Machine Learning methods for Thin Vessel Segmentation

in Fundus Images

Kushal Khandelwal

Birla Institute of Technology and Science,Pilani

This thesis is submitted in requirements of the course BITS C421T

May 2015

Table of contents

List of figures

List of tables

Chapter 1

Introdcution

1.1 Motivation

With the advent of medical imaging, computer aided diagnostic systems have become an integral part of today's medical diagnosis[3]. CAD systems have become a part of routine clinical work and are being used extensively for disesase diagnosis.A myriad of different medical imaging systems,like X-Ray, Magnetic Resonance Imaging (MRI), Computed Tomography Scans etc, are used for diagnosis. The output of such systems are multi dimensional digital images, interpretation of which require sophisticated digital image processing methods. Automated medical diagnosis systems can aid in easy and faster interpretation of these images.

Digital fundus imaging in ophthlamology is a vital component in diagnosis of various pathologies. Retinal vessel segmentation forms an important part of diagnosis of such pathologies. Changes in the retinal vascualture is precursor to many diseases such as diabetes,hypertension and stroke. Morphological properties such as diameter, length, branching angle, of the retinal vessel forms an important component in diagnosis and evaluation of opthalmologic diseases such as diabetic retinopathy [26] and hyperternsion. For example, vessel diameter measurement can be an aid in diagnosis of hypertension[1].

Vessel segmentation in itself a challenging and a tedious task which may take a couple of hours one done manually. Low contrast between vessel and background, noise in the image and variability in the width, brightness and shape alongwith the presence of exudates,lesions, hemorrhage spots and other pathological effects make the task much more difficult. Figure 1.1 shows a fundus image of a diseased eye.

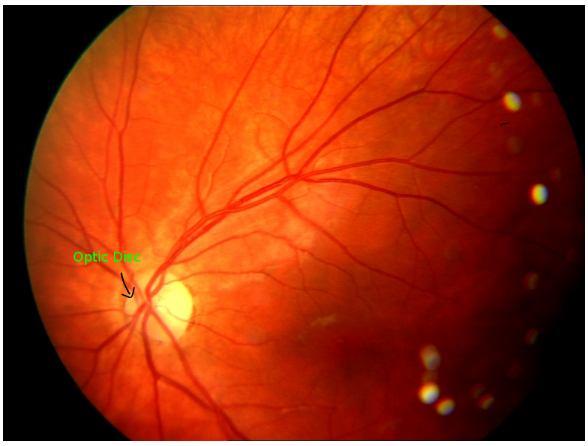


Fig. 1.1 A diseased fundus image

Developing an automated retinal vessel segmentation is a first step towards developing a full fledged CAD system for diagnosis of opthalmology pathologies. There have been a lot of work in literature on automatic retinal segmentation, including based on matched filtering, tracking methods, morphological methods and learning based methods. Some of the simplest approaches to segmentation use adaptive thresholding to segment out the blood vessels. Another simple approach is to use edge detection tecniques like canny edge detector for vessel segmentation. Some of the more sophsticated models use learning based algorithms which learn on a set of given image segmentations. Many of these models, learn local features at a pixel location and train a pixel based classifier.[21] present the limitations in some of the state-of-art methods and present a multi scale line detection based method for blood vessel segmentation.Issues, like poor segmentation at bifurcation and croosover regions, merging of close vessels and poor segmentation of small vessels limit the application of vasculature based medical diagnosis.For example, merging of two nearby vessels can lead to vessel being considered as one wide vessel, thereby affecting the width measurementes of the vessel. These limitations may contribute to inaccuracies in vascular network analysis and subsequently in characterization of diseases.

Supervised learning methods, in general are limited by the amount of training data.Also most of them are restricted to the type of training data and do not generalise on the task at hand. In our thesis, we propose two supervised learning models for accurate vessel segmentation. Our method is a patch-based framework and learns the local structure of the vessel at patch level. The proposed model, exploits the presence of common stuctures in a typical vasculature tree. Unlike most of the supervised learning methods which make a prediction at pixel level, classifying each pixel as vessel or background, our model predicts the local structure around each pixel at patch level. Figure, shows the basic architecture of our method.

1.2 Outline of thesis

The thesis is organized as follow,

- Chapter 2 provides some background knowledge on machine learning and used algorithms. It is followed by a brief overview on existing work on blood vessel segmentation.Finally, we explain in detail some of the methods by which our work is inspired.
- Chapter 3 starts by defining our patch-based framework for vessel segmentation. We then explain our models based on cluster learning and dictionary learning.
- Chapter 4 presents the evaluations and experimental results of our methods applied to three datasets. We present the effect of different setting on the performance. We also compare our evaluations with some of the work in the literature.
- Chapter 5, finally gives our concluding remarks and sets the tone for future work that migh be carried on this approach.

Chapter 2

Background and Literature Review

In this chapter we introduce the basic concepts and methods involved in our thesis. We start by introducing machine learning and the various machine learning algorithhms. We introduce the idea of clustering and dictionary learning.The sparse coding and dictionary learning formulations are discussed in this section. Later we discuss the relevant previous work relating to our thesis. In the last section we briefly describe some of the latest work which resembles our work and represents the state of art methods.

2.1 Machine Learning

Machine learning is the study dealing with the process to learn characteristic information from data. Given some data, X , a machine learning algorithm learns a function $f(X)$ which maps the input X to an output variable y . The learnt model than can be used to make predictions on previously unseen data X'.A machine learning algorithm learns the parameters of an adpative model from a training set, typically optimizing a function. The learnt model is then used to make predictions on previously unseen new data. Broadly, machine learning methods can be divided into two subfields, supervised learning and unsupervised learning. In a supervised setting the target labels or values are known for the test set and we focus on predicting the target value for previously unseen data.In an unsupervised setting no such target information is present for the training data and the focus is to learn structure and compact descriptions from the data.

In a supervised setting , given a set of labelled data points known as the training data:

$$T = \{(x_1, y_1), ..., (x_n, y_n)\}$$

, where $x_i \in X$ and $y_i \in X$,we find a function f, which maps any point in the domain of X to its corresponding label in Y. If Y is a set of discrete values then the it is known as a classifcation problem and if Y is in a continous range then it is a problem of regression.

In an unsupervised setting, the task is to find group relations between instances of the unlabeled training dataset with a subsequent aim of categorizing or clustering the data. The algorithm finds the previous unknown structure in the data.

The data point x_i is typically represented as a vector comprising of feature values and is known as a feature vector. For example, given a dataset $X = \{x_1, x_2, ..., x_m\}$ where x_i is an image patch of size $\sqrt{n}$ x $\sqrt{n}$,each patch can be represented as a vector of length n formed by concatenating the gray scale values. The entire dataset then can be represented as a matrix:

$$X = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{pmatrix}$$

Here each row of the matrix dentoes individual data points and $x_{i,j}$ is the gray scale intensities of patch x_i at pixel location j.

2.2 Clustering

A cluster is collection of data points grouped together on basis of some common properties.The data objects or points within a cluster are similar to each other, whereas the points in different groups are disimilar.

The process of partioning the data points into smaller groups (called as clusters) with an aim to minimize the intra cluster variance and maximize the inter cluster variance, is known as clustering. The grouping of data points is based on the similarity or disimilarity of the objects as described by their properties or features.

Similarity or disimilarity between two objects can be very subjective and hence various measures are used to describe them quantitatively with distance measure being the most common.The distance measure is used to specify the distance between two objects and can be used to create a distance matrix called as similarity/disimilarity matrix. The most commonly used distance metric is the Eucledian Distance.

The eucledian distance D between two points $p = \{x_1, x_2, ..., x_n\}$ and $q = \{y_1, y_2, ..., y_n\}$ can be defined as:

$$\begin{aligned} dist(p,q) &= \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2} \\ &= \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \\ &= \|\mathbf{p} - \mathbf{q}\| \end{aligned} \tag{2.1}$$

As the data labels are unknown,cluster analysis is known as an unsupervised method of data partionining.This is in contast to classification where the data can be partitioned on the basis of their class labels. Thus, clustering segments the data based on properties of the objects within the dataset and finds previously unkown grouping within the data.

2.2.1 K-Means Clustering

In this section we look into the K-Means clustering algorithm (MacQueen et al. [15]). We assume that the number of clusters 'K' is given and we use it to initiate our clustering algorirthm. The idea is to partition the dataset into 'k' clusters and assign 'k' centorids , one for each cluster

Given a datast D =$[x_1, x_2, ..., x_n]$ consisting of n objects ,where each object is a feature vector of length 'm'.The clustering algorithm partitions the dataset into 'K' clusters $C_1, C_2 ... C_k$ such that $C_i \subset$ D and $Ci \cap Cj = \emptyset$ and with an aim to have a low intra cluster varianace and a high inter cluster varianace.This means that the points within the same cluster musts be highly similar and disimilar to the points belogning to other clusters.

The partioning is performed in an iterative manner and follows what is known as Lloyd's algorithm[11]. We inititialze the 'k' centroid by some initialization methods such as k-means++ or random initilization.The cluster quality depends on a goood initialization.We start by assigning each point in the dataset to the nearest neareast centroid and we get 'k' clusters. We then calculate the mean points of the new clusters and assign them as the new centroids. The process then continue till a stopping criterion is reached. The partioning is made with an aim to minimize the objective function which is the withing cluster sum of squared error.

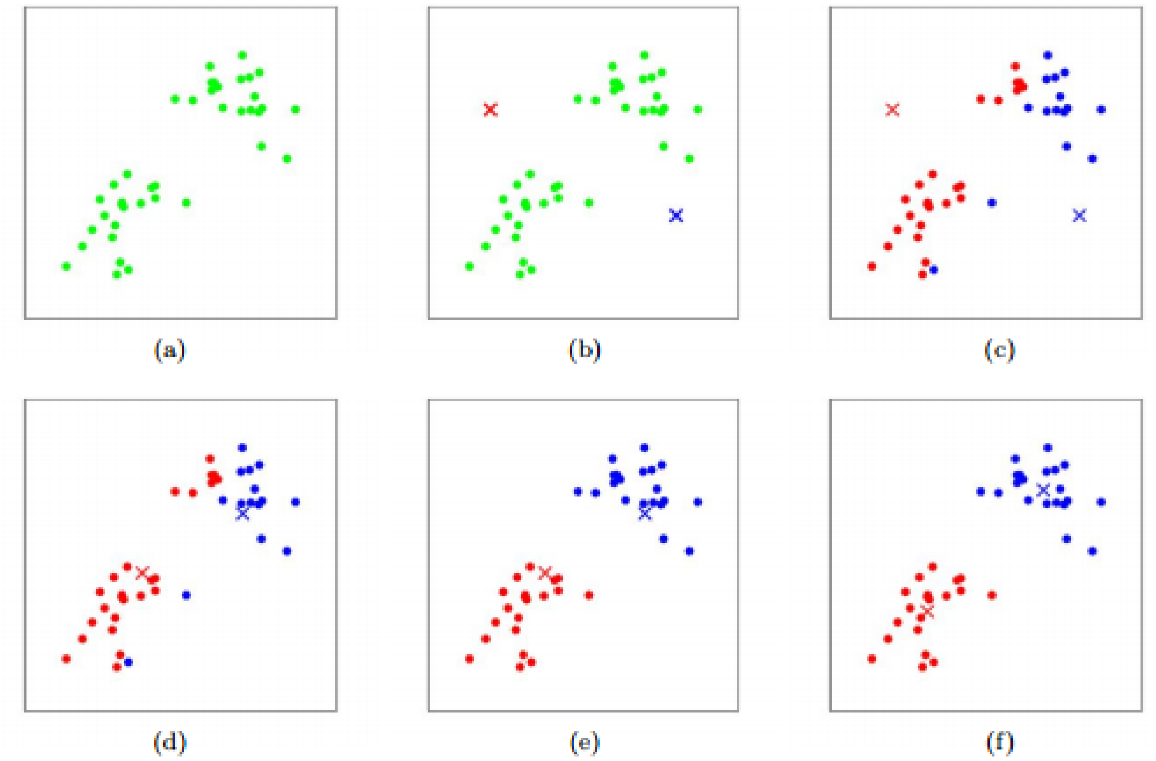


Fig. 2.1 K-means algorithm. Training examples are shown as dots, and cluster centroids are shown as crosses. (a) Original dataset. (b) Random initial cluster centroids. (c-f) Illustration of running two iterations of k-means. In each iteration, we assign each training example to the closest cluster centroid (shown by "painting" the training examples the same color as the cluster centroid to which is assigned); then we move each cluster centroid to the mean of the points assigned to it. (Images courtesy of Michael Jordan. Retrieved from : http://stanford.edu/ cpiech/cs221/handouts/kmeans.html)

Each of the clusters C_i consists of p points and is represented by its centroid c_i.The distance between point p and the cluster centre ci is defined as dist(p,ci). If the data points are in m dimensional , then the distance can be computed as in Eq 2.1. We can then write our objective function J as :

$$J = \sum_{j=1}^{K} \sum_{i=1}^{p} ||\mathbf{x_i^{(j)}} - \mathbf{c_j}||^2 \tag{2.2}$$

K-means clustering is the simplest unsupervised learning algorithm and can be applied to many different problems. An example of K-Means clustering on a toy dataset with 2 clusters is shown in Figure 2.1

2.3 Spare Coding and Dictionary Learning

The process of learning a finite set of basis elements ,called atoms, from a given training set of signals $X = [x_1, x_2, ..., x_n]$, such that each signal in X can be approximately reconstructed by a linear combiantion of atoms is called Dictionary Learning. This finite sets of atoms is called D, is the learnt dictionary. Normally we want to find a sparse decomposition of signals in D i.e, we aim to create a dictionary of sparse elements. This kind of decomposition is different from PCA where the basis elements have to be orthogonal to each other and these dictionaries are in general overcomplete.

By overcomplete here we mean that if any element from the dictionary is removed, we can still approximately construct the signal from the atoms. Also the number of atoms in dictionary is of a greater dimension than the data. Lewicki and Sejnowski [13] describes the process of learning overcomplete set of representations.Some recent work [18] has shown that sparsity captures high-order correlation in data and it captures higher order statistics in data.

Huang and Aviyente [10] talks about how to represent a given signal as a sparse decomposition of a dictionary D. In a sparse coding task, our goal is to represent a signal $x \in \mathbb{R}^m$, as a linear comibination of an overcomplete dictionary $D = [d_1, d_2, ..., d_k] \in \mathbb{R}^{nxk}$. Here 'k' is the no of atoms $d_k \in \mathbb{R}^n$. To ensure that the dictionary is overcomplete, we have $k > n$. The task is to find a representation in form a sparse code vector α of k x 1, such that:

$$\alpha = \min_{\alpha'} \|\alpha\|_0, \ \ s.t \ \ x = D\alpha, \tag{2.3}$$

where $\|x\|_0$ is l0 norm, which represents the number of non-zero components in x.

It has been found that this problem is hard to solve and thus we replace l0-norm with l1-norm, so then as others (Lee et al. [12] , Donoho [5]), the problem is reformulated to find α by mininimizing the objective function J:

$$J(\alpha;\lambda_1) = \min_{\alpha \in \mathbb{R}^k} \|\mathrm{x} - \mathrm{D}\alpha\|_2^2 + \lambda_1 \|\alpha\| \tag{2.4}$$

where the parameter λ_1 is the regularization parameter which ensures a trade off in sparsity and reconstruction error. It also ensures that the sparse codes are not very large.

To further ensure that the dictionary values D, do not become arbitarily large,[16] constarin the columns of D to have an l2-norm less than or equal to one. They call the convex set of matrices verifying this constrain as :

For our purpose we use the Online dictionary learning implementation of [16]. This is a very efficient methods and works brilliantly for millions of training examples. Sparse coding is done using the orthogonal matching pursuit (OMP) method [22, 29] implemented in the SPAMS libary provided by Mairal et al. [17]

2.4 Literature Review

In this chapter we aim to present a general overview of vessel segmentation methods.We particularly focus on approaches based on machine learning methods both supervised and unsupervised techniques. There has been a considerable work in the domain of automatic retinal vessel segmentation. Fraz et al. [7] give a complete review of retinal vessel segmentation methods. We briefly outline some of the methods before we move on to explain the most recent supervised learning based algorithms.

The earliest work on retinal segmentations are based on tracking methods to trace the blood vessels[2, 28]..In these methods the vessel is traced out from some starting seed points according to some relevant criterion. Recently,Vlachos and Dermatas [31] proposed a methods based on multi scale line tracking. They approach the problem by tracking the retinal vessel at multi scales and combining the individual image maps at different scales.Farnell et al. [6] employ multi scale line operators at various orientations to enhance the blood vessels.

There has been a recent surge in utilzing supervised machine lerning based methods for vessel segmentation. In such methods, generally, each image pixel is represented as a feture vector computed using local or gloabl information. A classifier is then trained to label every pixel to a vessor or background. Such classifiers rely on the training dataset, which is not available easily.Ricci and Perfetti [23] proposed a supervised classifier using line operators and using support vector classification. They obtain unsupervised pixel classifcation by thresholding the response of basic line detector. Additionaly, they use orthogonal line detectors along with grey level of target pixel to construct feature vector for support vector machines based supervised classification.Staal et al. [27] proposed a ridge based vessel segmentation method used together with a supervised learning technique.Nguyen et al. [21] proposed a method utilzing gabor filter responses at multiple scales as features for pixel classification.[9, 30] proposed neural network bases approach to vessel segmentation

We now look in detail some of the recent state-of-art work on retinal vessel segmentation. Dollár and Zitnick [4] recently proposed a structured learning framework for predicting local edges utilizing random forests.The work exploits the presence of local structure forms like straight lines or T-junctions in image patches to lean an accurate edge detector. They train random forests in structured output spaces. Local image labels are highly interdependent and utilzing the knowledge on such local structures tend to imprtove the the classifiers accuracy. The problem of edge detection is formulated in a way to predict local segmentation maps for input image patches. The structure labels are mapped to proxy discrete labels, in a way that similar structure labels are assigned to same discrete labels. Using this proxy mapping, existing random forest training approaches are used to learn structured random forests. Finally these forests are used to label each pixel to denote the presence or absence of edges.

Rigamonti and Lepetit [24] proposed a method to learn ad hoc features with learned filters.Many hand crafted features have been proposed to solve the problem of extracting linear structures like blood vessels, but are not always effective.In contrast, learned features are better at the task but are computationally expensive. The proposed algorithm complements handcrafted features with learned filters. Linear filters are computed from training images in a convolutional approach. The convolutional filters are constrained to be be dissimilar. Further, feature maps are computed by convolving the learned filters with the images. Several such feature maps our computed both for handcrafted features like(OOF,EF) and for learned filters. These features maps are then used to calculate descriptors at each image location. A random forest classifier is then trained to classify each image pixel as lying on a linear structure or background.

Ganin and Lempitsky [8] proposed a very elegant approach to natural edge detection or thin object segmentation. They employ a combination of convolutional neural networks with the neareast neighbor search. The problem is approached in a patch based framework, where the predictions are made patch by patch.

Chapter 3

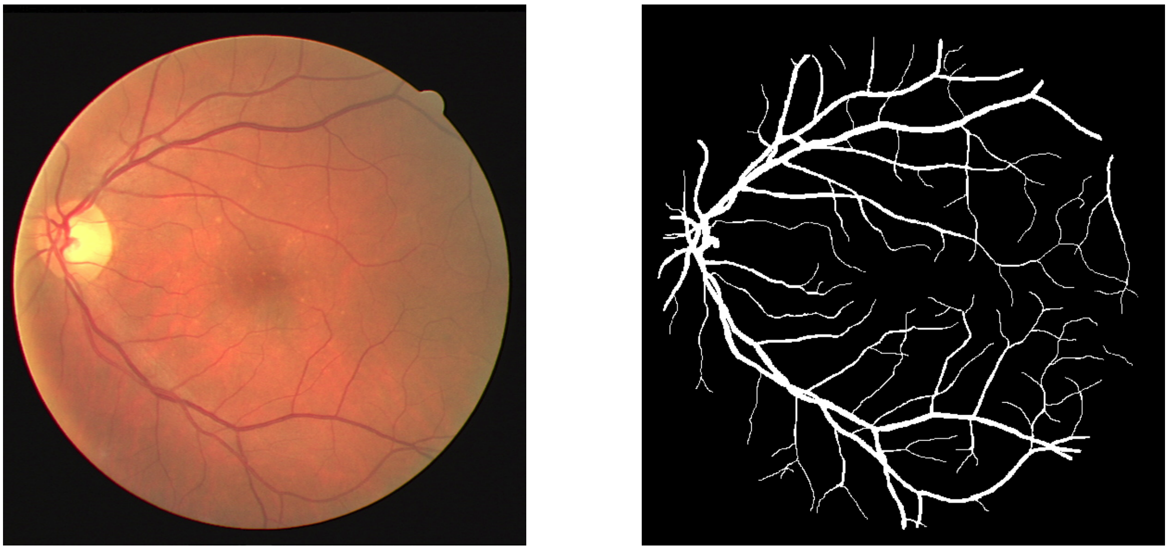
Framework

In this chapter we present the basic framework of our vessel segmentation method. We start by setting up the various notations used throughout the chapter and give an overview about how the framework is applied.Next we discuss the two different models applied in the framework.This forms the core of this thesis and represents the major work done. At last we mention the various datasets utilized to test our models. In the next chapter we experimentally evaluate the models.

3.1 Vessel segmentation Framework

We start by defining the basic notations and methods. The input to our vessel segmentation framework is a training set composed of fundus images and their corresponding segmentation or label maps. The collection of fundus images is represented by $\bar{I} = (I^{(1)}, I^{(2)} ...., I^{(n)})$ and their correspoding ground truth segmentation maps as $\bar{S} = (S^{(1)}, S^{(2)} ...., S^{(n)})$. Each given image is of size M x N. For simplicity, in the rest of the section a we would refer an image as I and its segmentation map S. We also assume that the image I is a single channel image.

We can represent an image as a collection of overlapping patches computed around every pixel take at center. A patch is of size $\sqrt{n}$ x $\sqrt{n}$ unless otherwise noted. We represent a patch centered at location (x ,y) as $p_{x,y}$, and patches for I and S can be represented as $Ip_{x,y}$ and $Sp_{x,y}$ respectively. For simplicity we would refer the pixels of patch as p_i for i =[1...n], where 'i' is the pixel location. A patch can also be represented as a vector , obtained by concatenating the image patch values.



(a) Fundus Image (b) Segmented Vessels

Fig. 3.1 Fundus Image and its segmentation

We then represent our dataset with each image in terms of their patches as:

$$I = \begin{pmatrix} I^{(1)}p_1 & I^{(1)}p_2 & \cdots & I^{(1)}p_n \\ I^{(2)}p_1 & I^{(2)}p_2 & \cdots & I^{(2)}p_n \\ \vdots & \vdots & \ddots & \vdots \\ I^{(M)}p_1 & I^{(M)}p_2 & \cdots & I^{(M)}p_n \\ \vdots & \vdots & \ddots & \vdots \\ I^{(N)}p_1 & I^{(N)}p_2 & \cdots & I^{(N)}p_n \end{pmatrix}, S = \begin{pmatrix} S^{(1)}p_1 & S^{(1)}p_2 & \cdots & S^{(1)}p_n \\ S^{(2)}p_1 & S^{(2)}p_2 & \cdots & S^{(2)}p_n \\ \vdots & \vdots & \ddots & \vdots \\ S^{(M)}p_1 & S^{(M)}p_2 & \cdots & S^{(M)}p_n \\ \vdots & \vdots & \ddots & \vdots \\ S^{(N)}p_1 & S^{(N)}p_2 & \cdots & S^{(N)}p_n \end{pmatrix} \tag{3.1}$$

Fig 3.1 shows an example of fundus image on the left and the correspoding segmentation image on the right.

The aim of our thesis is to learn a function F which maps the image to its segmentation as, $F : I \rightarrow S$, i.e, we want to learn a function to segment the vessel in a given fundus image. Inspired by the work of [14, 19] we approach the problem of predicting segmentation maps in a patch based framework. In this patch level method, to predict the segmentation map for an image I, we decompose the image into overlapping patches and predict the segmentation maps for individual patches. in a patch by patch fashion. The image and the segmentation map after decompostion can be represented as in Equation 3.1.Here 'I' denotes the fundus image and 'S' its segmentation map.//

At the test time, we compute the patches for a given image and the mapping fucntion F is the used to predict the segmentation map for all the test patches.Finally,the output segmentation patches are combined by averaging ,resulting in an output segmented image.As the patches are computed in an overlapping way, to each pixel several segmentation patches are added during reconstruction phase. More precisely, each pixel is composed of N pixel values from N nearby patches.

3.1.1 Learning Architecture

The major part of our work is to learn an effective mapping function F, to map the fundus image to its segmentation map. In both of our models we learn local structure representations in form of a dictionary. This dictionary is composed of patches representing the edge structures, t- junctions, crossovers, parallel lines and other commonly occuring local patterns. To these patches is associated their ground truth segmentations. At prediction time, we then approximately decompose each patch as a linear combinaton of dictionary elements.To compute the final segmentation of the patch, each atom of the dictionary element in the linear combination is replaced by its segmentation.

Both of the models presented in the section, learn this dictionary for predicting the ground truth map. Before we delve into our models, we list the common pre processing operations on the input data.

Preprocessing

For our task we have explored some of the preprocessing steps including patch normalization, contrasts streching and local contrast normalization. For all the experiments, we normalize each input patch by subtracting the patch mean and dividing by the standard deviation per patch. Also, all the input patches are vectorized before being fed to the predictor.

3.2 Cluster Learning

In this model we find common local structures within the data using clustering. Image patches are clustered to find a common set of locally appearing structures within the image. As we know the ground truth annotations associated to each image patch, we find the segmenation maps for our clustered structures by associating each segmentation image patch to its corresponding cluster. In this way we learn a dictionary in the form of {image patch :

segmentation} which is then used to find segmentation map for new patches. This method is inspired in part by work of [14] where they learn commonly occuring structures by clustering on annotated segmentation patches.
We call our method as 'Cluster Based - Common Local Structures(CB-CLS)'. Figure shows an exaple of learnt dictionary on drive dataset.

In the next section we describe in detail the implementation of learning and prediction steps of our algorithm.

3.2.1 Learning

The training procedure for our method requires learning clusters within the input patches. Training is performed on the patches drawn from the input images I_1,I_2,...I_n and corresponding segmentation images S_1,S_2,...S_n.

From the training images, we compute a random subset of m patches of fixed size $\sqrt{n}$ x $\sqrt{n}$ and their corresponding annotation as computed from the segmentation images. This subset of patches is used to learn a representative dictionary of patches representing the locally occuring structurein the image and the associated segmentation maps. Let us denote a patch as x_i and its corresponding segmentation map patch as y_i. Then the new trainig set can be represented as:

$$X = \{x_1, x_2, ..., x_n\}$$

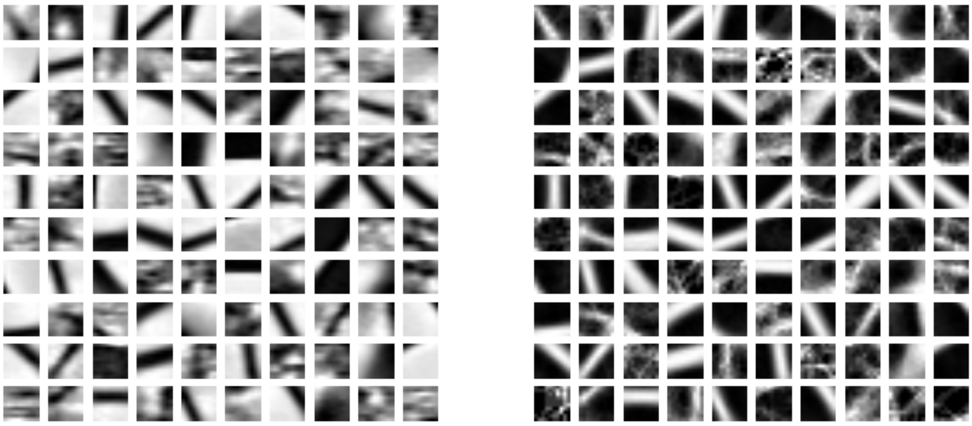
$$Y = \{y_1, y_2, ..., y_n\}$$

In the next step we find 'k' clusters within the patches in X usin K-Means clustering as explained in 2.2.1. Let the 'k' learnt cluster be denoted as $C = \{C_1, C_2, ..., C_k\}$ where the clusters then can be represented by their cluster centroids $c_1, c_2, ..., c_k$. A cluster with 'm' patches then can be represented as

$$C_k = \{x_1^k, x_2^k, ..., x_m^k\}$$

Now, since we know the annotation y_i for each training patch x_i, the image patches in each cluster can be repalced with their corresponding annotation patches to obtain a group of segmentation cluster where each such cluster can be represented as:

$$SC_k = \{y_1^k, y_2^k, ..., y_m^k\}$$



(a) Learned cluster centers. (b) Segmentation maps for learned clusters.

Fig. 3.2 The image on the left shows the local structures appearing in the images. We can see a mix of straight lines at different orientations and other complex structures. The vessels are in black. The image on the right depicts the corresponding vessels. Vessel pixels are in white and the background in black.

We then represent each segmentation cluster by the average of all the points within that cluster, found as:

$$sc_k = \frac{1}{m}\sum_{i=1}^{m} y_m^k$$

Finally, we obtain a dictionary which maps the image cluster to a ground truth segmentation patch.

$$D = \{(C_k : sc_k) | k = 1..m\}$$

An example of a learned dictionary over image patches on DRIVE dataset is showin in Figure 3.2

3.2.2 Prediction

At test time, to predict the segmenation map for an image, we start by extracting dense patches centered at all pixel in the image. For each extracted patch its cluster membership is predicted using the already trained clustering model. The cluster labels are then matched in the dictionary to obtain the respective segmentation map for the patch.Output image is reconstructed from the patch annotations by averaging.

3.2.3 Algorithm

The entire learning algorithm than can be summarized in following steps:

1. Extract a set of random Image-Segmentation patches of size m x m from the training images.

2. Train a K-Means clustering model on the input image patches and learn k clusters on the input.

3. Compute the ground truth patch for each cluster by averaging the ground truth annotations for each patch belonging to a cluster.

4. Construct a dictionary of Cluster and annotations.

3.3 Dictionary Learning

The model is based on learning an overcomplete dictionary over the training data. We aim to learn a good dictionary which can represent our data as a sparse linear combination of the dictionary atoms. This method is similar to our previous method where also we learn a dictionary represent the local structures withing the data. The mathematical reasoning behind is that, a signal (in our case an image) can be represented as a combination of a small number of basic structures, like edges and lines. It is also easy to obtain the segmentation maps for such basic atoms. In our method, for each such image patch, denoted as dictionary atoms, we find a corresponding segmentation label patch from the given segmentation map for the image.

For example, given an overcomplete dictionary of atoms D $\in R^{m,k}$, with k columns as the learnt image atoms, we can reconstruct a given image patch x $\in R^m$ as a linear combination of atoms in D. More formally, we learn a sparse code α over D to represent x as

$$x \approx D\alpha$$

This is very similar to our cluster based methdo. The learnt cluster centers can be considered as the dictionary with the cluster centers at dictionary atoms. And each point within a cluster can be represented as a linear combination of clusters centers with coefficients α, constrained as $\|\alpha\|_0$.This constraint forces our data point to be represented by only one cluster.

The advantage of the dictionary learning method is that it can represent complex structures like t-junctions and crossover regions much more accurately as a combination of various atoms. In the following section we describe the implementation of the learning model.

3.3.1 Learning

The learning phase starts by extracting image and label patches from the training data. Let us represent our training set composed of random patches as,

$$X = \{x_1, x_2, ..., x_n\}$$

$$Y = \{y_1, y_2, ..., y_n\}$$

where X is composed of image patches and Y the corresponding label patches. Each image patch can be representated as a vector by concatenating each of the pixel values. So given, 'm' patches of size $\sqrt{n}$ x $\sqrt{n}$,the dataset can than be represent as matrices X,Y $\in R^{m,n}$. Each row in the matrix denotes a patch or image atom.

We learn an overcomplete dictionary D of basis atoms from the image patches using the dictionary learning methods as described in section 2.3. The aim is to learn a dictionary D with k atoms which can best represent x_i using linear combination of a few atoms d_k of the learnt dictionary.

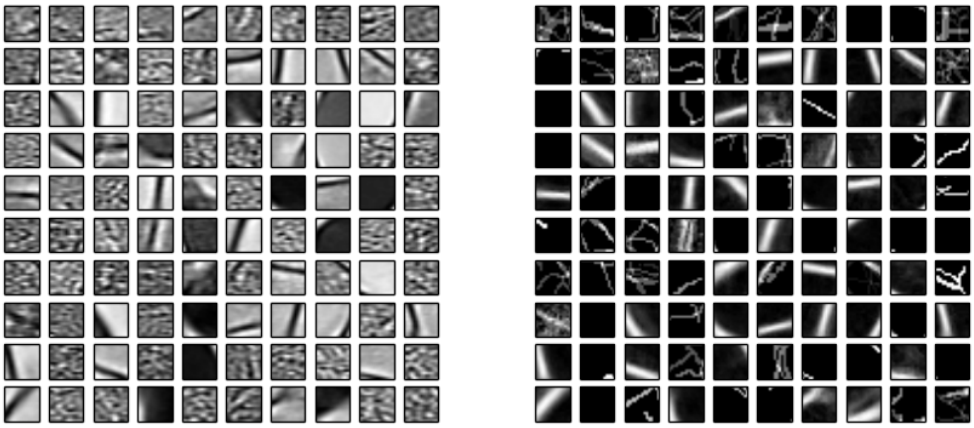
In the next step we learn a sparse code α to represent our input X in terms of dictionary atoms.

$$X = \alpha D$$

Since we already know the ground truth annotation Y for X, we can infer that Y can be approximately represented as a linear combination of atoms from a dictionary of labels L, corresponding to D.

As the sparse code remains same for both the representations, we can compute the dictionary atoms from L by taking a weighted average of all the ground truth patches in Y, weighed by sparse codes for all the patches.

So we learn two dictionaries, D of image patches and L consisting of label patches corresponding to atoms in D. An example of the learned dictionaries D and L over DRIVE dataset is shown in figure 3.3



(a) Dictionary D of atoms learned on the raw image patches

(b) Dictionary L, consisting of computed segmentatiom maps for atoms in D

Fig. 3.3 The image on the left shows the learnt representative atoms on raw image patches. Most of the atoms depict simple structures like lines and edges. The vessels are in black. The image on the right depicts the corresponding vessels. Vessel pixels are in white and the background in black.

3.3.2 Prediction

At the test time, given an image I, dense patches can be computed and the image can be represented as a matrix I ∈ R, where each row represents a patch of size n x n. We then compute a sparse decomposition of I over dictionary D represented by a sparse code α such that:

$$I = D\alpha$$

We can then compute the ground truth segmentation S, for the image patches in I, as :

$$S = L\alpha$$

where α is the sparse code computed over dictionary D and L is the label dictionary previously computed during the learning phase.

3.4 Datasets

For testing the performance of our algorithm , we train and test our system on the following publically available datasets. In this section we describe the characteristics of these datsets

3.4.1 DRIVE

The Digital Retinal Images for Vessel Extraction (DRIVE) dataset consists of 40 color retinal images randomly selected from diabetic retinopathy screening program for 400 diabetic patients.Each of these images in dataset are JPEG compressed and have a dimensions of 768 x 584 pixels captured at a resolution on bits per pixel. The images are captures with a 45degree field of view (FOV). Of the 40 images in the datset, 7 show sign of diabetic retinopathy , while the remaining 33 do not consist of any pathology. Each image is provided with a corresponding mask delineating the FOV.

The dataset is divided is provided with divisions in terms of training and testing set, with each set consisting of 20 images. Each of the 40 images have been manually segmented by human observers trained by an experienced optamologist. For the training set, single ground truth segmentation of the vessels is provided. The test set is provided with two ground truth segmentations, of which the first one is used as gold standard and the other is used to compare the performance with an independent human observer.

3.4.2 STARE

The STARE dataset consists of 20 images with blood vessel segmentations, out of which 10 show signs of pathology. The images have been capture with a FOV of 35degrees at 8 bit per pixel resolution, with dimesnsions of each image as 605 x 700 pixels. The dataset consists of segmentation provided by two human observers. In our experiements, we consider the segmentations provided by the first observer as ground truth.

For the experiements, the dataset is randomly divided into training and test sets each consisting of 10images.

3.4.3 HRF

he High Resolution Fundus (HRF) image databse consists of 45 images of which 15 images come from healthy patients , 15 from patients with diabetic retinopathy and 15 of glacumatous patients. The images were captured with a FOV of 60degrees, at a high resolution of 24bits per pixel. The size of each image is 3504 x 2336 pixels and stored with JPEG compression.

Table 3.1 Summary of Datasets

Datasets	FOV	# of Images
DRIVE	45	20+20
STARE	30	20
HRF	60	45
ARIA	50	212

Each image is provided a manual segmentation of vessesl as segmented by three independent human observers trained by experienced optahmologists.The dataset also provides a corresponging mask image of each image delineating the FOV.

3.4.4 ARIA

The ARIA dataset consists of three groups of images. One of the group consists of 92 images with age-related macular degeneration, the other with 59 images from diabetic patients and the last group with 61 images from a control group.

The images are captures with a 50degree FOV, stored in uncompressed TIFF format, with a resolution of 8bits oer pixel. Each image has dimensions of 768 x 576 pixels. The dataset provides with blood vessel segmentation images as manually segmented by experts and a corresponding mask delinating the FOV region.

Chapter 4

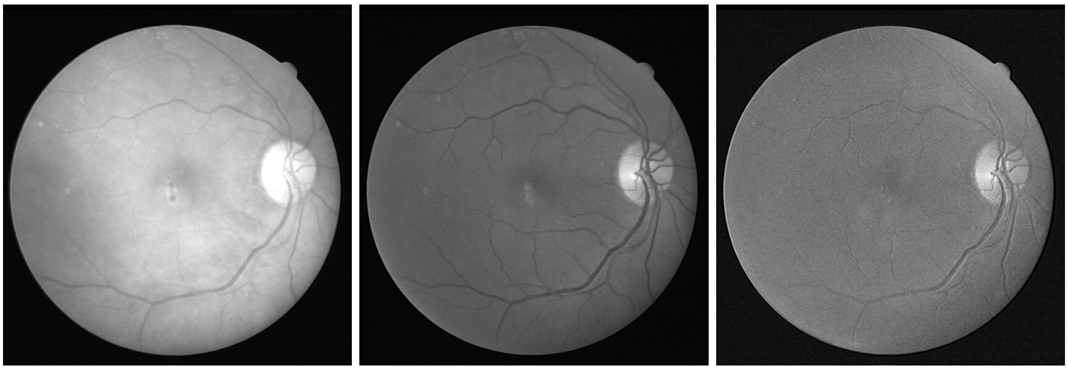
Experimental Evaluation and Results

In this chapter we present the quantitatve and qualitative evaluations of our learning algortighms as described in the last chapter.All the experiments are performed on the green channel of the image as it has the maximum contast between the vessel and background as can be seen in Fig 4.1.

Ther performance of the proposed vessel segmentation algorithm is evaluated using the segmented vaculature considered as the gold standard and the manually marked segmentations by the human observer.All prior work with which we compare our method, have done a segmentation performance analysis with the manual segmentations provided by the first human observer. Additionally for some of the datasets we have a manual segmentation provided by the second observer which can be used to compare the automated methods to that of manual segmentations.

To asses the performance of segmentation methods, various metric have been defined in literature [20, 25]. As reported in prioir works,we also compute the performance of our vessel segmentation model defined as, true positives (TP): number of correctly classified vessel pixels; false positives (FP): number of pixels falsely classified as vessels; true negatives(TN): number of correctly classified non-vessel pixels; false negatives(FN): number of pixels falsely classified as non-vessels.

Using these metrics we can compute the accuracy (number of TP+TN / total pixels) ,pixel classification sensitivity and specificity. We also calculate the area under curve for receiver operation characteristics and also the AUC under the precision recall curve obtained by varying the threshold value for the segmented images.



(a) Red channel (b) Green channel (c) Blue channel

Fig. 4.1 Different channels of a fundus image showing variation in contrast of blood vessels against background. Green channel shows the maximum contrast in blood vessels and background.

For a complete assesment of our segmentation model, we perform various experiments. We start by performing the experiment on all the 4 datasets and compute the performance metrics as described above. Next, to test the generalization of our system we do cross training, i.e, training on images from one dataset and prediction on other.

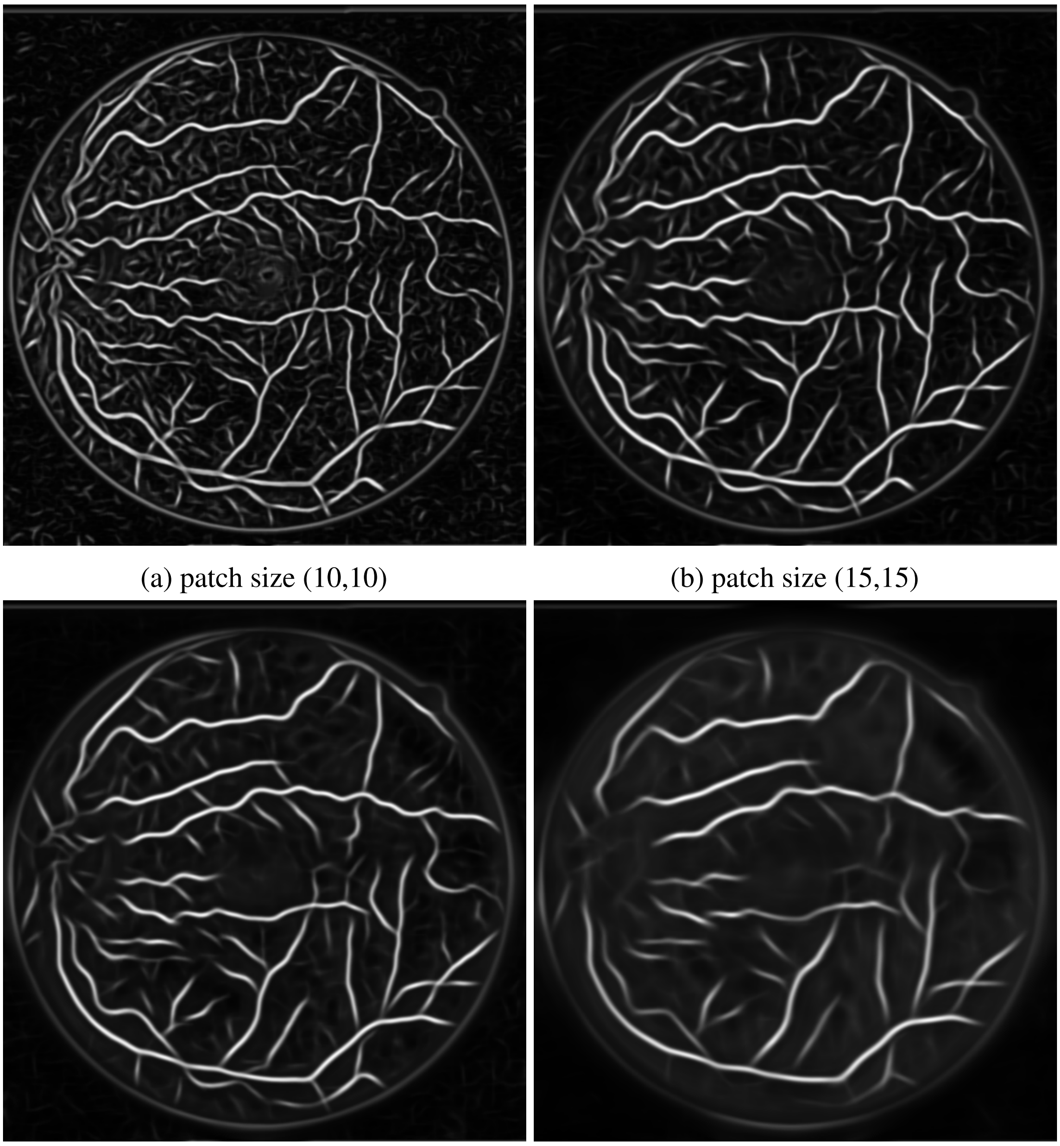
We test the performance of both our models.

4.1 Vessel Segmenetation Assessment

We start by training our model on the DRIVE training datset. The classifier is trained with a patch size of (10,10) and 1000 clusters. We also run the algorithm with the various preprocessing settings as in sections.Contrast enhancement of the images didn't have a major effect on our performance.

We applied our model with varying patch sizes ranging from (10,10) to (21,21). We note that with increase in patch size, we lose details on the thin vessels, but our confidence on thick vessels increases. We combine the results on various patch sizes by simple averaging of all the cases. This had a slight imporvement over our results.

Results for our model applied on DRIVE test datset are shown in the table. As we obtain an average value at each pixel by merging the ground truth annotation coming from different overlapping patches, we can threshold the intensities to obtain the binar segmentation.ROC and PRC plots are made by varying the thresholds in steps of 0.02.



(a) patch size (10,10) (b) patch size (15,15)

(c) patch size (21,21) (d) patch size (33,33)

Fig. 4.2 Here we show the effect of varying patch size in our patch based framework. As we increase the patch size, we lose details on thin vessels,but our confidence on thick vessess increases.

Chapter 5

Conclusion and Future Work

References

[1] Calvo, D., Ortega, M., Penedo, M. G., and Rouco, J. (2011). Automatic detection and characterisation of retinal vessel tree bifurcations and crossovers in eye fundus images. *Computer methods and programs in biomedicine*, 103(1):28–38.

[2] Chutatape, O., Zheng, L., and Krishnan, S. (1998). Retinal blood vessel detection and tracking by matched gaussian and kalman filters. In *Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual International Conference of the IEEE*, volume 6, pages 3144–3149. IEEE.

[3] Doi, K. (2007). Computer-aided diagnosis in medical imaging: historical review, current status and future potential. *Computerized medical imaging and graphics*, 31(4):198–211.

[4] Dollár, P. and Zitnick, C. L. (2013). Structured forests for fast edge detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1841–1848. IEEE.

[5] Donoho, D. L. (2006). Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306.

[6] Farnell, D. J., Hatfield, F., Knox, P., Reakes, M., Spencer, S., Parry, D., and Harding, S. (2008). Enhancement of blood vessels in digital fundus photographs via the application of multiscale line operators. *Journal of the Franklin institute*, 345(7):748–765.

[7] Fraz, M. M., Remagnino, P., Hoppe, A., Uyyanonvara, B., Rudnicka, A. R., Owen, C. G., and Barman, S. A. (2012). Blood vessel segmentation methodologies in retinal images–a survey. *Computer methods and programs in biomedicine*, 108(1):407–433.

[8] Ganin, Y. and Lempitsky, V. (2014). N4-fields: Neural network nearest neighbor fields for image transforms. *arXiv preprint arXiv:1406.6558*.

[9] Gardner, G., Keating, D., Williamson, T., and Elliott, A. (1996). Automatic detection of diabetic retinopathy using an artificial neural network: a screening tool. *British journal of Ophthalmology*, 80(11):940–944.

[10] Huang, K. and Aviyente, S. (2006). Sparse representation for signal classification. In *Advances in neural information processing systems*, pages 609–616.

[11] Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., and Wu, A. Y. (2002). An efficient k-means clustering algorithm: Analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):881–892.

[12] Lee, H., Battle, A., Raina, R., and Ng, A. Y. (2006). Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808.

[13] Lewicki, M. and Sejnowski, T. (2000). Learning overcomplete representations. *Neural computation*, 12(2):337–365.

[14] Lim, J. J., Zitnick, C. L., and Dollár, P. (2013). Sketch tokens: A learned mid-level representation for contour and object detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3158–3165. IEEE.

[15] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

[16] Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2009a). Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696. ACM.

[17] Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11:19–60.

[18] Mairal, J., Ponce, J., Sapiro, G., Zisserman, A., and Bach, F. R. (2009b). Supervised dictionary learning. In *Advances in neural information processing systems*, pages 1033–1040.

[19] Mnih, V. and Hinton, G. E. (2012). Learning to label aerial images from noisy data. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 567–574.

[20] Monteiro, F. C. and Campilho, A. C. (2006). Performance evaluation of image segmentation. In *Image Analysis and Recognition*, pages 248–259. Springer.

[21] Nguyen, U. T., Bhuiyan, A., Ramamohanarao, K., and Park, L. A. (2011). An effective supervised framework for retinal blood vessel segmentation using local standardisation and bagging. In *Machine Learning in Medical Imaging*, pages 117–125. Springer.

[22] Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE.

[23] Ricci, E. and Perfetti, R. (2007). Retinal blood vessel segmentation using line operators and support vector classification. *Medical Imaging, IEEE Transactions on*, 26(10):1357–1365.

[24] Rigamonti, R. and Lepetit, V. (2012). Accurate and efficient linear structure segmentation by leveraging ad hoc features with learned filters. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, pages 189–197. Springer.

[25] Sharma, M. (2001). Performance evaluation of image segmentation and texture extraction methods in scene analysis.

[26] Sinthanayothin, C., Boyce, J., Williamson, T., Cook, H., Mensah, E., Lal, S., and Usher, D. (2002). Automated detection of diabetic retinopathy on digital fundus images. *Diabetic medicine*, 19(2):105–112.

[27] Staal, J., Abràmoff, M. D., Niemeijer, M., Viergever, M. A., and van Ginneken, B. (2004). Ridge-based vessel segmentation in color images of the retina. *Medical Imaging, IEEE Transactions on*, 23(4):501–509.

[28] Tolias, Y. A. and Panas, S. M. (1998). A fuzzy vessel tracking algorithm for retinal images based on fuzzy clustering. *Medical Imaging, IEEE Transactions on*, 17(2):263–273.

[29] Tropp, J. A. and Gilbert, A. C. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666.

[30] Vega, R., Guevara, E., Falcon, L. E., Sanchez-Ante, G., and Sossa, H. (2013). Blood vessel segmentation in retinal images using lattice neural networks. In *Advances in Artificial Intelligence and Its Applications*, pages 532–544. Springer.

[31] Vlachos, M. and Dermatas, E. (2010). Multi-scale retinal vessel segmentation using line tracking. *Computerized Medical Imaging and Graphics*, 34(3):213–227.