra and Buhmann, 2015; Mahapatra, in press) for segmenting the optic cup and disc. Xu et al. (2014) focus on localizing the optic cup in fundus images and state an unsupervised closed form solution. Their technique estimates optic cup parameters from a code book and estimates the optic cup parameters through a weighted reconstruction based on training images. A prominent limitation of supervised learning methods is the definition of hand crafted features thought to be most relevant for the particular task. Such approaches do not generalize well for different datasets or application domains. Therefore many recent works on segmentation focus on learning the most discriminative features using deep learning and neural networks (Liao et al., 2013). CNNs are a general approach for learning discriminative features from training data in the form of convolutional filters.

Since we use CNNs for OC and OD segmentation we present related work on image segmentation. Mayraz and Hinton (2002) proposed a hierarchical learning procedure based on a probabilistic learning framework called the product of experts (Brown and Hinton, 2000) where the probability of an image is described by the normalized product of learned individual distributions. Kiros and Popuri (2014) use a hierarchical CNN at multiple scales for lung vessel segmentation by optimizing a 2-norm orthogonal matching pursuit problem. Given learned filters, new feature maps are extracted by convolving with the original images, and serve as input to the next layer of filter learning. Ciresan et al. (2012) use a deep neural network (DNN) to segment neuronal structures in electron microscopy (EM) images. Turaga et al. (2009) segment neuronal structures in EM images by learning an affinity graph using a CNN.

1.2. Our contribution

Previous approaches to OC and OD segmentation have used hand crafted features to segment the desired anatomy. However, it is not known whether the hand crafted features are optimal in their performance. As a result such methods do not perform equally well on a wide variety of datasets. An alternative approach is to use CNNs to learn the most discriminative features from the training data. However CNN training requires a large dataset as well as significant computing resources. Our work approaches the problem of learning feature representations from training data from a ensemble learning perspective. Our proposed method is inspired from CNNs with the learned output being a set of filters whose convolutional output provides the optimal representation of the training data. Hence there is no need to define hand crafted features since the CNN architecture learns the optimal representational features through the filters. An ensemble learning approach significantly improves the computational efficiency of training and can be used with limited training data.

The primary research contribution of our work is a hierarchical architecture of CNNs to segment the OC and OD from retinal fundus images. We introduce a novel learning procedure to construct a CNN architecture based on boosting and it shares characteristics with ensemble learning systems. Secondly, an entropy based sampling technique is presented to identify most informative samples from the training dataset and significantly reduce computational complexity. The entropy sampling method is shown to generally yield superior results when compared to uniform sampling. Overall, the proposed method is demonstrated to outperform several other state-of-the-art approaches on a public retinal image data set. Our proposed method differs from conventional CNNs in the following respects: (1) instead of backpropagation we adopt a greedy approach where each stage of filters is learned sequentially using boosting; (2) each stage considers the final classification error to update itself and not the error backpropagated through the next stages; (3) our method operates on patch level data instead of image level data used for traditional CNNs. In summary our proposed method is a ensemble learning system inspired from traditional CNNs and is an effective approach to learn convolutional filters in the absence of large numbers of training data. We describe different components of our method in Sections 2-6, present our results in Section 7, and conclude with Section 8.

2. Convolutional filters and networks

In this section we briefly describe the theory behind convolutional filters and networks. Hierarchical layers of convolutional filters that mimic the effects of visual receptive fields were inspired by Hubel and Wiesel's work on feedforward processing in the early visual cortex of cats (Hubel and Wiesel, 1963). Inspired by this CNNs use local spatial correlations in images and also exhibit robustness to natural transformations such as changes of viewpoint or scale.

2.1. Convolutional filtering

Convolution between functions *f* and *g* can be written as

$$y(t) = f(t) * g(t) = \int_{-\infty}^{\infty} f(t-\tau) * g(\tau) d\tau \qquad (1)$$

where * denotes the convolution operation. The equivalent discrete formulation is

$$y[i, j] = (I * K)[i, j] = \sum_m \sum_n I[m, n]K[i-m, j-n] \qquad (2)$$

where *K* is a two-dimensional matrix called kernel and *I* is a two-dimensional (gray-valued) image. Convolution can be extended to higher dimensional images and kernels. Generally kernels tend to be considerably smaller than the images.

Convolutional networks exploit three key ideas (Bengio et al., 2015): sparse interactions, parameter sharing and equivariant representations. Using convolutional filters as building blocks, complex convolutional networks can be constructed. Some of the important building blocks of CNNs are:

- **Convolutional filter**: Also called kernel, it is the fundamental building block of a CNN. Each filter is convolved with each point of an assigned input image and generates an output. Filters are generally of size $n_1 \times n_1 \times n_{maps}$ where n_1 is the specified filter size and n_{maps} describes the dimension of the input image, i.e., $n_{maps} = 3$ for an RGB image.

- **Max pooling layer**: Max pooling is a non-linear downsampling operation where the input image is partitioned into a set of rectangular patches and the maximum of each such patch is returned as the output.

$$a_j = \max_{N_2 \times N_2} (a_i^{n \times n} u(n, n)) \tag{3}$$

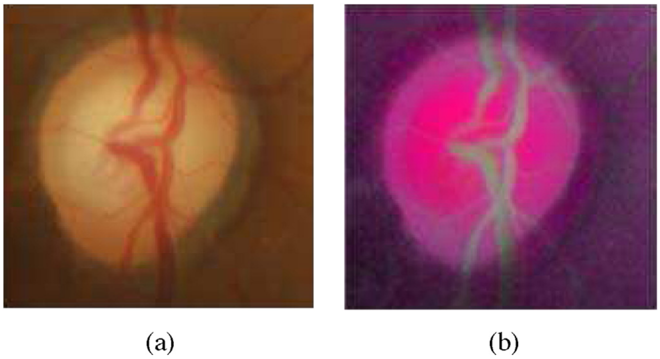
By returning the maximum for a given $N_2 \times N_2$ patch, max pooling introduces a considerable amount of robustness into the CNN since several configurations of maximal values in a given patch will yield the same output.

CNNs tend to be wide (many filters per layer) and deep (many layers). For a large scale system this requires training millions of parameters. Traditionally, this training is done using the backpropagation algorithm as explained in Bengio et al. (2015). Large number of parameters need to be trained on a large dataset to avoid overfitting. However we follow a *different approach*. With our relatively smaller dataset fewer parameters can be reliably trained by our method using entropy sampling and boosting such that the learned patterns generalize better to unseen data.

3. Preprocessing

We employ a two stage approach where the optic disc is first segmented followed by the smaller optic cup. Contrary to other works such as (Bock et al., 2010) we apply a domain independent preprocessing step to enhance the information content of the images. The optic disc is first localized by applying a circular Hough transform on the green channel image. Each image is first cropped so that the optic disc or cup is relatively central to the cropped image and a certain amount of "background" around the optic disc is retained. This allows the training procedure to capture the essential characteristics of the image while being able to focus more on the region of interest. Cropping also reduces the computational burden.

Since the retinal fundus images are in RGB color space they are converted to $L^* a^* b$ color space using a nonlinear transformation which mimics the nonlinear perceptive response of the eye. Empirical evidence suggests that $L^* a^* b$ yields better results compared to other color spaces (Wang et al., 2014). The mean image intensity is subtracted from all pixel intensities followed by division with the standard deviation. The intensities are scaled to lie in $[0, 1]$. All color channels are normalized individually. Fig. 1 shows results of this pre-processing step. Clearly, preprocessing enhances the contrast between optic disc or cup region with respect to background.



(a)

(b)

Fig. 1. (a) Image before pre-processing; and (b) image after preprocessing.

4. Entropy sampling

Analyzing every pixel in the cropped images constitutes a significant computational burden. Additionally, information at the pixel level is highly redundant since neighboring pixels tend to give highly correlated information. This problem is addressed using an entropy based sampling scheme to select the most informative pixels from the image. Uniform sampling with equal probability passes up the opportunity to extract relevant information for the algorithm. In the worst case, the sampled points cover the image but fail to provide a comprehensive account of where "interesting" patterns are present. Subsequently, nonuniform sampling approaches have been used in other applications (Ciresan et al., 2012). Since the proposed method is based on convolutional filters, it is important to select points with informative surroundings. Otherwise, local patches around selected points will not yield sufficiently discriminative information.

A first order entropy estimate for a given point can be calculated by recording each gray value in a neighborhood N using a histogram with 256 bins. With this probability estimate, the entropy is calculated as

$$H(x) = \sum_{x_i \in N_3} - p(x_i) \cdot \log(p(x_i)) \tag{4}$$

where N_3 is the neighborhood of pixel x_i.

The entropy map quantifies the informativeness for each pixel. However entropy maps over any single color channel are noisy as there are many pixels with high entropy, thus defeating the purpose of identifying informative points. Hence we calculate an additional quantity which we term as "total entropy". It is defined as

$$H_{total}(x) := \left(\sum_{l=1}^{n_{maps}} H_l(x) \right)^2 . \tag{5}$$

It is essentially the sum of squares of the individual maps. The square increases the difference between high and low entropy points, thus suppressing noise. For entropy sampling, points are sampled without replacement to ensure a greater coverage of different points. Fig. 2(a) shows the final ("total entropy") map of Fig. 1(b) using a $N = 7 \times 7$ neighborhood. A 7 × 7 neighborhood was used because it gives the best tradeoff between computational complexity and accuracy (see Section 7.4).

Fig. 2(b) and (c) demonstrates the difference between entropy sampling and uniform sampling with 1000 sampled points. Entropy sampled points (Fig. 2(b)) tend to lie much more frequently at edge points and at the border of the optic disc and cup. This makes it more likely for an algorithm to learn discriminative features between the optic disc and cup, and the background. However for uniform sampling (Fig. 2(c)) there are many points that lie on regions that are not particularly informative.

An added advantage of using entropy sampling is that it identifies informative points on various essential landmarks such as edges, blood vessels, etc., which provide discriminative information. This is advantageous when we aim to learn filters from a small dataset, where it is desirable to extract as much information as possible from the limited number of data samples.

5. Convolutional feature learning

The proposed method exploits the fact that convolutional networks are essentially an ensemble learning technique. Many of the characteristics of classical ensemble learning approaches also extend to CNNs. Convolutional networks are composed of individual convolutional filters that can be regarded as classifiers which together form an ensemble. Generally, CNNs have filters that are

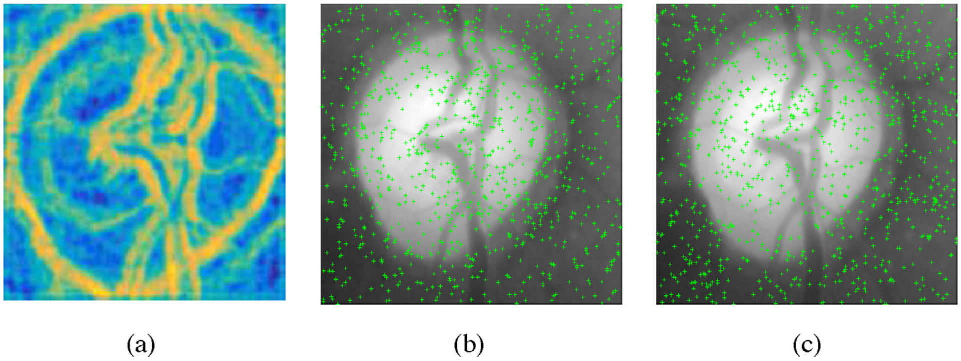


Fig. 2. (a) Output of "total entropy" map using $N = 7 \times 7$; (b) entropy sampled points; and (c) uniformly sampled points.

initialized to small random values and are incrementally adapted to a desired state using backpropagation (Cireşan et al., 2011). The random initialization ensures sufficient initial diversity in the filters. We use boosting to learn the diverse filters successively in a supervised fashion, each trying to minimize a weighted classification error.

5.1. Filter learning

3×3 patches are extracted around each sampled point. This patch size gives the best tradeoff between computational complexity and generalization ability (Section 7.4). The number of parameters increases with the square of the patch width. Hence more data is required to reliably estimate more parameters. Experiments with 5×5 or 7×7 patches do not show improved segmentation accuracy and do not generalize better than the smaller 3×3 convolutional filters. Additionally, convolutional filters are learned on five different scales. The same filter size is learned on images and their downsampled versions. The scaling of different images is illustrated in Fig. 3 to provide a better understanding of the coverage of the filters for different scales. Each scaled image of Fig. 3 has a central green square of size 3×3 to illustrate the size of the convolutional filters. Evidently, a filter of size 3×3 will capture different levels of image information depending on the scale. Furthermore, scaling the image instead of using larger and larger filters enables a smaller filter to coarsely emulate larger filters.

To ensure that the filters do not learn redundant patterns, such as similar patterns with different magnitude, all patches are subjected to *Local Contrast Normalization* as in Kiros and Popuri (2014). Each patch is reshaped into a vector x_{patch}, divided by its L_2 norm and its mean value subtracted. That is,

$$x_{patch} \leftarrow \frac{x_{patch}}{\|x_{patch}\|} \tag{6}$$

$$x_{patch} \leftarrow x_{patch} - \mu_{patch} \tag{7}$$

where μ_{patch} denotes the mean of the normalized patch vector and $\|x_{patch}\|$ is the ℓ_2 norm of the original patch vector. The convolutional filters are learned from these normalized patches using boosting. This is the core novelty of our proposed filter learning algorithm. To accomplish this, the following weighted ℓ_1-norm optimization problem is solved for each filter individually

$$\underset{w,b}{\text{minimize}} \quad \sum_{i=1}^{N} \upsilon_i \cdot |y_i - x_i^T w - b| + \lambda ||w||_1 \tag{8}$$

where $y_i \in Y = \{-1, +1\}$ is the binary label of a given point i, $x_i \in X_{patch}$ represents the corresponding patch around point i in vector form. w is the convolutional filter in vector form that is to be learned, b is a learned constant offset, $|\cdot|$ is the vector dot product, and υ_i are the positive weights on an individual data point. The optimization environment CVX (Grant and Boyd, 2008, 2014), a package for specifying and solving convex programs, was used to determine the final parameter values.

The above equation can be re-written as

$$\underset{w,b}{\text{minimize}} \quad \sum_{i=1}^{N} \upsilon_i \cdot |y_i - \mathbf{x}_i^T \mathbf{w}| + \lambda ||\mathbf{w}||_1 \tag{9}$$

$\mathbf{w}$ and $\mathbf{x}$ include offset b and an extra 1, respectively, in the column vector. The ℓ_1-norm constraint is imposed since the labels are -1 or $+1$. We are interested in ensuring that each individual filter tries to solve the segmentation problem whose label is the sign of the expression in Eq. (8). The algorithm does not focus on finding a way to minimize large square errors and hence the ℓ_2-norm would not serve the purpose. The ℓ_1-norm imposes lower penalty to deviations from the actual labels $\{-1, +1\}$.

In contrast to Kiros and Popuri (2014), consecutively learned filters are not required to be orthogonal to each other. Rather "exploration" of different filters is done through reweighting of data points based on *Gentle AdaBoost* as described in Doğan and Akay (2010). *Gentle AdaBoost* is a version of *AdaBoost* (Freund and Schapire, 1999) which places less weight on outlier points. This is meant to generalize better by avoiding overfitting. In the context of imperfect labels and preliminary tests with the proposed algorithm, *Gentle AdaBoost* seems to indeed generalize better than the standard *AdaBoost approach*. Henceforth boosting refers to filters learned using *Gentle AdaBoost*.

Gentle AdaBoost is used in the following manner. Initialize the weights as $\upsilon_i = \frac{1}{m}$, for $i = 1, \ldots, m$. For $n = 1, \ldots N$:

1. Estimate the "weak" hypothesis $h_n(x)$, i.e. learn filter w and bias b in the optimization problem Eq. (8).
2. Update weights

$$\upsilon_i \leftarrow \frac{\upsilon_i \cdot \exp(-y_i h_n(\mathrm{x}_i))}{Z_n} \tag{10}$$

with Z_n chosen so that $\sum_{i=1}^{m} \upsilon_i = 1$.

The difference with respect to standard *AdaBoost* lies in the fact that the term in the exponential of Eq. (10) has a weighting factor $\alpha = \frac{1}{2} \log \frac{1-\epsilon}{\epsilon}$ when using *AdaBoost*. For *Gentle AdaBoost* the factor α is always set to 1. The factor α is determined by the error ϵ of the individual classifier, where a highly accurate classifier yields a high α and an inaccurate classifier possesses low α.

In addition to *Gentle AdaBoost* reweighting, samples of each class are reweighted such that each class is equally weighted, i.e. the

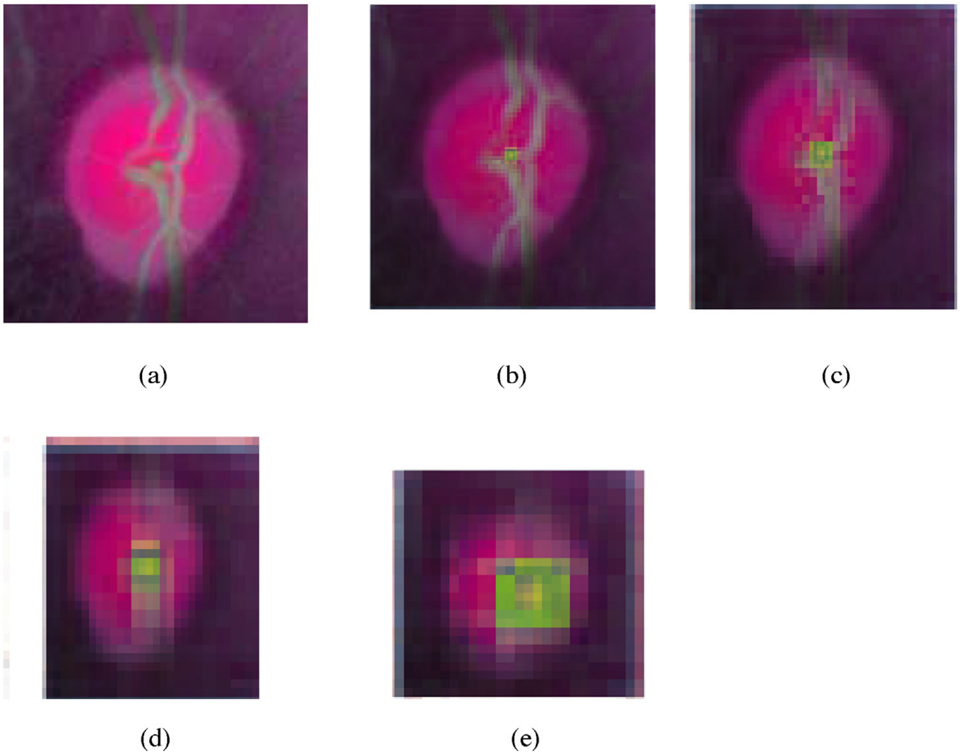


Fig. 3. (a) Scale 1; (b) scale 2; (c) scale 3; (d) scale 4; and (e) scale 5. The green square illustrates the coverage of the same 3 × 3 filter on images of different scale. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

sum of the weights of samples of each class is equal. This has been empirically shown to improve exploration of different convolutional filters. Sample learned filters are shown in Fig. 4 where the first row shows filters learned for the optic disc and second row shows filters learned for optic cup segmentation. The two first filters of each scale are presented. Each of the filters are scaled to lie in the range of [0, 1] to demonstrate the pattern of weights in more comparable setting.

5.2. Convolutional network architecture

Fig. 5 illustrates the architecture of our proposed CNN. Some of its salient points are:

- The proposed network has two fully connected layers.
- Filters are learned for 5 scales in the first layer and 4 scales in the second layer.
- The input to the second layer is the processed and max-pooled output of the first layer.
- The second layer has access to all the processed output of the first layer which makes the CNN fully connected.
- 6 filters are learned for each scale in the first layer giving a total of 6 × 5 =30 filters in the first layer.
- 1 filter is learned for each scale in the second layer giving 4 filters.
- In total 30 + 4 =34 convolutional filters are learned.

Convolutional filters are learned for each scale individually, implying that weights are reset for each scale and boosting is only applied within the same scale. For images at different scales, sampled points at one scale do not necessarily offer the same kind of information as points sampled at another scale. Furthermore, points sampled at a certain scale need not correspond to points sampled at the same position at a downsampled scale. The “original” point does not generally exist in the downsampled image. Consequently, new points are sampled for each scale. Filters are

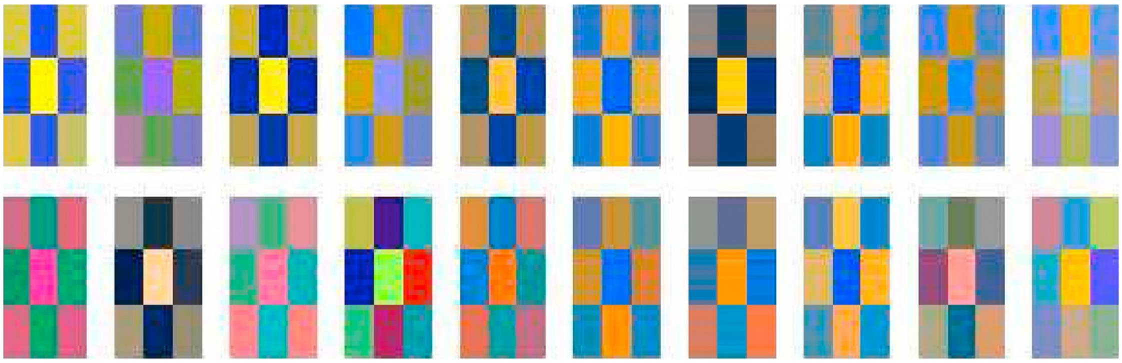


Fig. 4. Illustration of the learned convolutional filters for optic disc and cup.

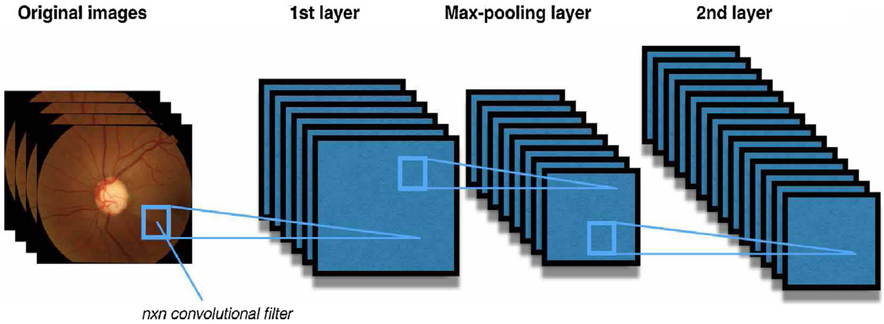


Fig. 5. Illustration of the implemented convolutional network architecture.

learned through optimization and reweighting as discussed in Section 5.1. The weighting factors are reset for each scale, since different points are sampled.

We observe that the same amount of coverage is achieved with far fewer sampled points since the total amount of available points has decreased. As a heuristic, the number of sampled points in a scale is set to

$$N_{samples} = \frac{500}{2^{(\#scale-1)}} \tag{11}$$

where $\#scale$ refers to the number of the scale. The original scale has number one, the next scale of half the width and height has scale number two and subsequent scales are numbered accordingly. This enables the algorithm to learn smaller downsampled scales with higher scale number faster since less data points are sampled while at the same time having more coverage compared to the original image scale. Each learned convolutional filter is later convolved with each input image at the respective scales. The standard discrete convolution operation applied by each filter is

$$y[i, j] = (I * K)[i, j] = \sum_m \sum_n I[m, n]K[i - m, j - n] \tag{12}$$

where K is the kernel representation of filter w_i of Eq. (8) and I is the $m \times n \times n_{maps}$ dimensional image. Here y denotes the output of the convolution.

The convolved images are then postprocessed to standardize them. Since these will later serve as input to the second convolutional layer, greater structured data is highly desirable. The output of the convolution of each filter is passed through a hyperbolic tangent saturation function as it gave the best accuracy in comparison to Relu and sigmoid functions. As previously done for preprocessing, the mean of the image is subtracted and all values divided by the standard deviation. The intensity values are rescaled to lie in the range [0, 1].

$$X_{patch-layer2} \leftarrow \tanh(X_{patch-layer1}w + b)$$

$$X_{patch-layer2} \leftarrow X_{patch-layer2} - \mu_{patch-layer2}$$

$$X_{patch-layer2} \leftarrow \frac{X_{patch-layer2}}{\sigma_{patch-layer2}}$$

where $\mu_{patch-layer2}$ is the column-wise mean and $\sigma_{patch-layer2}$ the column-wise standard deviation of patch matrix $X_{patch-layer2}$ of dimension $n_4 \times k$ with n_4 the number of data points and k is the dimension of an individual patch.

The output of convolving filters with the input images highlights the image characteristics learned by the filter. Fig. 6 shows the processed output of the first convolutional filters at each scale for the sample image shown in Fig. 1(a). The output of the filters becomes increasingly coarse and at the smaller scales the output closely resembles the optic disc we aim to segment, although their edges are blurred. Each filter focuses on different aspects of a given image. The combination of these diverse features or viewpoints is expected to offer better discriminative features than individual features. Images at lower scales are later upsampled to the original scale while convolving the learned filters.

The convolved images are passed through a max-pooling operation of dimension 2 × 2 to introduce further robustness into the system (Scherer et al., 2010). The max-pooling operation takes the maximum of all values within each 2 × 2 patch and discards all other values. Thus after max-pooling, the size of all images is halved along each dimension. The robustness of the max-pooling operation stems from the fact that slight shifts in the input data do not change the output of the max-pooling operation as explained in Section 2.

In the next step postprocessed and max-pooled convolved images are stacked on top of each other resulting in images of

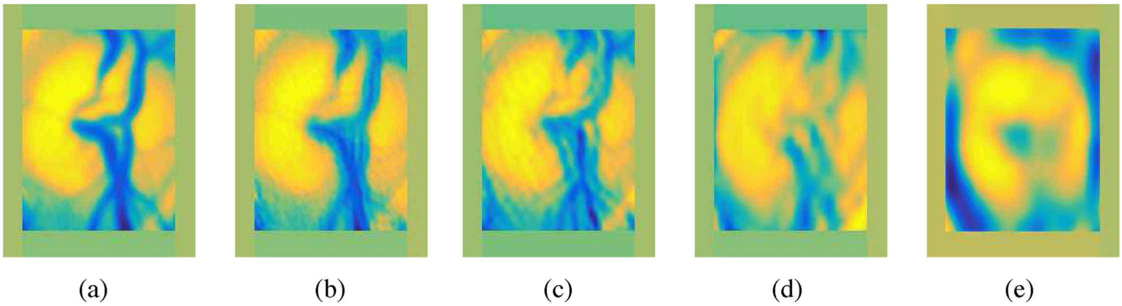


Fig. 6. Illustration of sample image convolved with the first learned filters of each scale (total of 5 scales): (a) filter 1 at scale 1; (b) filter 1 at scale 2; (c) filter 1 at scale 3; (d) filter 1 at scale 4; (e) filter 1 at scale 5.

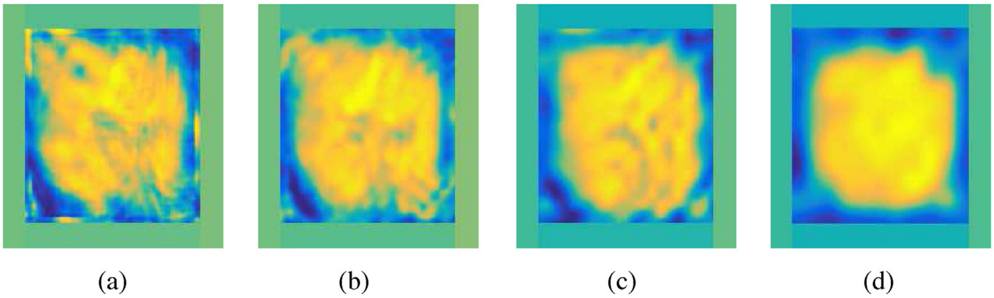


Fig. 7. Illustration of the second layer output for sample image with one filter learned per scale: (a) filter learned at scale 1; (b) filter learned at scale 2; (c) filter learned at scale 3; (d) filter learned at scale 4.

size $m \times n \times n_{maps}$, where n_{maps} is equal to the number of filters learned in the first layer. These images are the input images to the second layer of the convolutional network. In the second layer, filters are learned for four different scales. The learning procedure of the optimization problem in Eq. (8) is applied and the new set of learned filters combine the output of the previously learned filters in the first layer. In boosting (Freund and Schapire, 1999), individual classifiers are generally combined by a weighted average of the form

$$C(x) = \sum_{i=1}^{K_2} \alpha_i c_i(x). \tag{13}$$

$C(x)$ denotes the ensemble classifier, K_2 is the total number of individual classifiers and α_i is the accuracy based weighting of the individual classifier $c_i(x)$. The bagging approach of (Breiman, 1996) differs from conventional bagging in that all α_i equal 1. Bagging averages the individual classifiers while boosting performs a weighted average. The filters in the second layer exhibit similar characteristics as the ensemble classifier in Eq. (13). These can be written as

$$C_{conv}(x) = \sum_{i=1}^{K} \sum_{p \in patch} w_{i,p} c_i(p) \tag{14}$$

where $C_{conv}(x)$ is a filter of the second layer (or any further layers), p denotes the positions of points in the patch around point x, $w_{i,p}$ are the learned weights for the convolutional filter in layer two and $c_i(p)$ describes the processed output of convolutional filter i of the previous layer.

Filters in the second layer act as an extended ensemble classifier as it considers not only output of individual classifiers (convolutional filters of a previous layer) but also spatial information around each point. This approach constructs an hierarchical ensemble learning framework using convolutional filters. Similar to the first layer, the output of all filters of the second layer on the sample image is illustrated in Fig. 7.

In summary, the following specifications for the convolutional network were used:

- Filters of size $3 \times 3 \times n_{maps}$ are trained, where $n_{maps} = 3$ corresponds to the number of color channels of each input image.
- Filters are learned on five scales in the first layer and four scales in the second layer. This is meant to give the local algorithm (3×3 filters) greater global coverage. Multiple scales indicate that the first set of filters are learned on the original image followed by their downsampled versions.
- Filters are learned using reweighting by applying *Gentle AdaBoost*.
- The convolutional filters are learned in a hierarchical manner.
- A max-pooling operation is performed on the processed output of the first layer, which serves as input to the second layer.
- A total of 6 filters per scale for 5 scales are trained in the first layer. 1 filter per scale for 4 scales are trained in the second layer.
- All configurations are used for optic cup and disc equally.

6. Obtaining the final segmentation

After learning the filters, the training images are convolved with each of them to produce a set of 34 maps (equal to the number of learned filters). Note that there are two sets of 34 filters corresponding to optic cup and optic disc. 2000 points each belonging to optic cup and disc are sampled from *each image*, and the values from the convolved images are used as features. Additionally, the L^*a^*b color values of each sampled point are also included as a feature, giving a total of 37 features.

The features are used to train a softmax logistic regression classifier as in Kiros and Popuri (2014). The softmax classifier calculates the probability of a given point belonging to the optic disc, cup or the background. This is written in the form

$$h_\theta(\mathbf{x}) = \begin{pmatrix} P(y=1|\mathbf{x};\theta) \\ P(y=2|\mathbf{x};\theta) \\ \vdots \\ P(y=K|\mathbf{x};\theta) \end{pmatrix} = \frac{1}{\sum_{i=1}^{K} \exp(\theta^{(i)T}\mathbf{x})} \begin{pmatrix} \exp(\theta^{(1)T}\mathbf{x}) \\ \exp(\theta^{(2)T}\mathbf{x}) \\ \vdots \\ \exp(\theta^{(K)T}\mathbf{x}) \end{pmatrix} \tag{15}$$

where $\theta^i \in \mathbb{R}^n$ are the learned parameters for class i, $\mathbf{x}$ is the feature vector supplied to the classifier and $y \in Y = \{1, \ldots, K\}$ is the label corresponding to a given feature vector. The fraction $\frac{1}{\sum_{i=1}^{K} \exp(\theta^{(i)T}\mathbf{x})}$ ensures that all values sum to 1, giving a probability distribution over all classes. The softmax regressor outputs probability maps such as the one displayed in Fig. 8(a).

A unsupervised graph cut algorithm (Salah et al., 2011) is then applied to the probability map to obtain an initial segmentation of the disc as shown in Fig. 8(b). Image segmentation using graph cuts is a label assignment problem where each pixel is assigned a label l. Graph cut optimization finds a set of labels that minimizes the following energy function:

$$F(\lambda) = D(\lambda) + \alpha R(\lambda) \tag{16}$$

where $D(\lambda)$ is the data dependent term, $R(\lambda)$ is the regularization or smoothing term and α determines their relative contributions. The applied method finds the average intensities of different segments using a k-means approach. Then the difference of each pixel from these average values is subjected to a radial-basis function (rbf) kernel mapping. Eq. (16) can be written as

$$F(\{\mu\}, \lambda) = \sum_{l \in L} \sum_{p \in R_l} (\phi(\mu_l) - \phi(I_p))^2 + \alpha \sum_{\{p,q\} \in C} r(\lambda(p), \lambda(q)) \tag{17}$$

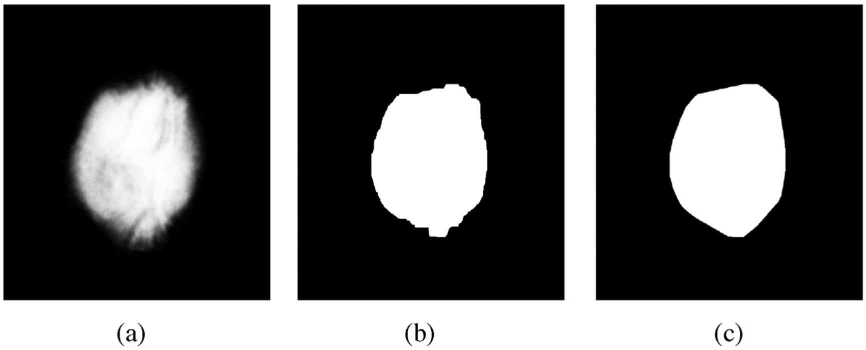


Fig. 8. (a) Output of softmax classifier for sample image; (b) graph cut partitioning of probability classification map for sample image; (c) final segmentation of sample image after applying a convex hull transformation on graph cut output.

where ϕ is the non-linear kernel transformation and r is the regularization term. Minimizing the energy term in Eq. (17) using graph cuts (Boykov and Veksler, 2001) yields the desired unsupervised segmentation.

A convex hull transform is applied to the graph cut segmentation output. Given the oval shape of both the optic disc and cup it is apriori known that the desired shape is convex. Taking the convex hull of the output of the graph cut algorithm can combine previously disjoint regions into the optic disc or cup. Thus a better segmentation is achieved and the final segmentation is illustrated in Fig. 8(c).

7. Experiments and results

7.1. The DRISHTI-GS data set

Our method is validated on the DRISHTI-GS dataset (Sivaswamy et al., 2014) which consists of 50 patient images obtained using 30 degree FOV at a resolution of 2896 × 1944. We use a 5 fold cross validation scheme with 40 training images and 10 test images in each fold. The ground truth disc and cup segmentation masks were obtained by a majority voting of manual markings by 4 ophthalmologists. Quantitative evaluation is based on F-score ($F = 2P \times R/(P + R)$) to measure the extent of region overlap and absolute pointwise localization error B in pixels (measured in the radial direction); P is precision and R is recall. Additionally we report the overlap measure $S = Area(M \cap A)/Area(M \cup A)$. M is the manual segmentation while A is the algorithm segmentation.

To generate the “ground-truth” images four human experts segment the optic disc and cup in each image resulting in a softmap for the segmentation with values denoting the fraction of annotators who agree. These values vary from 0 to 1 in 0.25 increments, where zero corresponds to no experts segmenting a given pixel as cup or disc and 1 corresponds to all experts labeling the region as optic up or disc. A value of zero corresponds to background, while a value of 1 corresponds to the optic disc or cup. To further illustrate this point, the ground truth softmaps for optic disc and cup segmentation are displayed in Fig. 9.

As is obvious from the figures, all experts do not always agree on the exact position of a segment, especially for optic cup segmentation. There is no “perfect” ground truth available. Only a “gold standard” is used which is defined to be the agreement of at least three of the four experts as specified by the authors of the data set (Sivaswamy et al., 2014).

7.2. Error measures

Similar to the error measures in Sivaswamy et al. (2014), precision and recall values are calculated for each segmentation as follows

$$\text{Precision} = \frac{tp}{tp + fp}, \quad \text{Recall} = \frac{tp}{tp + fn} \tag{18}$$

where tp, count of true positive; fp, count of false positive and fn, count of false negative pixels. The F-score (F) is computed as the harmonic mean of precision and recall defined as:

$$F = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{19}$$

F-score values are in the range of [0, 1] with higher values indicating better performance.

Boundary localization error: The distance between computed region boundary and ground truth is a metric for boundary localization. This error measurement is meant to indicate how well a given algorithm can find the object boundary when compared to the actual object boundary. Let C_g and C_o be the ground truth and

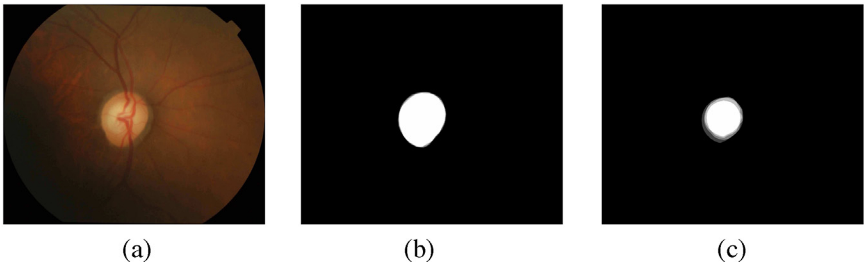


Fig. 9. Illustration of softmap segmentations. (a) Original image from the DRISHTI-GS data set (Sivaswamy et al., 2014); (b) ground truth of optic disc; (c) ground truth of optic cup.

Table 1
Comparison of *F*-score for different sizes of entropy filter, and its comparison with uniform sampling.

Uniform	3 × 3	5 × 5	7 × 7	9 × 9	11 × 11
0.82 ± 0.10	0.83 ± 0.11	0.84 ± 0.09	0.87 ± 0.1	0.85 ± 0.07	0.82 ± 0.13

computed boundary by a method, respectively. The distance (*D*, in pixels) between two curves is then defined as:

$$D = \frac{1}{N}\sum_{i=1}^{N-1}\sqrt{|(d_g^n) - (d_o^n)|^2} \tag{20}$$

where d_g^n and d_o^n are the distance from disk center to points on C_g and C_o, respectively in the angular direction indexed by $n \cdot N$ is set to 24 in this work to be comparable to Sivaswamy et al. (2014).

Cup-to-disc ratio (CDR): After segmentation of optic disc and optic cup, the cup-to-disc ratio (CDR) is computed as the ratio of the maximal diameter of the optic cup to the maximal diameter of the optic disc

$$\text{CDR} = \frac{\text{Cup diameter}}{\text{Disc diameter}}. \tag{21}$$

To compare different methods their mean error of the CDR estimation and the standard deviation of CDR estimation errors are evaluated.

7.3. Entropy sampling

We analyze the effects of different neighborhood sizes on the output of entropy sampling. For different sizes of the entropy filter we calculate the *F*-score for optic cup segmentation, which is more challenging than disc segmentation. The results are shown in Table 1. All results were obtained using 5-fold cross-validation. Since there is the chance that different points may be selected for entropy and uniform sampling, we perform 10 runs of the entire pipeline in every fold to reduce any possible bias.

For some window sizes entropy sampling performs better than uniform sampling. For very small neighborhoods (3 × 3, 5 × 5) the entropy map assigns high weights to many points and is noisy as a small neighborhood finds it difficult to identify informative samples. In contrast, larger neighborhoods are more robust in identifying informative points. However too big neighborhoods (11 × 11) include a lot of superficial information from uninformative pixels. Fig. 10 illustrates the results of entropy filtering using different neighborhoods. Since a 7 × 7 neighborhood gives the best results we use it for our experiments.

7.4. Kernel size

We analyze the effects of different kernel sizes on the segmentation accuracy in terms of *F*-score for the optic cup. Table 2 summarizes the results obtained using 5-fold cross-validation. The best results are obtained for 3 × 3 patches which is the motivation for using this patch size during learning.

Table 2
Comparison of *F*-score for different kernel sizes used in optic cup segmentation.

3 × 3	5 × 5	7 × 7	9 × 9	11 × 11
0.87 ± 0.18	0.85 ± 0.14	0.84 ± 0.11	0.84 ± 0.15	0.82 ± 0.23

Table 3
Comparison of performance of bagging and boosting.

F-score	0.830 ± 0.12	0.86 ± 0.09
Precision	0.81 ± 0.20	0.84 ± 0.1
Recall	0.88 ± 0.1	0.92 ± 0.07
Boundary loc. (px)	18.05 ± 13.62	16.53 ± 11.807

7.5. Effect of boosting vs. bagging

In bagging variability is achieved by providing each individual classifier with a random subset of the training data. Boosting uses a more direct approach where previously incorrectly classified points receive higher weight than correctly classified samples in learning individual classifiers. Finding diverse and accurate classifiers is essential for creating an accurate ensemble classifier. Table 3 highlights the difference in performance between bagging and boosting using 5-fold cross-validation and entropy sampling with a 7 × 7 neighborhood. Bagging in this case amounts to providing each convolutional filter with a different random subset of data points to learn from, where consecutively learned filters are required to be orthogonal to each other. Boosting on the other hand uses a reweighting policy based on *Gentle AdaBoost* to arrive at accurate yet distinct convolutional filters.

A closer look at the learning framework reveals that the most discriminative convolutional filter in each scale is the first one. Subsequently learned filters do not contribute nearly as much discriminative information. The difference in performance of bagging and boosting is that filters learned through bagging tend to focus on very similar aspects of an image whereas boosting allows consecutively learned filters to explore very different facets of the images. The sampling of different subsets in bagging is not able to provide the same variability as reweighting does for boosting. This difference in information gives boosting a significant advantage over bagging.

7.6. Optic disc and cup segmentation

The same algorithmic setup was used for both optic cup and optic disc segmentation and the results are compared to the benchmark methods on the used data set presented in Sivaswamy et al. (2015). As mentioned before, our method is inspired from CNNs

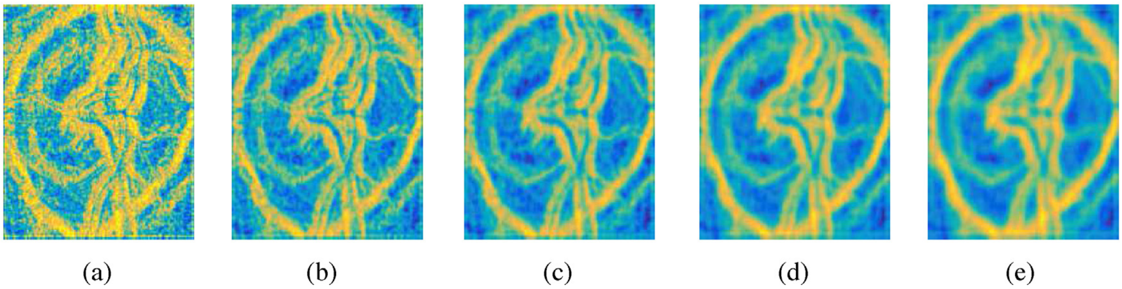


Fig. 10. Illustration of total entropy for different neighborhoods; (a) 3 × 3; (b) 5 × 5; (c) 7 × 7; (d) 9 × 9; (e) 11 × 11.

Table 4
Segmentation accuracy in terms of *F* score, overlap and boundary distance for different methods. *D*, *C* indicate if the method segments the optic disc or optic cup or both. *B* is in pixels; *F* – *F* score; *S* – overlap measure; *B* – boundary error.

	Proposed	*CNN*	Aquino et al. (2010)	Cheng et al. (2013)	Wong et al. (2008)	Joshi et al. (2011)	Xu et al. (2014)
Type	*D*, *C*	*D*, *C*	*D*	*D*, *C*	*D*, *C*	*D*, *C*	*C*
Optic disc							
F	97.3	96.7	93.2	92.1	91.1	96.0	–
S	91.4	90.4	86.1	85.1	83.9	90.1	–
B	9.9	10.5	12.3	12.9	14.8	11.1	–
Optic cup							
F	87.1	86.4	–	78.9	77.1	84.0	79.1
S	85.0	83.6	–	77.1	76.3	78.4	76.8
B	10.2	10.8	–	14.7	16.7	11.1	13.7

but is different from conventional CNN architectures. Hence we also compare the results of our method with a standard CNN architecture, referred to henceforth as *CNN*. The *CNN* is trained on the patches using 3 hidden layers having respectively 3,4,3 filters (or kernels) of size 3 × 3. Each convolutional layer is followed by a max-pooling operation which reduces the image dimension by a factor of 2 along both axes. The output of the third layer is the input to a fully connected layer whose output is the input to a soft-max classifier like our proposed method. A test image is subjected to the same sequence of operations and is classified using the soft-max classifier. The probability maps generated by the softmax classifier are processed using unsupervised graph cuts followed by a convex hull transform to get the final segmentation. Thus we see that the only difference between *CNN* and our *Proposed* method is the architecture.

Optic disc: Optic disc segmentation is less challenging than optic cup segmentation and existing methods have already achieved high accuracy values. Table 4 summarizes the segmentation performance of different methods. The competing methods are categorized as those which segment only the optic disk (*D*), optic cup (*C*) or both (*D*, *C*). Our method (*Proposed*) with *F*-score of 0.973 outperforms all the competing methods (which use hand crafted features) as is evident from the higher *F* and *S* values, and lower *B* values. The difference in *F* and *S* score values is also statistically significant since $p < 0.01$ (from Student-*t* tests) for all methods compared to *Proposed*. *CNN* also performs better than previously proposed methods. The average performance measures of *Proposed* is higher than *CNN*, although statistical tests suggest a very small significance between two sets of results ($p = 0.042$). This quite clearly indicates that our proposed method compares well with standard CNN architectures and also outperforms them. The superior performance of our method can be attributed to the fact that boosting allows the algorithm to explore the feature space and learn distinct representations of the training data.

Since Cheng et al. (2013) is a superpixel based approach, pixels from different classes may be grouped in one superpixel which affects its performance. Joshi et al. (2011) uses a modified Chan–Vese model, which finds it challenging to segment the optic disc using only intensity information. Aquino et al. (2010) uses only morphological features which is good enough for disc segmentation, but does not perform as well for cup segmentation. Xu et al. (2014) was designed specifically for cup segmentation and hence performs well. However *Proposed* outperforms all these methods.

Fig. 11 shows the comparative results of *Proposed* and selected other methods listed in Table 4. The example image is a typical case of glaucoma where the optic cup is enlarged and is almost as big as the disc. This example has a high CDR and has regions affected with PPA surrounding the optic disc. Our *Proposed* model outperforms even *CNN* and (Joshi et al., 2011) which are the best among the competing methods.

Certain images tend to throw the algorithm off track when evaluating the images as illustrated in Fig. 12. This could be due to somewhat local understanding of the algorithm. Additionally, the difficult cases for optic disc segmentation are mostly caused by peripapillary atrophy of the eye. This condition manifests in the images as a region around the optic disc that shares many of the characteristics of the optic disc such as color and texture. Due to this similarity, the convolutional filter algorithm is led astray by eliciting a similar response as for the correct optic disc region. A more global understanding of the image such as exhibited by a human observer would help to remedy this problem.

Table 5
CDR errors when compared to each individual expert and the average of all experts.

	CNN	*Proposed*	Sivaswamy et al. (2015)
Expert-1	0.131 ± 0.12	0.112 ± 0.08	0.15 ± 0.12
Expert-2	0.11 ± 0.09	0.098 ± 0.074	0.13 ± 0.10
Expert-3	0.095 ± 0.08	0.081 ± 0.062	0.10 ± 0.10
Expert-4	0.1 ± 0.08	0.091 ± 0.07	0.11 ± 0.11
Average	0.1 ± 0.08	0.080 ± 0.063	0.12 ± 0.09

Fig. 12 highlights the best case and worst case scenario for optic disc segmentation using *CNN*. In the best case a nearly perfect segmentation can be achieved. In the worst case the algorithm misses part of the optic disc region, possibly due to PPA.

Optic cup: Segmentation of the optic cup is more challenging than optic disc. This is especially marked by the disagreement in segmentation of clinical experts. *Proposed* gives the best results for cup segmentation, followed by *CNN*. The next best results are obtained by Joshi et al. (2011) using their $r-$bends technique that identifies the region of bending of blood vessels. However vessel bends can sometimes throw up erroneous candidates for cup boundary. Our convolutional filter based scheme using boosting seems more reliable than other algorithms for detecting the optic cup and hence outperforms *CNN* and Joshi et al. (2011) by a significant margin. Fig. 13 shows the best case and worst case result of our method for the optic cup. In the worst case scenario the algorithm experiences difficulties with images having a particularly small optic cup region. The size of small optic cup regions tends to be overestimated. However, from the overall results we can conclude that learning features through convolutional networks is of advantage in the challenging task of optic cup segmentation.

7.7. Cup-to-disc ratio

Cup-to-disc ratio (CDR) is calculated from the optic disc and cup segmentations by determining the ratio of the maximal diameter of the optic cup and the optic disc. The obtained CDR estimation results are compared to the performance obtained in Sivaswamy et al. (2015). All results were obtained using 5-fold cross-validation. The performance values are displayed in Table 5. As can be seen, the proposed method outperforms the combination of the best method for optic disc and optic cup segmentation used in Sivaswamy et al. (2015). Higher accuracy for optic cup and disc segmentation makes

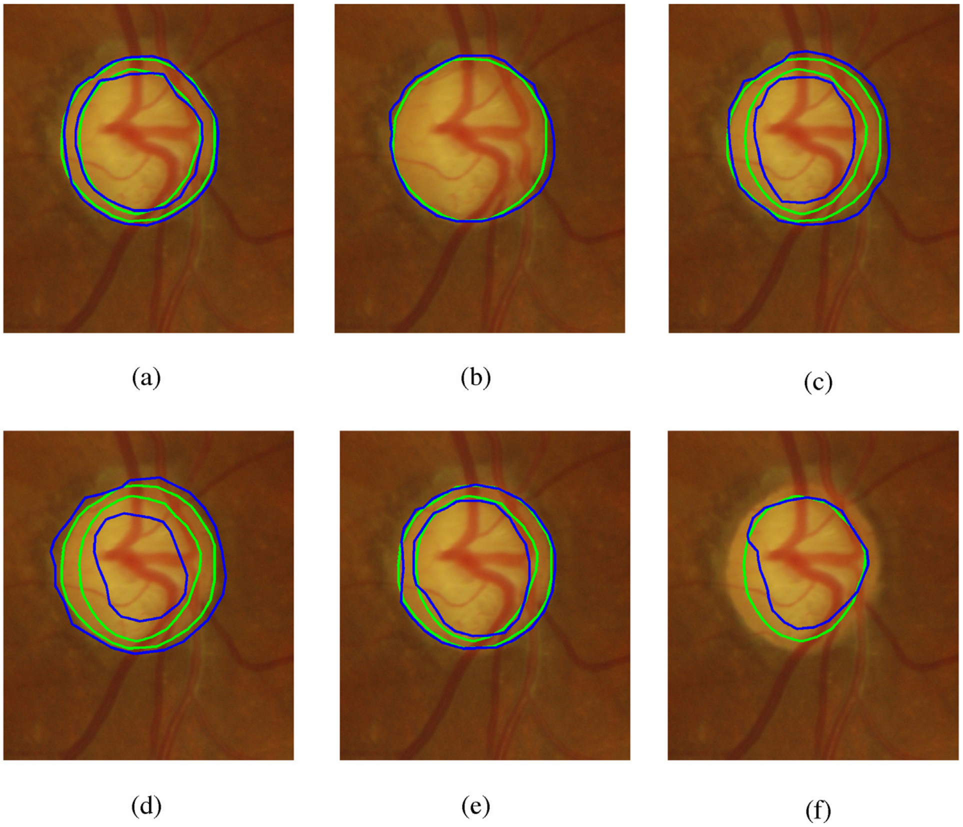


Fig. 11. Segmentation results for different methods: (a) our *Proposed* model; (b) Aquino et al. (2010); (c) Joshi et al. (2011); (d) Cheng et al. (2013); (e) *CNN*; (f) Xu et al. (2014). Note that Aquino et al. (2010) segments only the disc while Xu et al. (2014) segments only the cup. Green contour is ground truth while blue contour is algorithm segmentation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

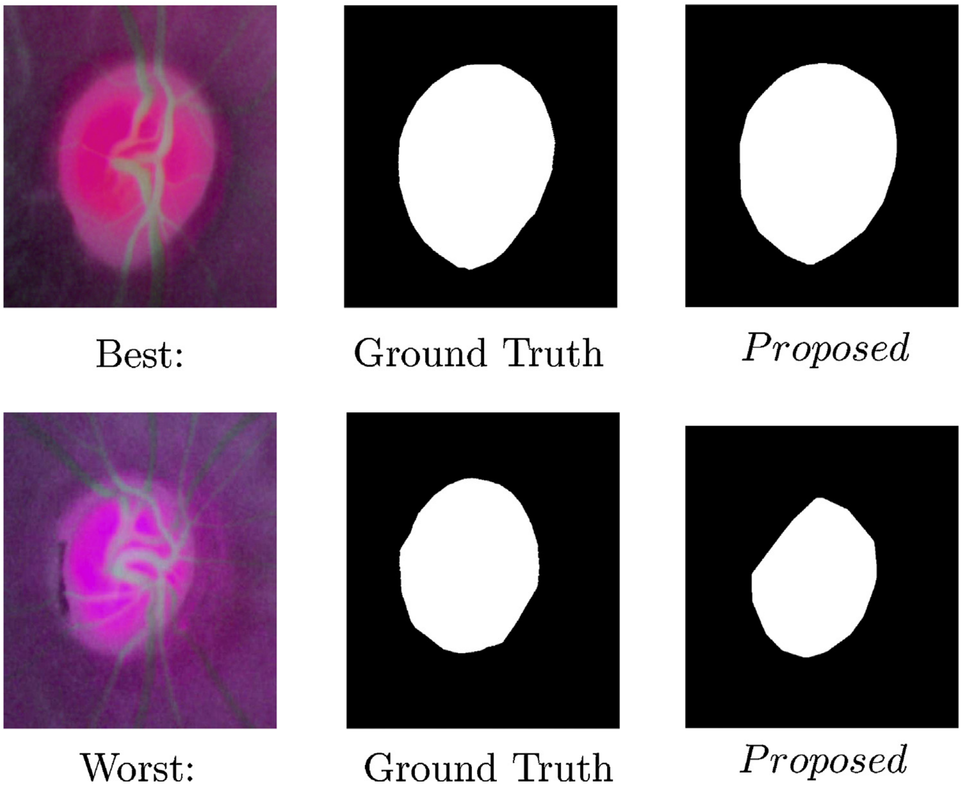


Fig. 12. Results of best/worst case segmentation for optic disc.

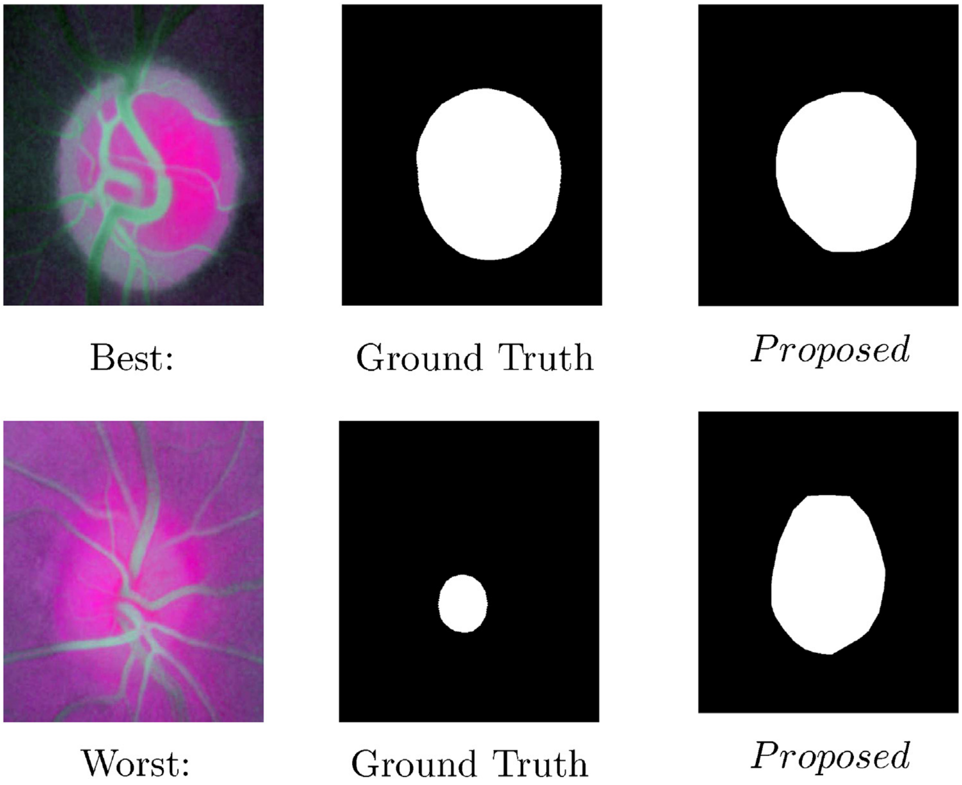


Fig. 13. Results of best/worst case segmentation for optic cup segmentation.

the overall CDR estimation of our method more accurate than other methods, including *CNN*.

Results on RIM-ONE v3 dataset: The RIM ONE dataset (Pena-Betancor et al., 2015) consists of 159 stereo retinal fundus images with optic disc and cup ground truth. The reference segmentations have been provided by two expert ophthalmologists. The dataset has 85 normal cases (no presence of glaucoma) and 74 confirmed glaucoma cases. We used our proposed method to segment the optic cup and disc from the images. We then fit ellipses through the OC and OD, and calculate the CDR as the ratio of the vertical diameters. A 5-fold cross validation was then used to determine classification accuracy.

We include all 85 normal subject images and 70 glaucoma cases for our evaluation. For every test fold we pick one-fifth of the dataset (17 normal and 14 glaucoma) for testing and the remaining for training. The average classification and segmentation results are reported in Tables 7 and 6. We observe that similar to Table 4, our proposed method performs better than competing methods. Table 7 also highlights that our method had the best performance in terms of sensitivity (fraction of correctly identified glaucoma cases), specificity (fraction of correctly identified normal cases) and overall accuracy. This clearly indicates that by using our proposed method we can accurately identify normal and glaucoma cases. The median CDR for normal cases was 0.63 and 0.76 for glaucoma patients. Note that in Table 7 we only present results for the methods common to both OC and OD segmentation.

7.8. *Optic disc segmentation On MESSIDOR dataset*

We apply our method to the MESSIDOR dataset for optic disc segmentation, and the results are summarized in Table 8. Results are shown in terms of Jaccard index which has been used to validate the performance of different algorithms on the dataset. Our method outperforms three competing methods and hence justifies our approach to learn convolutional filters.

7.9. *Comparison with deep CNNs*

The primary novelty of our approach is a method which can be used to learn convolutional filters when a very small dataset is available that cannot be used for reliably training deep CNNs. However, we train deep CNNs using patches from the original images

Table 6
Segmentation accuracy of RIM-ONE in terms of *F* score, overlap and boundary distance for different methods. *D*, *C* indicate if the method segments the optic disc or optic cup or both. *B* is in pixels; *F* – *F* score; *S* – overlap measure; *B* – boundary error.

	Proposed	*CNN*	Aquino et al. (2010)	Cheng et al. (2013)	Wong et al. (2008)	Joshi et al. (2011)	Xu et al. (2014)
Type	*D*, *C*	*D*, *C*	*D*	*D*, *C*	*D*, *C*	*D*, *C*	*C*
Optic disc							
F	94.2	93.4	90.1	89.2	88.3	93.1	–
S	89.0	88.3	84.2	82.9	81.2	88.0	–
B	10.8	11.6	12.9	14.0	15.1	11.9	–
Optic cup							
F	82.4	81.2	–	74.4	72.6	80.1	75.3
S	80.2	79.9	–	73.2	70.6	76.4	73.1
B	13.4	14.0	–	18.1	19.6	14.5	17.9

Table 7
Glaucoma classification accuracy for RIM-ONE using CDR values in terms of *Sen*, *Spe* and *Acc*.

	Proposed	*CNN*	Cheng et al. (2013)	Wong et al. (2008)	Joshi et al. (2011)
Sen	92.3	90.1	87.4	86.4	89.8
Spe	95.6	94.3	92.5	92.0	94.0
Acc	94.1	92.4	90.2	89.4	92.1

Table 8
Segmentation accuracy for MESSIDOR dataset in terms of Jaccard index for optic disc.

	Proposed	Yu et al. (2012)	Aquino et al. (2010)	Giachetto et al. (2014)
Jaccard	0.90	0.84	0.86	0.88

Table 9
Segmentation accuracy in terms of *F* score, overlap and boundary distance for different methods. *D*, *C* indicate if the method segments the optic disc or optic cup or both. *B* is in pixels; *F* – *F* score; *S* – overlap measure; *B* – boundary error; *HD* – 95 percentile Hausdorff distance; *p* – result of paired *t*-test (between the *F*-values of our proposed method and the given method).

	Optic disc		Optic cup	
	Proposed	Long et al. (2015)	*Proposed*	Long et al. (2015)
F	97.3	97.6	87.1	87.5
S	91.4	91.9	85.0	85.7
B	9.9	9.4	10.2	9.8
HD	10.1	9.5	10.5	9.8
p	–	0.042	–	0.039

and compare their performance with our method. We extract 3,000,000 patches to train the architecture proposed in Long et al. (2015). It proposes a novel skip architecture for pixel wise segmentation of a given image. The results are summarized in Table 9. The deep CNN outperforms our method by a small degree, which is not surprising given that this architecture is considerably deeper than our proposed architecture. This clearly illustrates that our method may not perform as well as deep networks when a large image database is available. However, it is not possible to train the skip architecture of Long et al. (2015) using only 50 images. Therefore, depending upon the size of the database, an appropriate architecture between the two can be chosen.

7.10. Computation time

Our whole pipeline was implemented in MATLAB on a 2.66 GHz quad core CPU running Windows 7. Training our algorithm takes around 15–20 min for 40 images (average 17.1 min per cross validation cycle). However segmenting a test image is very fast with an average of 5.3 s per image. Further increases can be expected when applying parallelization to processes that need not be calculated consecutively. Correspondingly, for *CNN* the training time is 37–42 min for 40 images (an average of 39.4 min per cross validation cycle), and the segmentation time for a test image is 8.1 s per image. These numbers clearly highlight the computational efficiency of our method, even on a small dataset.

7.11. Algorithm limitations

Some of the limitations of our algorithm are

- **Data:** Working with only 50 images for training does not permit a CNN inspired algorithm to develop its full potential. Very few filters were trained when compared to other CNN methods. Additionally, it is not clear that the provided images capture all the variability of retinal images. Abnormal exception cases might not be represented in the data set.
- **Algorithm:** The proposed method might prove to be not as directed as CNNs trained using backpropagation. Furthermore due to the sampling of points of the provided images there is a stochastic element in the algorithm. This means that the same performance cannot always be repeated. Interestingly, sampling more points does not in fact improve results but might simply reduce the fluctuation in performance. More research needs to be conducted into entropy sampling and how to ensure that sampling "interesting" points also translates into a better performance.

8. Discussion and conclusion

In this paper we have proposed a general framework for learning most discriminative representations of the training data in the form of convolutional filters. This eliminates the need for designing hand crafted features which are not always robust for different tasks. Our proposed CNN inspired ensemble learning architecture has been shown to better the state-of-the-art on the public DRISHTI-GS data set. From a research point of view our work makes two main contributions. First, a novel entropy sampling method is proposed that allows the algorithm to considerably reduce its computational effort while performing better than the simple uniform sampling approach. Secondly, building upon this technique, an original framework for learning convolutional filters in a principled manner using reweighted boosting was described. The aim behind this learning framework is twofold. First, a novel way of training convolutional filters in a network architecture was explored. Second, we expect the proposed method will be more amenable to theoretical insights into the fundamental principles governing CNNs or even deep neural networks (DNN) in general. These insights might be based on the realization that many of the characteristics of the proposed method are equivalent to ensemble learning systems.

Although a deep CNN network trained on numerous patches from the same dataset outperforms our method, it is not possible to reliably train a deep CNN from 50 images. However it is not possible to reliably train a deep CNN using a few images. Hence our proposed method can be used to learn convolutional filters when a small dataset is available.

Experimental results show that for the same number of samples, entropy sampling achieves superior results to uniform sampling. Similarly, boosting filters has shown to yield improved results when compared to bagged filters. This clearly indicates that boosting produces more discriminative filters than bagging. For convolutional network standards, the used DRISHTI-GS data set is rather small. In this particular case, it was essential to have an effective learning procedure that focused on the most important information. Having less data meant on the one hand that there was a smaller computational burden placed on any algorithm. On the other hand, less data means that less parameters can be reliably trained. Consequently, the proposed convolutional network is rather small compared to traditional CNN architectures. Nevertheless, the learned network has proved to be effective by out-competing several other methods that use hand crafted features.

References

Aquino, A., Gegundez-Arias, M.E., Marin, D., 2010. Detecting the optic disc boundary in digital fundus images using morphological, edge detection, and feature extraction techniques. IEEE Trans. Med. Imaging 29 (11), 1860–1869.

Aquino, A., Gegundez-Arias, M., Marin, D., 2010. Detecting the optic disc boundary in digital fundus images using morphological edge detection and feature extraction techniques. IEEE Trans. Med. Imaging 20 (11), 1860–1869.

Bengio, Y., Goodfellow, I.J., Courville, A., 2015. Deep Learning. Book in preparation for MIT Press.

Bock, R., Meier, J., Nyul, L.G., Hornegger, J., Michelson, G., 2010. Glaucoma risk index: automated glaucoma detection from color fundus images. Med. Image Anal. 14 (3), 471–481.

Boykov, Y., Veksler, O., 2001. Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. 23, 1222–1239.

Breiman, L., 1996. Bagging predictors. In: Machine Learning, pp. 123–140.

Brown, A., Hinton, G., 2000, November. Products of Hidden Markov Models. Technical Report GCNU TR 2000-008. Gatsby Computational Neuroscience Unit, University College London.

Chakravarty, A., Sivaswamy, J., 2014. Coupled sparse dictionary for depth-based cup segmentation from single color fundus image. In: Polina, G., Nobuhiko, H., Christian, B., Joachim, H., Robert, H. (Eds.), Medical Image Computing and Computer-Assisted Intervention MICCAI 2014. Lecture Notes in Computer Science, vol. 8673. Springer International Publishing, pp. 747–754.

Cheng, J., Liu, J., et al., 2013. Superpixel classification based optic disc and optic cup segmentation for glaucoma screening. IEEE Trans. Med. Imaging 32 (6), 1019–1032.

Cireşan, D.C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J.,2011. Flexible, high performance convolutional neural networks for image classification. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence – Volume Two, IJCAI'11. AAAI Press, pp. 1237–1242.

Ciresan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J., 2012. Deep neural networks segment neuronal membranes in electron microscopy images. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (Eds.), In: Advances in Neural Information Processing Systems, vol. 25. Curran Associates, Inc., pp. 2843–2851.

Doğan, H., Akay, O., 2010. Using adaboost classifiers in a hierarchical framework for classifying surface images of marble slabs. Expert Syst. Appl. 37 (December (12)), 8814–8821.

Freund, Y., Schapire, R.E., 1999. A Short Introduction to Boosting.

Giachetto, A., Ballerini, L., Trucco, E., 2014. Accurate and reliable segmentation of the optic disc in digital fundus images. J. Med. Imaging 1 (2), 024001.

Grant, M., Boyd, S., 2008. Graph implementations for nonsmooth convex programs. In: Blondel, V., Boyd, S., Kimura, H. (Eds.), Recent Advances in Learning and Control. Lecture Notes in Control and Information Sciences. Springer-Verlag Limited, pp. 95–110.

Grant, Michael, Boyd, Stephen, 2014, March. CVX: Matlab Software for Disciplined Convex Programming. Version 2.1.

Hubel, D.H., Wiesel, T.N., 1963. Shape and arrangement of columns in cat's striate cortex. J. Physiol.

Joshi, G.D., Sivaswamy, J., Krishnadas, S.R., 2011. Optic disk and cup segmentation from monocular color retinal images for glaucoma assessment. IEEE Trans. Med. Imaging 30 (6), 1192–1205.

Joshi, G., Sivaswamy, J., Krishnadas, S.R., 2012, June. Depth discontinuity-based cup segmentation from multiview color retinal images. IEEE Trans. Biomed. Eng. 59 (6), 1523–1531.

Kiros, R., Popuri, K., Cobzas, D., Jagersand, M., 2014. Stacked multiscale feature learning for domain independent medical image segmentation. In: Machine Learning in Medical Imaging. In: Guorong, W., Daoqiang, Z., Luping, Z. (Eds.), Lecture Notes in Computer Science, vol. 8679. Springer International Publishing, pp. 25–32.

Liao, S., et al., 2013. Representation learning: a unified deep learning framework for automatic prostate MR segmentation. In: Proc. MICCAI. Part 2, pp. 254–261.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: Proc. IEEE CVPR, pp. 3431–3440.

Mahapatra, D., Buhmann, J., 2015. A field of experts model for optic cup and disc segmentation from retinal fundus images. In: Proc. IEEE ISBI, pp. 218–221.

Mahapatra, D., Buhmann, J., 2015. Obtaining consensus annotations for retinal image segmentation using random forest and graph cuts. In: Proc. MICCAI-OMIA, pp. 1–8.

Mahapatra, D., 2016. Semi-supervised learning and graph cuts for medical image segmentation. Comput. Vis. Image Underst. (in press).

Mayraz, G., Hinton, G.E., 2002, February. Recognizing handwritten digits using hierarchical products of experts. IEEE Trans. Pattern Anal. Mach. Intell. 24 (2), 189–197.

Michelson, G., Hornegger, J., Wärntges, S., Lausen, B., 2008, August. The papilla as screening parameter for early diagnosis of glaucoma. Dtsch. Ärztebl. Int. 105, 34–35.

Pena-Betancor, C., Gonzalez-Hernandez, M., Fumero-Batista, F., Sigut, J., Mesa, E., Alayon, S., de la Rosa, M.G., 2015. Estimation of the relative amount of hemoglobin in the cup and neuro-retinal rim using stereoscopic color fundus images. Invest. Ophthalmol. Vis. Sci. 56 (3), 1562–1568.

Salah, M.B., Mitiche, A., Ayed, I.B., 2011. Multiregion image segmentation by parametric kernel graph cuts. IEEE Trans. Image Process. 20 (February (2)), 545–557.

Scherer, D., Müller, A., Behnke, S.,2010. Evaluation of pooling operations in convolutional architectures for object recognition. In: Proceedings of the 20th International Conference on Artificial Neural Networks: Part III, Berlin, Heidelberg, ICANN'10. Springer-Verlag, pp. 92–101.

Singer, D.E., Nathan, D.M., Fogel, H.A., Schachat, A.P., 1992. Screening for diabetic retinopathy. Ann. Intern. Med. 116 (8), 660–671.

Sivaswamy, J., et al., 2014. Drishti-GS: retinal image dataset for optic nerve head (ONH) segmentation. In: IEEE EMBC, pp. 53–56.

Sivaswamy, J., Krishnadas, S., Chakravarty, A., Joshi, G., Ujjwal, Syed Tabish, A., 2015. A comprehensive retinal image dataset for the assessment of glaucoma from the optic nerve head analysis. JSM Biomed. Imaging Data Pap. 2 (1), 1004.

Turaga, S.C., Murray, J.F., Jain, V., Roth, F., Helmstaedter, M., Briggman, K., Denk, W., Sebastian Seung, H., 2009. Convolutional networks can learn to generate affinity graphs for image segmentation. Neural Comput. 22 (2), 511–538, 2015/05/17.

Wang, X., Hänsch, R., Ma, L., Hellwich, O.,2014. Comparison of different color spaces for image segmentation using graph-cut. In: 9th International Conference on Computer Vision Theory and Applications. SCITEPRESS Digital Library.

Wong, D.W.K., et al., 2008. Level set based automatic cup to disc ratio determination using retinal fundus images in argali. In: Proc. IEEE EMBC, pp. 2266–2269.

World Health Organization, 2006. Vision 2020. The Right to Sight. Global Initiative for the Elimination of Avoidable Blindness. WHO Press.

Xu, Y., Duan, L., et al., 2014. Optic cup segmentation for glaucoma detection using low rank superpixel representation. Proc. MICCAI Part 1, 788–795.

Yu, H., Barriga, E.S., Agurto, C., Echegaray, S., Pattichis, M.S., Bauman, W., Soliz, P., 2012. Fast localization and segmentation of optic disk in retinal images using directional matched filtering and level sets. IEEE Trans. Inf. Technol. Biomed. 16 (4), 644–657.

Julian Zilly is a MSc student in the department of Mechanical Engineering at ETH Zurich. His interests are in the field of convolutional neural networks and applying them to different problems such as medical image analysis and robotics.

Joachim M. Buhmann is full Professor for Computer Science at ETH Zurich since October 2003. His research interests cover the area of pattern recognition and data analysis, computer vision and image analysis, remote sensing and bioinformatics. He also serves on the board of IEEE Transactions on Neural Networks and of IEEE Transactions on Image Processing.

Dwarikanath Mahapatra is currently a research scientist at IBM Research Melbourne. He was a post-doctoral research scholar at the Department of Computer Science, ETH Zurich. His interests are in applying machine learning for improving healthcare systems.