Automated Image Segmentation for Retinal Images of Subjects Diagnosed with Choroideremia Disorder

Mina Abdi Oskouie
Department of Computer Science
University of Alberta
abdiosko@ualberta.ca

Ali Jahani Amiri
Department of Computer Science
University of Alberta
Jahaniam@ualberta.ca

Mohammad Motallebi
Department of Computer Science
University of Alberta
motalleb@ualberta.ca

Mohammad Riazi
Department of Computer Science
University of Alberta
riazi@ualberta.ca

Mansour Saffar. M
Department of Computer Science
University of Alberta
saffarme@ualberta.ca

Abstract

Choroidermia is a scarce inherited disorder that leads to complete blindness due to cell loss in the retina. Gene therapy has been proposed as a potential treatment. To examine the effectiveness of the treatment, the size of the impairment islands should be tracked carefully through a series of images from the defected retina. Manual segmentation and use of classical image processing techniques does not result in an accurate nor an ideal segmentation. Hence, automated segmentation methods that could accurately delineate the healthy and non-healthy parts is desirable. In this project, we have investigated several promising machine learning algorithms based on similar previous research and have tried to make some progress towards overcoming the aforementioned complications. We have used the SVM, Random Forest and LDA classifiers along with U-Net. The best performance was achieved by Random Forest segmenting images with 54.09% Jaccard score.

1 INTRODUCTION

Choroideremia (CHM) is an inherited vision impairment, starting from early childhood with "night blindness" symptom. The disorder is due to the cell loss in the retina and the network of blood vessels around it. The dead cells create some islands that grow over time leading to complete blindness. The position and progression of the islands are different among defected individuals [3].

There is no proven therapy for Choroideremia yet. A new treatment, Gene Therapy, is proposed for this disorder which is now in Human Clinical Trials in the United States, the United Kingdom, and Canada [1]. To examine the efficiency of the treatment, the size of the impairment islands should be tracked carefully. For this matter, images are taken from defected retina within a specific time interval. It is crucially important to distinguish the infected and healthy parts accurately.

Currently, the segmentation of healthy retina is done manually by experts. This manual delineation introduces a measurement bias which is not desired in a clinical trial. Hence, a segmentation method or an algorithm that is completely automated is desirable.

However, the challenge in an automated segmentation method is in high variability between the appearance of the islands specific to each subject. In addition, usually the dark parts in the islands and blood vessels crossing them complicate the segmentation and the identification of distinct edges. This leads to a poor or faulty prediction of the shape of the island. Thus, simple image processing techniques would not result in an ideal segmentation.

In this work, we apply different machine learning techniques in order to discriminate defected areas from the healthy ones automatically. The main purpose is to segment the images in the most accurate way. Our project can be divided into three phases: feature extraction, feature selection, and classification. In the first phase, we extract 467 features from images by eight manual methods (pixel intensity value, Histograms, Difference of Gaussians (DoG) and Laplacian of Gaussian (LoG) filters, MR8 filter bank, and density-based clusters, and closeness to the center) and two automatic methods (pre-trained Convolutional Neural Network (CNN) on ImageNet, and dictionary learning). LDA method is used for feature selection. Finally, we segment images using four classifiers (SVM, Random Forest, U-Net and LDA).

The rest of this report is organized as follows: In section 2, we describe previous works related to our project. We formulate the problem as a machine learning task in section 3. The methods we have used are explained briefly in section 4. We present our results and discuss them in section. Finally, we conclude in section 6 and give some directions for future research in section 7. Also at the end, we provide some sample results in appendix.

2 Related Work

Retinal image segmentation can be considered as a subarea of *medical* image segmentation. Many research endeavors are devoted to this area. The efforts can be divided into three steps: 1) Extracting informative features for each pixel of the given images, 2) Selecting the most discriminating features and 3) Classifying pixels using the given features.

2.1 Feature Extraction

Retinal images are in high resemblance with some other medical images (*e.g.* MRI images of brain and mammograms). Feature extraction techniques which are used to segment these images seem to be useful in our work as well. In different works, features are extracted from medical images manually or automatically:

2.1.1 Manual Feature Extraction

Filtering: Locating edges and blobs in the image can give us an important clue about different areas of the image. Knowing the intensity of the neighborhood can help algorithms to find edges and blobs. Applying filters gives us information about the intensity of a pixel together with the density in the neighborhood. Blob detection filters like DoG and LoG are widely used in medical image segmentation (*e.g.* [30] and [9] use DoG filter for detecting mammographic regions.)

Histograms: Our challenge is to classify pixels into healthy and infected groups. Given the fact that every pixel is neighbor to some other pixels, it is reasonable to use context-rich features to take into account the information of neighboring pixels. Najafi *et al.* [28] used histogram analysis to extract statistical features of patches of multi-spectral MRI images of patients diagnosed with Glioblastoma, a malignant brain tumor, to assess the efficiency of a possible treatment. They showed that the histogram based feature extraction can provide discriminative features in the task of assessing the tumor context. We decided to use the histogram analysis on our retinal images due to the similarity of our tasks in which in both the infected area is needed to be distinguished from the healthy background.

MR8 filter bank: A linear filter is an N by N matrix that is convolved with an image to produce a new one. We can convolve multiple filters with an image to produce a bundle of new images. The aforementioned collection of filters is called a filter bank. As mentioned in the previous section, using different types and orientations of filters can reveal and extract important information about various areas of an image. For example, rotating a specific filter, *i.e.* Prewitt filter, can detect edges

at different angles, or detect edges at different scales *e.g.* coarse, medium, and fine. Hence, we can shrink the number of non-zero entries in the Prewitt filter.

The MR8 filter bank consists of 38 filters including edge filters at six different orientations and three scales (18 filters), bar filters at six different orientations and three scales (18 filters), a Gaussian filter and a Laplacian of Gaussian filter. Convolving the MR8 filter bank with an image produces a feature vector with 38 entries, from which we take the maximum response, hence the term MR, over the orientations.

$$\vec{v} = (e_{coarse}, e_{medium}, e_{fine}, b_{coarse}, b_{medium}, b_{fine}, v_{37}, v_{38}) \quad (1)$$

Our motivation for considering the MR8 filter bank is its extensive use in many segmentation tasks especially in medical segmentation tasks[29, 23, 36].

Density based clusters: Another way to detect blobs in the image is to make different clusters in it, using clustering algorithms. Due to the shape of the infected area, density-based clustering algorithms were chosen. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [15] is the most famous density-based algorithm in use. There have been studies on using this method in either general image segmentation [39] or more specifically in medical image segmentation [26]. However, few works have used clustering algorithms for feature extraction. [40] Yu et. al used another clustering algorithm for feature extraction in wood knot images which are similar to the infected area in our images. In this project, we use implementation of DBSCAN method provided in [2] to extract features from images.

2.1.2 Automatic Feature Extraction

Pre-trained CNN on ImageNet: ImageNet is a public dataset of 1.2 million labeled images in 1000 categories. Many people use transfer learning using the deep convolutional neural networks trained on ImageNet dataset in order to fine-tune the network based on their own dataset. We decided to use the trained patches in the first and second convolutional layers of VGG network[11], which is one of the best trained CNNs on ImageNet data.

Dictionary learning: Dictionary learning is a method of finding a specific representation of the data which can be used for automatic feature learning [25]. Kiros et al.[22] used stacked multiscale dictionary learning to create a domain-independent medical image segmentation method. Their method for dictionary learning for medical image segmentation is similar to our task for segmenting infected islands in retinal images, and therefore, it is reasonable to use this automatic feature learning method in our project.

2.2 Feature Selection

Applying techniques to select the best features and eliminate the irrelevent or redundant ones can reduce the size of the data without losing any important information. Working with this trimmed data helps classifiers to work more efficiently. Generally, there are two types of feature selection methods namely supervised and unsupervised.

Both types of techniques are widely used in medical image processing. For instance, [14, 12] and [7] benefit from supervised techniques to find the best features extracted from neuroimages while [38, 17] and [24] apply one unsupervised feature selection technique, Principle Component Analysis (PCA), for this matter. Although there are a number of empirical studies comparing different feature reduction approaches, ([13, 27]), none of these studies claims any technique as the best. Observing the good performance of PCA on neuroimage processing, we applied a feature selection method, Linear Discriminant Analysis (LDA), which is closely related to PCA. This technique will be explained further in section 4.

2.3 Classification

Random Forest and Bagging- Ensemble Method: One of the methods used for classification is Random Forest. Criminisi et al.[18] in their research at Microsoft obtained a significant result using this method for Multiple Sclerosis (MS) lesion segmentation. Although retina and MRI segmentation share similar challenges *i.e.* lesion shapes are variable and their location varies across patients,

there are some differences between these two tasks; firstly, image segmentation in retina is challenging and more difficult in the way that veins and vessels cross the retina; secondly, different type of imaging (*i.e.* MRI vs Chloro-fluorescence) is used to study the images.

SVM: There have been many works in the field of medical image segmentation that have made use of Support Vector Machine (SVM)[37]. Ricci and Perfetti[34], Osareh and Shadgar[31] used SVM to detect blood vessels in color retinal images. They used linear and Radial Basis Function (RBF) kernels respectively. This serves as a motivation for us to use SVM for classifying images in our project.

U-Net: U-Net is a Convolutional Neural Network (CNN) which was recently designed to segment medical images [35]. The network has performed remarkably well in recent challenges. In ISBI cell tracking challenge 2015, U-Net outperformed other competitors by a large margin. These outstanding results on similar tasks convinced us to investigate the performance of this method on our data.

3 Problem Formulation

To segment retinal images, we have defined the segmentation problem as a pixel-wise classification. This means, each pixel of the given image is considered as a new observation. We extract features for each pixel and our classifiers should label each pixel as healthy or infected.

Our data consist of retinal images of 16 patients. Images are taken from each eye of an individual within specified time intervals. There are between 3 to 10 images for each subject. So in total, we have 111 images for training and testing with resolution of 768×768. We trained the classifiers using images of 12 subjects and tested them with images of 4 different patients.

Inspired by other works in medical image segmentation i.e [8] and [33], to evaluate the performance of our classifiers, we have used Jaccard similarity coefficient [20] as shown below:

$$J(G, S) = \frac{|G \cap S|}{|G \cup S|}, \tag{2}$$

where G is our ground truth of an image (which includes the labels of all pixels in that image) and S is the segmented image. $J(G, S)$ takes values between zero and one; where one shows that the segmentation is perfectly accurate.

4 Method

4.1 Feature Extraction

4.1.1 Manual Feature Extraction

Table 1 shows a list of features we have extracted as our manual features:

Feature number	1	2:5	6	7:8	9	10:17	18:19
Feature	Closeness to the centre	Histogram	LoG	DoG	Pixel intensity	MR8	Density-based clustering

Table 1: Manual features

The most trivial feature of each pixel is its intensity value. We consider this value as one of our features. Also, since islands are located mainly in the center of the image, we added a feature named closeness of the pixel to the center of the image.

Histogram: We performed histogram analysis on every image in our dataset. For every pixel in an image, we extracted an 11x11 patch centering at that specific pixel, and then computed the mean, variance, skewness, and kurtosis of the intensity histogram of the pixels in that patch.

Density-based Clustering: In order to extract the density-based feature, DoG is initially computed over the image. Next, using Otsu's method [32] a threshold is calculated. Given this threshold, any pixel above the threshold will be considered as a valid data point and pixels below the threshold are discarded. Then, DBSCAN is run over these data points. We selected the minimum points setting a cluster to be 30 and epsilon (maximum distance between two points to be considered in one cluster) equal to 5 since we would be able to detect different blobs in the image so they are not combined together. Thereafter, the maximum intensity of all pixels in each blob is assigned to all other pixels in the cluster. Finally, the value of each pixel is multiplied by the closeness of the closest pixel in the blob to the center of the image. The last step is performed to get higher value for blobs closer to the center. This is because of the fact that infected areas are closer to the center of the image than healthy areas.

4.1.2 Automatic Feature Learning

Dictionary Learning: We used the Orthogonal Matching Pursuit (OMP) algorithm to train a sparse dictionary with our dataset. In OMP algorithm we aim to solve the following optimization problem:

$$\begin{aligned} \min_{D, x_{(i)}} \quad & \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2} ||Dx^{(i)} - \alpha^{(i)}||_2^2, \\ subject \quad to \quad & ||D^{(c)}||_2^2 \leq 1 \quad , \forall c \\ & ||\alpha^{(i)}||_0 \leq 1 \quad , \forall i \end{aligned} \tag{3}$$

We created a Gaussian pyramid (up to level 5) of every image in our training set and then extracted 11x11 patches with a stirde of 5 from images of each level. Patch-wise mean reduction and varinace-contrast normalization was done to remove the effect of different illuminations of images. The leftmost picture of Figure 2 shows the 128 dictioanry elements that were learned from our training dataset using the OMP sparse dictionary learning method.

Pre-trained CNN on ImageNet: We used the learned patches from the first and the second convolutional layers of VGG network [11] to extract more features from our images. The learned patches of the first and the second convolutional layers of VGG network are shown in the middle and the rightmost pictures of Figure 2 respectively.

4.2 Feature Selection

4.2.1 Linear Discriminant Analysis (LDA)

LDA tries to reduce the dimensionality of data while keeping the most possible information. Unlike PCA, not only it considers variance, but also tries to minimize within-class scatter while maximizing the between-class scatter. This technique can be used either as a linear classifier or as a feature selection method [16]. We used LDA in both cases using the implementation in [4].

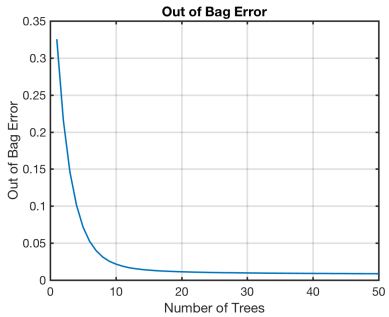

Figure 1: OOB error

4.3 Classification

Random Forest and Bagging - Ensemble Method: Our detection and segmentation problem can be seen as a binary classification of every pixel as either healthy or infected. This classification can be addressed by a supervised method such as the decision tree. However decision trees are recognized to suffer from over-fitting[6].

Bootstrap aggregation or Bagging[19, Ch.15] is a simple yet powerful ensemble method that incorporates the predictions from various machine learning algorithms to obtain more accurate predictions rather than a prediction from a single model. Since decision trees contain high variance, bagging which is a Bootstrap procedure could be used to remedy this issue. When applying bagging to decision trees, the only parameter required is the number of samples and the number of trees to be included.

The implementation of the Random Forest classifier [5] we have used is done in Matlab using the TreeBagger ensemble method. The training process of a TreeBagger is as described below:

- Create many random sub-samples of our data set with replacement
- Grow *n* trees
- Inspect how the ensemble error changes with aggregation of trees
- Estimate feature importance

By default, Matlab's TreeBagger for classification sets the minimal leaf size to 1 and randomly selects the square root of the total number of features for each decision split [21, Ch.8]. For each bootstrap sample selected from the training data, there are left-out samples which are not included in the training set. These samples are called Out-Of-Bag samples or OOB.

We can calculate the estimated accuracy of the bagged models through averaging the performance of each model on its left-out samples. This approximate accuracy is called the OOB estimate of performance as shown in Figure 1.

Next, we would like to find out the most important features that could further increase the accuracy of our classifier. With TreeBagger in Matlab this is simply done by turning on the feature importance measure (oobvarimp), and plotting the results. Using these rankings we can now run the classifier with the most important features and a larger number of trees to speed the training process.
Finally, the produced model is given to the predict function with our test data to classify the unlabeled data.

SVM: We can use SVM as a classifier for each pixel. SVM tries to classify by finding maximum boundaries while minimizing classification error. We used libSVM in MATLAB that was suggested by Chang and Lin[10]. We tried RBF kernels as was used by earlier works.

U-Net: In this work we use the code of U-Net provided by Olaf Ronneberger [1] The network is trained using stochastic gradient descent implementation of Caffe. The network is consists of 10 layers. In the first half of them filter convolution and max pooling are applied alternatively. For the second half, operations are alternatively filter and up convolution.

5 Results & Discussion

As mentioned before, we are provided with retinal images of 16 patients which were collected within a specific time interval. At first, We used images of 12 patients (82 images) for train and 4 patients (29 images) for test. Thinking about each pixel as an observation, we had about 49 million observations for training and 16 million samples for testing. Our limited resources made this task impossible for use (to the extend that even loading data in Matlab took about 5 minutes!). To solve this problem, we had to reduce the data. We used only two images of each patient for training and testing.

Before applying any machine learning techniques, we needed to pre-process our data to maintain consistency among them. Firstly, we had images with different sizes. So, we resized all the images to 768×768. Secondly, some images were much brighter than the others. We applied three methods, mean subtraction, patch-wise centering, and variance-contrast normalization, to achieve images with the same illumination. Another problem we were faced with was that the number of healthy pixels in images were much more than infected ones. Therefore, giving all the pixels to the classifiers could make them biased. To solve this problem, we needed approximately same number of healthy and

[1] http://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

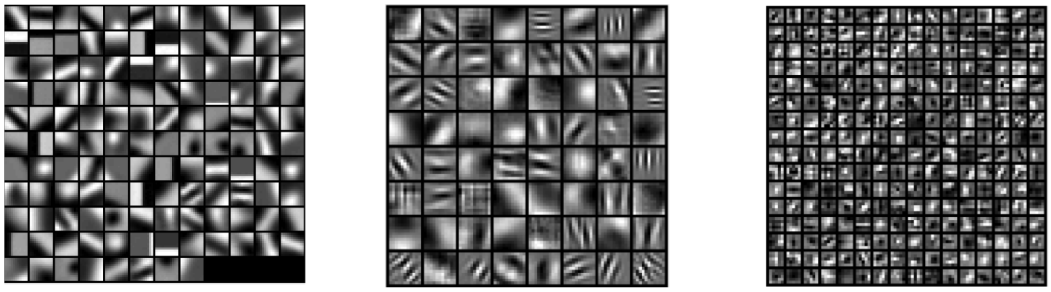

Figure 2: From left to right: 1) Dictionary Learned features with patch size of 11×11, 2) Features found by Pre-trained CNNs On ImageNet Dataset (64 filters of size 11×11 in the 1st convolutional layer), 3) Features found by Pre-trained CNNs On ImageNet Dataset (256 filter of size 5×5 in the 2nd convolutional layer)

infected pixels. Hence, we only selected n healthy pixels randomly from each image, where n is the number of infected pixels in that image.

As it was explained in section 4, we applied different techniques to extract 467 features for each pixel. In the next step, we used a well-known dimension reduction method, LDA, to project our features to a space with less dimensions. The weights LDA considered for each feature shows the contribution of that feature in creation of the new feature. In other words, the larger the weight that LDA considers for a feature, the more informative that feature is. The results of the LDA shows that manually extracted features are more informative than the other features. This sounds surprising for us, but figure 2 shows that this is reasonable. Figure 2 shows the automatically extracted features are similar to edge and blob detection filters. As a result, LDA finds them correlated to manual ones and make them less effective on resulting features.

In the next step, we used U-Net to classify our data. Unfortunately, not all needed files were provided. We only had access to pre-trained network. This network was trained on data from ISBI cell tracking challenge. We classified our test data by this pre-trained network to investigate the results and run it over 220k iteration. However, the result was not promising to the extend that after 20k iterations, the output was a pure white image. These results are provided in table 2. The underlying reason for these poor results can be because of the difference in training data, cell tracking data and retinal images.

At the end, we examined two classifiers, SVM and Random Forest, using the original data, 467 features, and the LDA's results. The question we came up with was whether applying LDA is helpful or not. To answer this question, we provided all classifiers with both original data and LDA's output. As table 3 suggests, dimensionality reduction using LDA deteriorates the performance of the Random Forest (54.09% Vs. 33.48%). In SVM with RBF kernel, trained with LDA's results, almost all training data were considered as a support vector point. This shows that the model will not give us any promising result; Therefore, we did not continue testing our data with this approach.

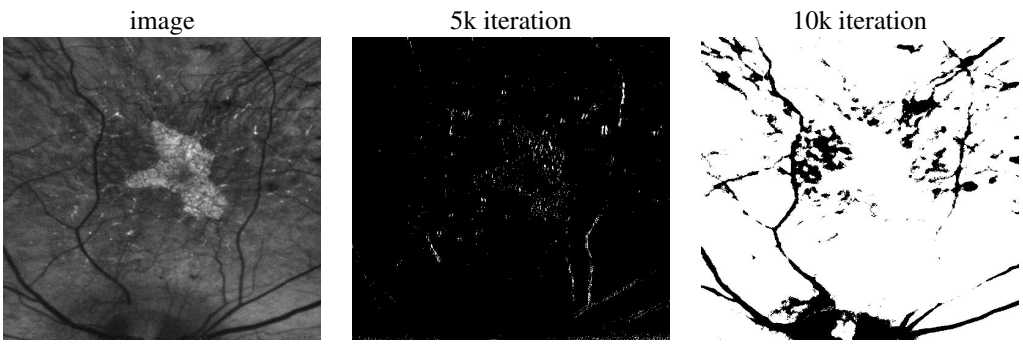

Table 2: Results from U-Net after some iteration

At the end, we tested the performance of all three classifiers (Random Forest, SVM with RBF kernel, and LDA) using the original data (without any feature selection). Results are demonstrated in table 3. According to results, Random Forest outperformed two other classifiers followed by SVM with RBF kernel. Some samples of the test images are provided in appendix.

It is not surprising to see that LDA shows the worst performance since it consists of LDA feature selection and a simple linear classifier. As we discussed in the previous paragraph, classifiers like SVM and Random Forest cannot achieve promising results using LDA feature selection data, hence, we cannot expect a simple linear classifier to perform well under these circumstances.

Method	**Average Jaccard**
Random Forest with LDA	33.48%
Random Forest without LDA with manual features	54.09%
SVM RBF without LDA (gamma =0.75, with 19 manual features)	45.22%
LDA	32.29%

Table 3: Comparing average results of different classifiers

6 Conclusion

Our project has focused on the task of image segmentation of the impairment island in patients diagnosed with Choroidermia disorder. The data were provided by the Department of Medical Sciences at the University of Alberta. Both automatic and manual feature extractions were applied to the images resulting in 467 features for every pixel of each image. Since our task was a pixel-wise image segmentation, the volume of data we were confronted with accumulated to a gigantic amount which suggests the need for feature conditioning. Linear Discriminant Analysis was performed to reduce the dimension of our data and different linear classifiers (SVM, LDA) as well as non-linear classifiers (Random Forest, UNet) were applied on both original feature sets (467) and the conditioned data. Our experiments yield that the classifiers did not produce a comparative result considering the conditioned features set so we have rolled back to using the original 467 features. Final results show that the Ensemble Random Forest classifier outperforms the rest of the classifiers with a total average Jaccard score of %54.09.

7 Future Work

Despite novel approaches we have taken in this project, still there are plenty of improvements that can be done over this segmentation task. As mentioned earlier, images are taken from each patient within specific time intervals. Since retinal structure is relatively consistant over time, they are similar in all images of one patient. Hence, this information can be taken into account to make the segmentaton more accurate.

References

[1] Choroideremia research foundation. http://curechm.org/what-is-choroideremia/. Accessed: 2016-12-8.

[2] Density-based spatial clustering of applications with noise (dbscan). http://yarpiz.com/255/ypml110-dbscan-clustering. Accessed: 2016-12-8.

[3] Genetics home reference. https://ghr.nlm.nih.gov/condition/choroideremia/. Accessed: 2016-12-8.

[4] Linear Discriminant Analysis, howpublished = http://yarpiz.com/430/ypml114-linear-discriminant-analysis, note = Accessed: 2016-12-8.

[5] Random Forest example. https://davidlary.info/blog/2015/11/25/1760/. Accessed: 2015-11-25.

[6] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[7] Sara Calderoni, Alessandra Retico, Laura Biagi, Raffaella Tancredi, Filippo Muratori, and Michela Tosetti. Female children with autism spectrum disorder: an insight from mass-univariate and pattern classification analyses. *Neuroimage*, 59(2):1013–1022, 2012.

[8] Rubén Cárdenes, Rodrigo de Luis-García, and Meritxell Bach-Cuadra. A multidimensional segmentation evaluation for medical image data. *Computer methods and programs in biomedicine*, 96(2):108–124, 2009.

[9] David M Catarious Jr, Alan H Baydush, and Carey E Floyd Jr. Characterization of difference of gaussian filters in the detection of mammographic regions. *Medical physics*, 33(11):4104–4114, 2006.

[10] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[11] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[12] Carlton Chu, Ai-Ling Hsu, Kun-Hsien Chou, Peter Bandettini, ChingPo Lin, Alzheimer's Disease Neuroimaging Initiative, et al. Does feature selection improve classification accuracy? impact of sample size and feature selection on classification using anatomical magnetic resonance images. *Neuroimage*, 60(1):59–70, 2012.

[13] R. C. Craddock, P. E. Holtzheimer, X. P. Hu, and H. S. Mayberg. Disease state prediction from resting state functional connectivity. *Magnetic Resonance In Medicine: Official Journal Of The Society Of Magnetic Resonance In Medicine/Society Of Magnetic Resonance In Medicine*, 62:1619–1628, 2009.

[14] Zhengjia Dai, Chaogan Yan, Zhiqun Wang, Jinhui Wang, Mingrui Xia, Kuncheng Li, and Yong He. Discriminative analysis of early alzheimer's disease using multi-modal imaging and multi-level characterization with multi-classifier (m3). *Neuroimage*, 59(3):2187–2195, 2012.

[15] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[16] Ronald A Fisher. The statistical utilization of multiple measurements. *Annals of eugenics*, 8(4):376–386, 1938.

[17] K Franke, G Ziegler, S. Kloppel, and C. Gaser. Estimating the age of healthy subjects from t1-weighted mri scans using kernel methods: exploring the influence of various parameters. *NeuroImage*, 50:883–892, 2010.

[18] Ezequiel Geremia, Olivier Clatz, Bjoern H Menze, Ender Konukoglu, Antonio Criminisi, and Nicholas Ayache. Spatial decision forests for ms lesion segmentation in multi-channel magnetic resonance images. *NeuroImage*, 57(2):378–390, 2011.

[19] Trevor J.. Hastie, Robert John Tibshirani, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2011.

[20] Paul Jaccard. The distribution of the flora in the alpine zone. *New phytologist*, 11(2):37–50, 1912.

[21] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 6. Springer, 2013.

[22] Ryan Kiros, Karteek Popuri, Dana Cobzas, and Martin Jagersand. Stacked multiscale feature learning for domain independent medical image segmentation. In *International Workshop on Machine Learning in Medical Imaging*, pages 25–32. Springer, 2014.

[23] Mayuresh Kulkarni and Fred Nicolls. Interactive image segmentation using graph cuts. *Pattern Recognition Association of South Africa*, pages 99–104, 2009.

[24] M. Lopez, J. Ramirez, J. Garriz, I. Alvarez, D. Salas-Gonzalez, F. Segovia, R. Chaves, P. Padilla, and M. Gomez-Rao. Principal component analysis-based techniques and supervised classification schemes for the early detection of alzheimer's disease. *Neurocomputing*, 74:1260–1271, 2011.

[25] Julien Mairal, Francis Bach, and Jean Ponce. Sparse modeling for image and vision processing. *arXiv preprint arXiv:1411.3230*, 2014.

[26] R Manavalan and K Thangavel. Trus image segmentation using morphological operators and dbscan clustering. In *Information and Communication Technologies (WICT), 2011 World Congress on*, pages 898–903. IEEE, 2011.

[27] Benson Mwangi, Tian Siva Tian, and Jair C Soares. A review of feature reduction techniques in neuroimaging. *Neuroinformatics*, 12(2):229–244, 2014.

[28] Mohammad Najafi, Hamid Soltanian-Zadeh, Kourosh Jafari-Khouzani, Lisa Scarpace, and Tom Mikkelsen. Prediction of glioblastoma multiform response to bevacizumab treatment using multi-parametric mri. *PLoS One*, 7(1):e29945, 2012.

[29] Rodrigo Nava and Jan Kybic. Supertexton-based segmentation in early drosophila oogenesis. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2656–2659. IEEE, 2015.

[30] Samuel Oporto-Díaz, Rolando Hernández-Cisneros, and Hugo Terashima-Marín. Detection of microcalcification clusters in mammograms using a difference of optimized gaussian filters. In *International Conference Image Analysis and Recognition*, pages 998–1005. Springer, 2005.

[31] A Osareh and B Shadgar. Automatic blood vessel segmentation in color images of retina. *Iranian Journal of Science and Technology*, 33(B2):191, 2009.

[32] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[33] Marcel Prastawa, Elizabeth Bullitt, Sean Ho, and Guido Gerig. A brain tumor segmentation framework based on outlier detection. *Medical image analysis*, 8(3):275–283, 2004.

[34] Elisa Ricci and Renzo Perfetti. Retinal blood vessel segmentation using line operators and support vector classification. *IEEE transactions on medical imaging*, 26(10):1357–1365, 2007.

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[36] Mihran Tuceryan, Anil K Jain, et al. Texture analysis. *Handbook of pattern recognition and computer vision*, 2:207–248, 1993.

[37] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[38] Ying Wang, Yong Fan, Priyanka Bhatt, and Christos Davatzikos. High-dimensional pattern regression using machine learning: from medical images to continuous clinical variables. *NeuroImage*, 50:1519–1535, 2010.

[39] Qixiang Ye, Wen Gao, and Wei Zeng. Color image segmentation using density-based clustering. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pages III–345. IEEE, 2003.

[40] Gang Yu and Sagar V Kamarthi. Texture classification using wavelets with a cluster-based feature extraction. In *Systems and Control in Aerospace and Astronautics, 2008. ISSCAA 2008. 2nd International Symposium on*, pages 1–6. IEEE, 2008.

8 Appendix

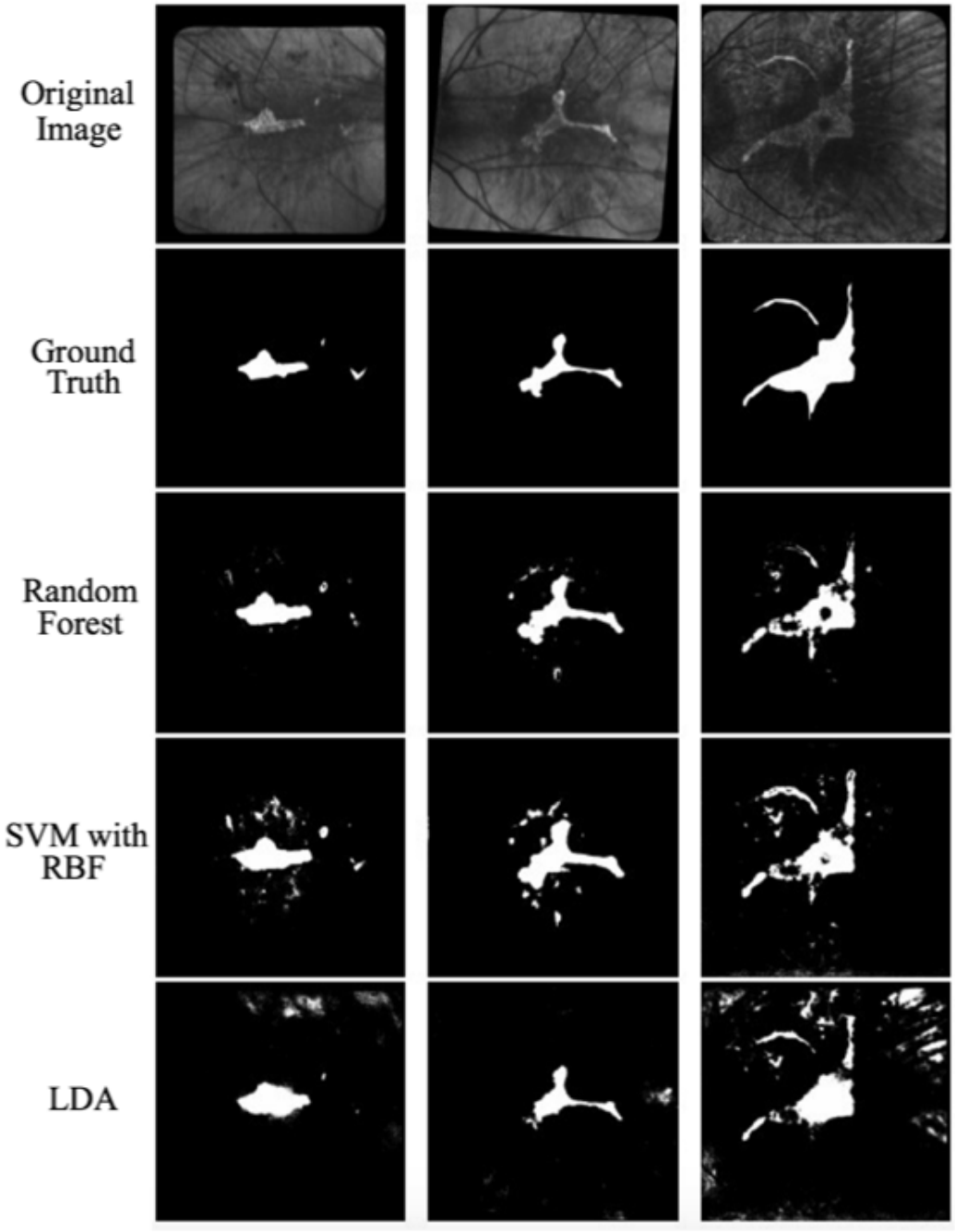

Figure 3: Results of three classifiers on three samples from the test set. All classifiers are trained using data with 467 features.