A Deep Learning Based Pipeline for Image Grading of Diabetic Retinopathy

Yu Wang

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science & Application

Weiguo Fan, Chair

Edward A. Fox

Chandan K. Reddy

May 1, 2018

Blacksburg, Virginia

Keywords: Retina Image, Image Grading, Diabetic Retinopathy, Early Detection, Feature Extraction, ConvNN, Deep Learning, Boosting Decision Tree

A Deep Learning Based Pipeline for Image Grading of Diabetic Retinopathy

Yu Wang

(ABSTRACT)

Diabetic Retinopathy (DR) is one of the principal sources of blindness due to diabetes mellitus. It can be identified by lesions of the retina, namely microaneurysms, hemorrhages, and exudates. DR can be effectively prevented or delayed if discovered early enough and well-managed. Prior image processing studies on diabetic retinopathy typically extract features manually but are time-consuming and not accurate. We propose a research framework using advanced retina image processing, deep learning, and a boosting algorithm for high-performance DR grading. First, we preprocess the retina image datasets to highlight signs of DR, then employ a convolutional neural network to extract features of retina images, and finally apply a boosting tree algorithm to make a prediction based on extracted features. The results of experiments show that our pipeline has excellent performance when grading diabetic retinopathy images, as evidenced by scores for both the Kaggle dataset and the IDRiD dataset.

A Deep Learning Based Pipeline for Image Grading of Diabetic Retinopathy

Yu Wang

(GENERAL AUDIENCE ABSTRACT)

Diabetes is a disease in which insulin can not work very well, that leads to long-term high blood sugar level. Diabetic Retinopathy (DR), a result of diabetes mellitus, is one of the leading causes of blindness. It can be identified by lesions on the surface of the retina. DR can be effectively prevented or delayed if discovered early enough and well-managed. Prior image processing studies of diabetic retinopathy typically detect features manually, like retinal lesions, but are time-consuming and not accurate. In this research, we propose a framework using advanced retina image processing, deep learning, and a boosting decision tree algorithm for high-performance DR grading. Deep learning is a method that can be used to extract features of the image. A boosting decision tree is a method widely used in classification tasks. We preprocess the retina image datasets to highlight signs of DR, followed by deep learning to extract features of retina images. Then, we apply a boosting decision tree algorithm to make a prediction based on extracted features. The results of experiments show that our pipeline has excellent performance when grading the diabetic retinopathy score for both Kaggle and IDRiD datasets.

Acknowledgments

I would like to thank my advisor and my committee members, Prof. Patrick Fan, Prof. Edward A. Fox and Prof. Chandan K. Reddy, for their guidance and support during my studies at Virginia Tech. Their patience, encouragement and insightful suggestions greatly aided my research. I am thankful to all of my group members, Jeff Wu and Ting Zhou, for helping with the diabetic retinopathy grading project, and my colleagues Yufeng Ma, Xuan Zhang, Liuqing Li, Ziqian Song, Siyu Mi for their valuable ideas in deep learning knowledge and encouragement during my research. I also want to thank the Department of Computer Science at Virginia Tech for giving me an opportunity to work in the deep learning area. As always, I would like to thank my family for their unconditional love and support over the years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Diabetes mellitus is a chronic, progressive disease caused by inherited or acquired deficiency in the production of insulin by the pancreas. If blood sugar is not kept in a specific range, some long-term complications of the eyes, feet, and kidneys can start developing quickly. According to the WHO (World Health Organization), at the end of 2014, 422 million people in the world had diabetes – a prevalence of 8.5% among the adult population. Many of the deaths caused by diabetes (43%) occur under the age of 70 [1]. However, with a balanced diet, proper physical activity, immediate medication, and regular screening for complications, diabetes can be treated, and its consequences avoided.

Diabetic retinopathy is a diabetes complication that affects the eyes, triggered by high blood sugar levels. It occurs as a result of long-term accumulated harm to the small blood vessels in the retina and is the leading cause of loss of vision. A sample diabetic retinopathy image

[1] http://apps.who.int/iris/bitstream/handle/10665/204871/9789241565257_eng.pdf

Figure 1.1: Sample Color Retina Image

is shown in Figure 1.1.

People can still live well with diabetes if early diagnosis occurs; the longer a person lives with undiagnosed and untreated diabetes, the worse the health outcomes are likely to be. Easy access to automatic diagnostics for diabetes is therefore essential.

Currently, people mainly use handcrafted methods to detect DR. Doctors can recognize DR by the signs of lesions associated with the abnormalities in blood vessels caused by diabetes. This method is useful, but it requires many resources. The infrastructure required for handling DR is usually short in places where the proportion of diabetes is high, and DR identification is most needed. As the number of people with diabetes continues to grow, the expertise and equipment needed to alleviate sightless caused by DR will become even more inadequate.

Computer-aided diagnosis has recently become more and more popular. One important motivation is the superior performance of deep learning. In many research areas, like the spotting of lung cancer or brain tumors, deep learning has been used to ascertain the severity

Figure 1.2: Diabetic Retinopathy Segmentation and Grading Challenge Logo [6]

level from medical images; it has achieved fantastic performance, even competitive with experienced doctors.

In 2018, grand-challenge.org [2] hosted the Diabetic Retinopathy Segmentation and Grading Challenge (Prasanna Porwal and Meriaudeau [37]), associated with the IEEE International Symposium on Biomedical Imaging (ISBI 2018). Over 700 people joined this challenge, and over 30 teams made submissions. The challenge logo is shown in Figure 1.2.

grand-challenge.org is a platform mainly for grand challenges in the biomedical image analysis area. It aims to help the research community and industry to develop better algorithms. Since 2007, 25 biomedical image analysis challenges have been launched.

The proposed pipeline, which aims at capturing distinctive features related to diabetic retinopathy and making reasonable predictions of the DR severity level, leverages the advantages of deep learning to extract features. It has earned third place in sub-challenge 2 of the ISBI 2018 challenges.

1.2 Problem Statement

Diabetic Retinopathy is a clinical diagnosis that affects eyes, represented by the presence (see Figure 1.3) of several retinal lesions like microaneurysms (MA), hemorrhages (HE), hard

[2] https://grand-challenge.org/

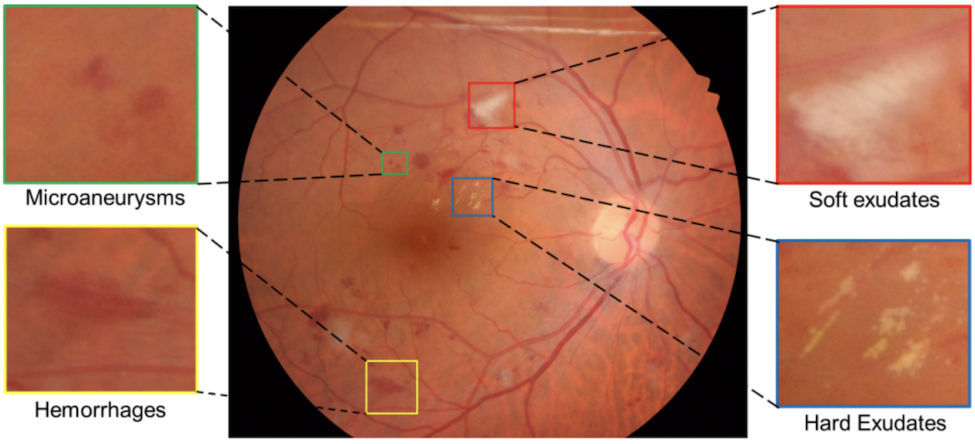

Figure 1.3: Lesions of Diabetic Retinopathy [7]

exudates (EX), and soft exudates (SE).

Based on lesions identified, the DR status of a retina can be classified into two phases, known as NPDR (Non-Proliferative Diabetic Retinopathy) and PDR (Proliferative Diabetic Retinopathy), as shown in Figure 1.4 a and b.

Therefore, success in the diabetic retinopathy grading task mainly depends on how well lesions can be extracted. Lesions extraction considers some very small details, like microaneurysms, and some larger features, such as exudates, and sometimes even their position relative to each other on images of the eye.

For ordinary people, those lesions may be very small and hard to detect because of image noise; thus achieving good results in the grading task would be very difficult. The primary problem this research tries to address is how to extract features of diabetic retinopathy from retina images, especially lesions of DR. This problem can be further decomposed into the following three sub-problems.

First, we try to discover the key features related to the DR grading task. We explore preprocessing methods for retinal images that make the signs of DR more visible. We consider those

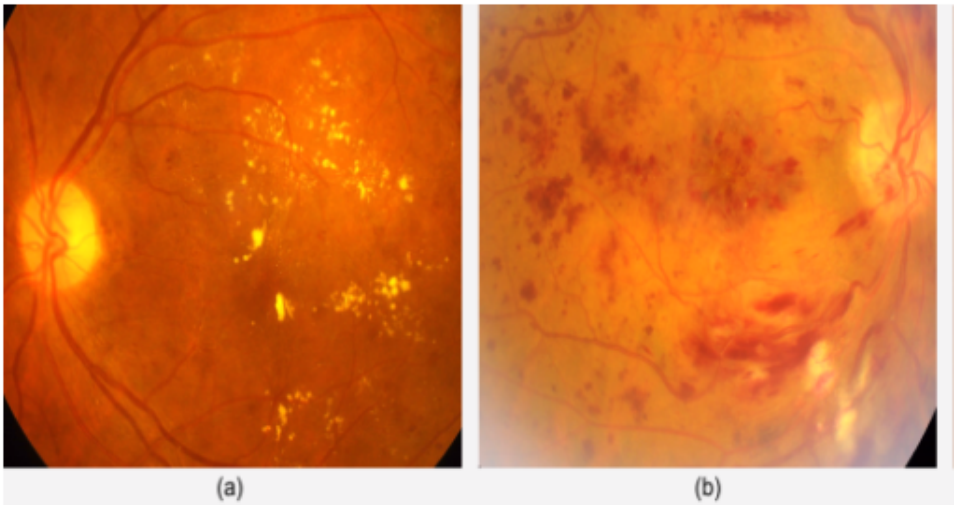

Figure 1.4: NPDR and PDR [8]

preprocessing methods which are commonly used with medical images, especially diabetic retinopathy detection, including morphological closing, contrast enhancement, etc.

Then, we move one step further to identify how to extract features from preprocessed retinal images. Particularly, we experiment with deep learning methods to extract features from retinal images. That is because deep learning can identify higher-level features as its network goes deeper.

Finally, we try to find which model, or which algorithm, is more suitable to predict diabetic retinopathy level. We apply a gradient boosting tree algorithm and neural network to train a model based on the extracted features.

1.3 Hypothesis

The main hypotheses in this research are: First, preprocessing steps are very instrumental in the DR grading task; they can reduce the noise and highlight the signs of DR. Second, our deep learning based approach can extract good features after image preprocessing.

Specifically, DenseNet can achieve better performance due to stronger connectivity between neurons. Third, a gradient boosting decision tree algorithm can generate more accurate results on classifying DR level based on extracted features than neural networks.

In contrast to existing methods of extracting features of diabetic retinopathy, our work preprocesses the retinal image first, then feeds into a convolutional neural network to train and identify the features. Deep learning methods can extract higher-level features in a deep layer, thus capturing what other models would miss.

The use of a gradient boosting tree algorithm in our research is a very new but efficient method, which aims to help us make better use of features, where those features are well extracted.

1.4 Thesis Organization

This research explores methods in grading diabetic retinopathy level. The proposed workflow is developed based on image preprocessing, feature extraction, and model training.

The remainder of this thesis is organized as follows. Chapter 2 reviews the literature in deep learning in medical image analysis and diabetic retinopathy grading. Chapter 3 introduces datasets we have used. Chapter 4 explains the whole process of our experiments. Chapter 5 shows the results of experiments. Chapter 6 discusses some advantages and further work we can do with this workflow. Chapter 7 gives the conclusions.

Chapter 2

Literature Review

This study aims to use deep learning methods to extract DR features from retinal images, and grade diabetic retinopathy based on severity level. It is related to deep learning, medical image analysis, diabetic retinopathy grading, and tree-based algorithms, so the four sections below review some relevant works in those areas.

2.1 Deep Learning

Deep learning, which has been the new research frontier, has gained popularity in many tasks. The main advantage of many deep learning algorithms is that networks composed of many layers, can transform input data (e.g., image) to outputs (e.g., binary value, True or False) while capturing increasingly higher level features. Unlike traditional machine learning methods, in which the creator of the model has to choose and encode features ahead of time, deep learning enables the model to automatically learn features that matter. That is very important because feature engineering typically is the most time-consuming part of machine learning practice.

In the 1940s-1960s, deep learning was referred to as "cybernetics", and known as "connectionism" in the 1980s-1990s. The most recent wave of neural network research began with a break-through in 2006 after Geoffrey Hinton showed that one kind of neural network named deep belief network is able to be trained efficiently using greedy layer-wise pretraining (Hinton et al. [26]). Many other kinds of deep neural networks also applied the same strategy later (Bengio et al. [12]; Poultney et al. [36]). Since then, the term "deep learning" became popular among neural network researchers. Training deeper and deeper neural networks became possible, and researchers began to realize the importance of depth in neural networks (Bengio et al. [13]; Delalleau and Bengio [19]; Pascanu et al. [35]; Montufar et al. [34]).

One crucial reason deep learning took off, is that the dataset size is continuously increasing. Figure 2.1 shows the size of datasets used as classification task benchmarks have significantly increased in size over time.

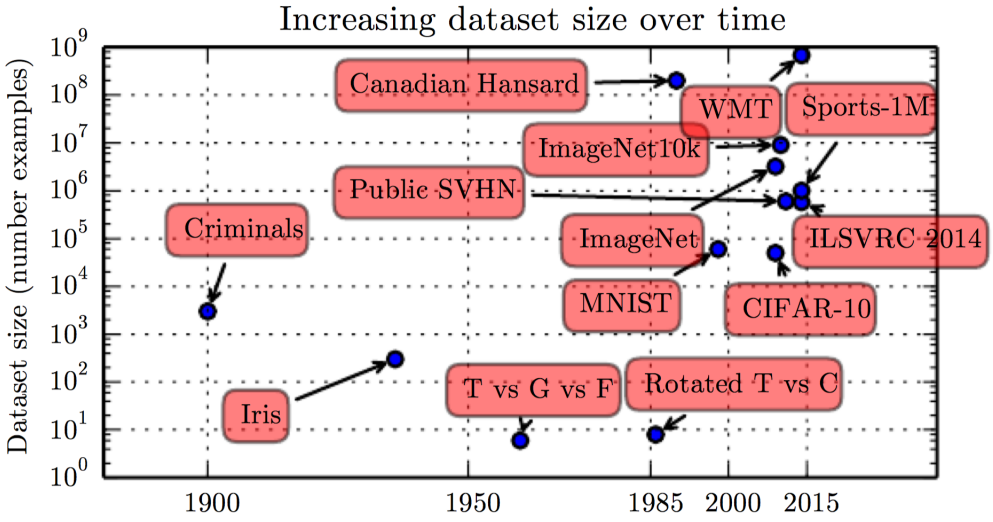

Figure 2.1: Increasing Dataset Size Over Time [23]

On the other hand, computational resources are easier to get which made models run faster and faster. Thus, larger and/or deeper models became possible. Due to more advanced GPUs, faster network connectivity, and better software infrastructure for distributed com-

puting, many neurons can work together and lead to increased accuracy, complexity, and huge impact. That is an important observation, because from biology, researchers noticed that animals become more intelligent when there is stronger neuron connectivity. Figure 2.2 shows the decreased error rate in deep learning over time.

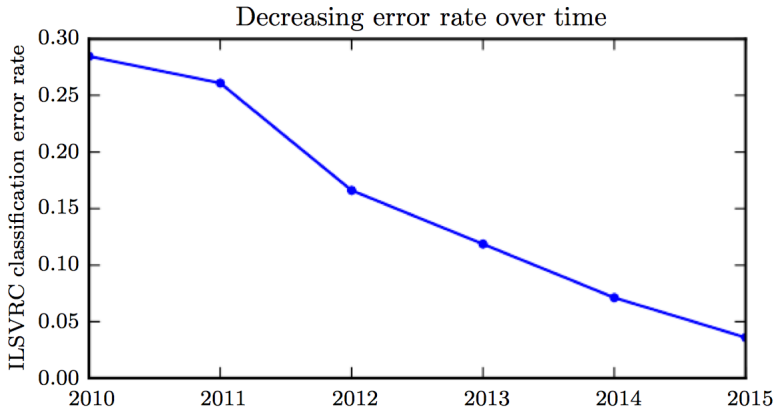

Figure 2.2: Decreasing Error Rate in Deep Learning Over Time [23]

Among deep learning architectures, Convolutional Neural Networks (CNNs) are the state-of-art architectures for many image analysis tasks. The key operations in CNNs are the convolution operation, non-linearity transformation, spatial pooling, and feature maps.

Pioneering work on CNNs includes LeNet-5 (LeCun et al. [31]) for hand-written digit recognition. The architecture of LeNet-5 is shown in Figure 2.3. The authors constructed a 5-layer network, trained on the MNIST digit dataset with 60K training examples. This architecture was embraced by many banks to recognize hand-written numbers on checks loaded in 32x32 pixel images. Handling higher quality images requires larger and more convolutional layers, so the constraint in this technique is the availability of computing resources.

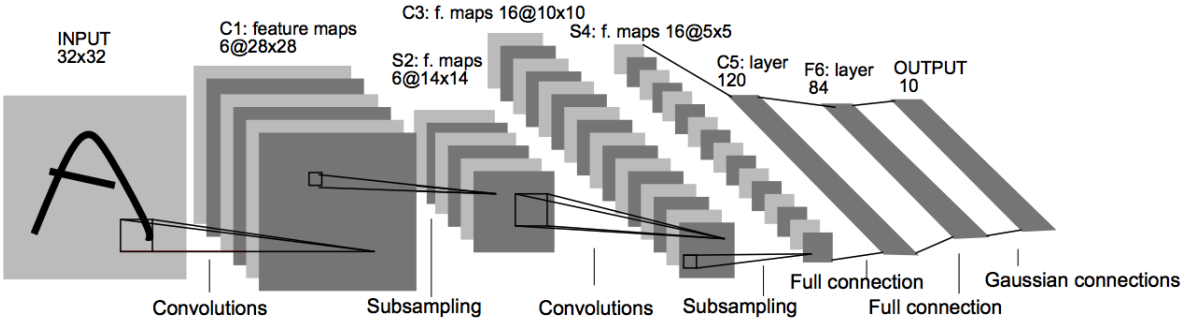

Figure 2.3: Architecture of LeNet-5 [31]

Then, in 2012, AlexNet (Krizhevsky et al. [29]) was invented for image classification. This framework is fairly similar to LeNet, but it applied Max pooling, ReLu nonlinearity, Dropout regularization, and composed a larger model (7 hidden layers, 650k units, 60m parameters). Because of the large model size, this architecture can classify higher resolution images (227*227*3). Figure 2.4 shows the architecture of AlexNet.

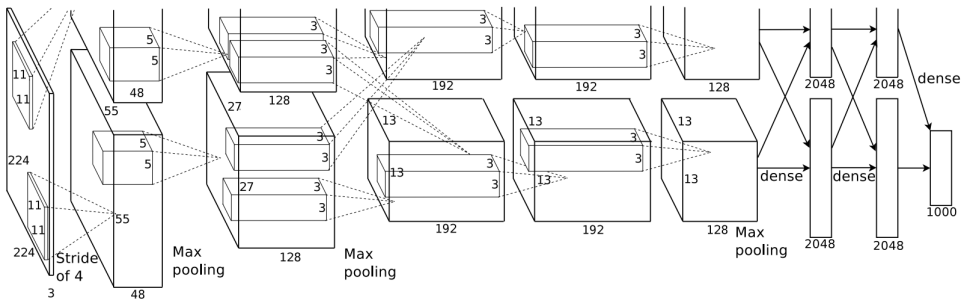

Figure 2.4: Architecture of AlexNet [29]

However, both LeNet and AlexNet are very shallow. In the following years, people started to use far deeper models to capture features, and to apply some similar functions with fewer parameters. Examples include VGG (Simonyan and Zisserman [44]), ResNet (He et al. [25]), and DenseNet (Huang et al. [27]). These all achieved superior performance and used less memory during inference, which enables mobile computing devices to deploy such systems.

VGG (Figure 2.5) uses smaller filters (3*3 Conv stride 1) and deeper networks (16-19 layers), compared to AlexNet. It uses a stack of three 3*3 conv (stride 1) layers, which has the same effective receptive field as one 7*7 conv layer. But it goes deeper, and has more non-linearities.

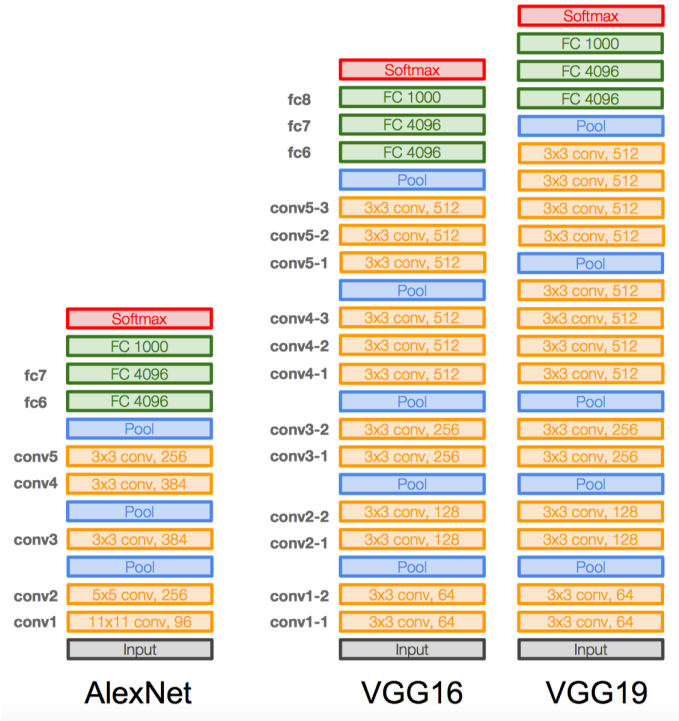

Figure 2.5: Architecture of VGG [5]

Then researchers realized that the error increases if we continue stacking deeper layers on a "plain" convolutional neural network. Figure 2.6 shows that the deeper the model goes, the worse it performs.

Figure 2.6: Errors Comparison [25]

So, researchers proposed a hypothesis: deeper models are harder to optimize; this is an optimization problem.

In order to solve this problem, in ResNet (as introduced in He et al. [25]), the authors used network layers to fit a residual mapping instead of directly fitting a desired underlying mapping (Figure 2.7).

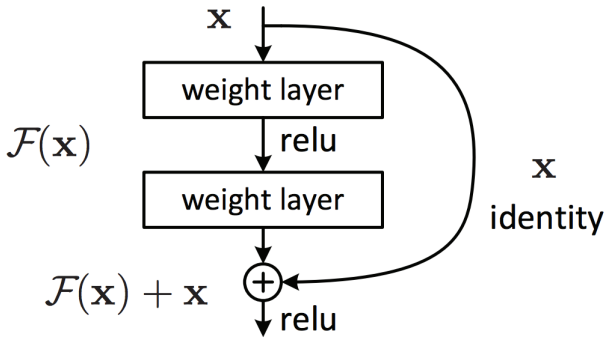

Figure 2.7: Residual Learning: a building block [25]

Experimental results showed that ResNet is able to train very deep neural networks (152 layers on ImageNet, 1202 layers on CIFAR), and achieved 1st place in both ILSVRC and COCO 2015 competitions.

Built upon that, in 2017, Huang et al. [27] proposed DenseNet which uses stronger connections between layers. It has several compelling advantages:

- strengthen feature reuse
- enhance feature propagation
- alleviate the vanishing-gradient problem
- greatly reduce the number of parameters

The network architecture and parameters are shown in Figure 2.8 and Figure 2.9 [27].

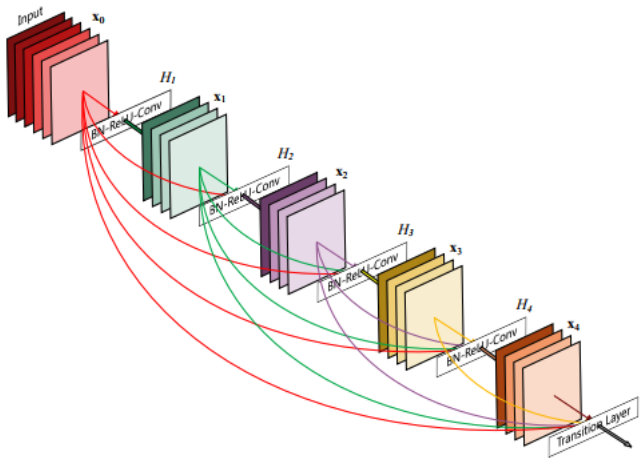

Figure 2.8: DenseNet Architecture [27]

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6	[1 × 1 conv; 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12	[1 × 1 conv; 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv; 3 × 3 conv] × 24	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48	[1 × 1 conv; 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv; 3 × 3 conv] × 16	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 32	[1 × 1 conv; 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Figure 2.9: DenseNet Parameters [27]

In DenseNet, there are direct connections between any two layers with the same feature map size. This introduces $\frac{N*(N+1)}{2}$ connections in an N-layer network, instead of just N connections. Compare to ResNet, DenseNet uses concatenation to combine features instead of summation. As is shown in Figure 2.8, the l^{th} layer collects the feature maps from all previous layers, $x_0, ..., x_{l-1}$ as input (Equation 2.1):

$$x_l = H_l(x_0, x_1, ..., x_{l-1}) \tag{2.1}$$

where $x_0, ..., x_{l-1}$ represents the concatenation of the feature maps generated in layers 0, 1, ..., l-1. H is defined as a blended function of three consecutive operations: BatchNormalization $\Rightarrow$ ReLU $\Rightarrow$ Conv(3*3).

The crucial part of CNNs is using down-sampling layers to change the size of feature maps and reduce parameters. To accelerate down-sampling in DenseNet, the network is divided into multiple densely connected dense blocks. The layers between blocks are transition

layers, which are designed to do the convolution and pooling operations. The transition layers consist of: BatchNormalization ⇒ Conv(1*1) ⇒ AvgPooling(2*2).

2.2 Medical Image Analysis

Imaging is a cornerstone of medicine. Experts rely heavily on medical image to diagnose diseases to treat patients. Medical image analysis is a science analyzing medical puzzles and solving medical problems via images, that can be used for diagnosis, segmentation, and therapeutic purposes. It is based on different imaging modalities and digital image analysis techniques.

Common modalities for imaging include:

- X-ray
- CT (Computed Tomography)
- MRI (Magnetic Resonance Imaging)
- Ultrasound

X-ray technology is the oldest but most widely used type of medical imaging. The X-rays work on a wavelength and frequency that are not able to be seen with human eyes, but are able to be absorbed in different amounts depending on the density of the material, thus creating a picture of what is going on underneath. They are quick, low cost and relatively easy for the patient to endure. Nonetheless, some risks are strongly associated with the use of X-rays due to the radiation, like radiation-induced cancer.

Figure 2.10: X-ray [2]

Computed Tomography (CT) is a medical imaging technology that uses a quickly rotated narrow beam of X-rays to produce thorough cross-sectional images of the body. It generates better clarity compared to traditional X-rays by providing detailed images of blood vessels, internal organs, and bones inside the body.

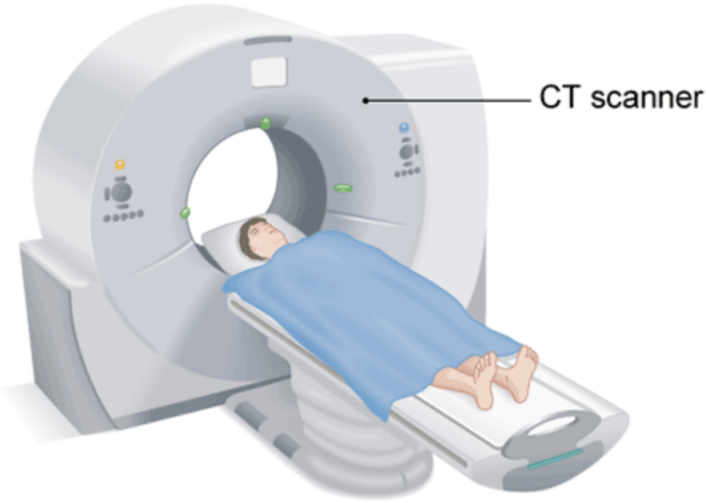

Figure 2.11: Computed Tomography (CT) [3]

Magnetic Resonance Imaging (MRI) is another medical imaging method that uses radio waves and a magnetic field to create detailed images of tissues and organs. It is commonly

used in examining internal body components to diagnose strokes, tumors, injuries, and brain functions. A crucial consideration in MRI is that it does not use X-rays and radiation; this distinguishes it from X-ray or CT scans.

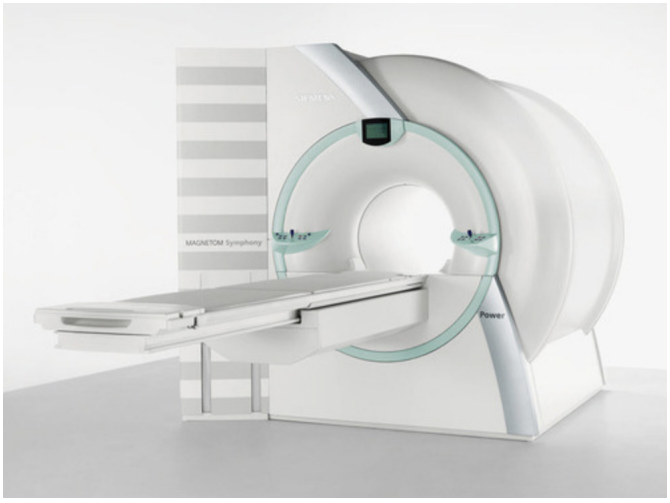

Figure 2.12: Magnetic Resonance Imaging (MRI) [4]

Ultrasound is the safest way of medical imaging and has a wide range of applications. It applies high-frequency sound waves to generate images of the inside of the body. It is often used to detect abnormalities in the heart and blood vessels, organs in the pelvis and abdomen, and pregnancy.

Figure 2.13: Ultrasound [9]

As the imaging modalities develop, the number of medical images grows rapidly, and the size and dimensionality of these images grow as well. This trend pushes researchers to improve medical imaging analysis and devise higher quality medical processes. Once it was possible for a machine to scan and load medical images, people started trying to build intelligent systems for automated analysis. Between the 1970s and 1990s, medical image analysis was conducted mainly with sequential application of pixel level preprocessing (e.g., region growing, line and edge detector filters) following by mathematical modeling (e.g., line, ellipse, curve fitting) to build solid rule-based systems that are able to solve specific tasks.

Starting from the end of the 1990s, supervised learning techniques, in which labeled training data is used to build model-based systems, were becoming increasingly popular in the medical image area, like active shape models (for segmentation), atlas models, and statistical classifiers. Those machine learning or pattern recognition approaches were very popular. That was a big shift from human-designed systems to training data based systems from which features are extracted. Computer-aided systems were able to determine the optimal

decision boundary in high dimensional feature space. However, extraction of distinctive features from images still needed to be done by human experts.

Then, researchers started to think about useful ways to let computers extract features by themselves and become able to optimally represent the data for various problems at hand.

As deep learning techniques took off, especially convolutional network architectures, they have been applied to analyzing medical images, and have pervaded the whole field of medical image analysis. The main difference between deep learning methods and traditional machine learning methods is deep learning methods are able to learn discriminative features automatically, rather than choose and encode features determined ahead of time. Figure 2.14 shows how the different parts of intelligent systems relate to each other. The shaded boxes indicate components that are able to learn from data directly.

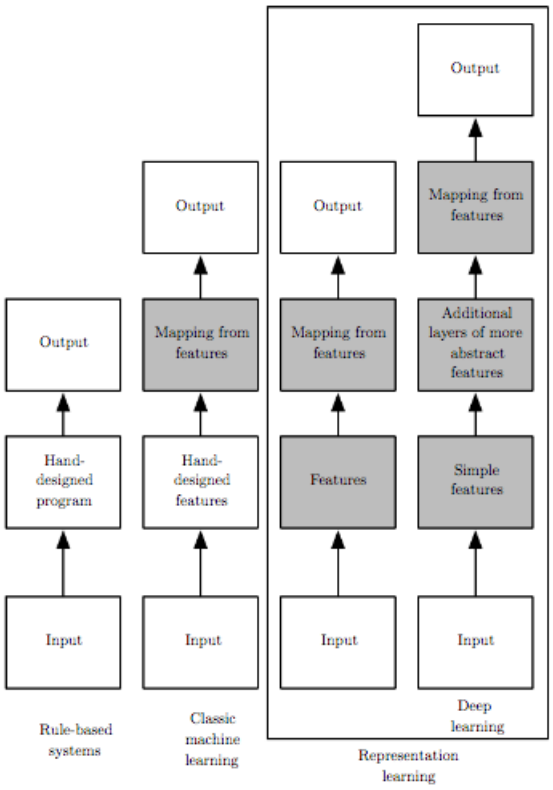

Figure 2.14: History of Intelligent Systems [23]

Within the brain imaging domain, deep neural networks (DNNs) have several applications. In [17], the authors introduced a deep learning based manifold learning method to learn the manifold of 3D brain images. They applied a deep generative model made up of multiple restricted Boltzmann machines (RBMs) [26] to the ADNI dataset [1], which contains 300 T1-weighted MRIs of Alzheimer's disease (AD) and normal subjects. In order to accelerate the computation process, they applied Convolutional RBMs (convRBMs), a form of RBM that uses weight-sharing to reduce the number of trainable weights. Results show that it is much more efficient to train with a resolution of 128*128*128 in practice and be able to extract brain-related disease features by automatically learning a low-dimensional manifold of brain

[1] http://adni.loni.usc.edu/

volumes. According to Suk et al. [47], they used a stacked auto-encoder (SAE) to discover latent representations from the original biological features; even nonlinear latent features can be found using SAE. They also applied another method – a combination of sparse regression models with deep neural networks (as introduced by [48]) – to the ADNI dataset. In [32], the authors overcame the problem that not all subjects have all modalities by applying 3-D CNN to predict the missing Positron-emission tomography (PET) patterns from the MRI images. They trained a 3-D CNN model which is able to capture the nonlinear relationship between modalities by feeding pairs of volumetric data modalities into the network.

Those DNN based methods had completely taken over many brain image analysis challenges. In the 2014 and 2015 BRATS (brain tumor segmentation challenges), the 2015 ISLES (ischemic stroke lesion segmentation) challenge, and the 2013 MRBRAINS (MR brain image segmentation challenge), the top teams all used end-to-end CNNs.

Within chest images, many deep learning based applications were applied in detection and classification of nodules as well. As introduced in [11], the authors built an X-ray image retrieval system by experimenting on binary features, texture features, and deep learning (CNN) features. Best results are achieved by using deep learning features in a classification scheme. According to Wang et al. [50], they proposed using deep feature fusion from the non-medical training and hand-crafted features to reduce the false positive results in the lung nodules detection task. Compared to previous methods, their method achieved better results in sensitivity and specificity (69.3% and 96.2%) at 1.19 false positives per image on the JSRT (Japanese Society of Radiological Technology) database [43]. And in [51], researchers invented a cascade architecture of CNN, used for learning the mapping between the gradients of the chest radiographs and the corresponding bone images. They evaluated on 504 cases of real two-exposure dual-energy subtraction chest radiographs and were able to produce high-resolution bone and soft-tissue images.

Moreover, in recent challenges for nodule detection, top systems all used CNN architectures. Compared to previous challenges, like ANODE09, where handmade features were used to extract nodules, deep learning based methods outperformed by a large margin.

Transfer learning in the medical image has also been widely applied, which typically uses pre-trained networks to work around the specific large datasets for further training. Two use cases were identified: (1) use a pre-trained model as a fixed feature extractor and (2) fine-tuning a pre-trained network on medical data. The first one does not need people to train the network, and the second one allows people to train the network by leveraging the already existing labeled data of some related task or domain. Both of these methods are widely applied, according to Litjens et al. [33].

2.3 Diabetic Retinopathy Grading

In the 1990s, detecting DR was a very time-consuming and manual process that required a trained clinician to examine and evaluate digital color fundus photographs of the retina. Although some papers, like Klein et al. [28], proposed methods to detect DR, they were mainly focus on tools, like ophthalmoscopy, non-mydriatic camera, and standard fundus camera, not on methodology.

Then researchers started to use image classification, pattern recognition, and machine learning methods in automated DR detection, and made good progress. In 1996, Gardner et al. [22] constructed a simple backpropagation neural network to recognize features in the retinal images, which achieved better results compared to the ophthalmologist and proved that recognizing vessels, exudates, and hemorrhages is possible. In 2002, Walter et al. [49] contributed to image analysis for the diagnosis of diabetic retinopathy. They started to use image enhancement to improve contrast and sharpness and reduce noise, then applied mass

screening to detect features of the retina. Some morphological operations were also applied in their experiment. In 2009, Ravishankar et al. [39] applied an optic disk detection pattern which pinpoints major blood vessels first, then locates the optic disk based on intersections of blood vessels and color properties. They also used morphological operations to detect blood features of diabetic retinopathy, like exudates, microaneurysms, and hemorrhages.

In recent years, with the improvement in medical image quality and quantity, also with the development of deep learning and computational infrastructure, many works in retina related areas achieved good performance; they all used simple CNNs for color fundus imaging analysis. In 2016, Fu et al. [20, 21] applied a multi-scale and multi-level CNN with a side output layer to learn a rich hierarchical representation, and utilized a Conditional Random Field (CRF) to model the long-range interactions between pixels, then combined CNN with CRF layer into an integrated deep network called DeepVessel, to segment the blood vessel. In the same year, Abràmoff et al. [10] created a hybrid system using CNNs as a high performing lesion detector.

In 2015, Kaggle held a diabetic retinopathy detection competition; they provided around 35,000 color fundus retina images for training, and around 53,000 for testing. The top teams, like Graham [24], all used end-to-end CNN models and achieved good performance.

2.4 Tree-based Algorithms

Tree-based algorithms are treated as one of the most widely used supervised learning methods and can be used for both regression and classification problems. They partition the space and identify some representative centroids. Unlike linear methods, where classification boundaries are determined by using hyperplanes, tree algorithms always use a hierarchical way of partitioning the space.

In 1984, Breiman et al. [16] created Classification and Regression Tree (CART). It applied a greedy splitting strategy, recursively splitting the input space. Like in a sequence of questions, the answer to the current question determines what the next question is. The results of those questions is a tree-shaped structure. Figure 2.15 shows a simple example of CART on the Titanic problem [2].

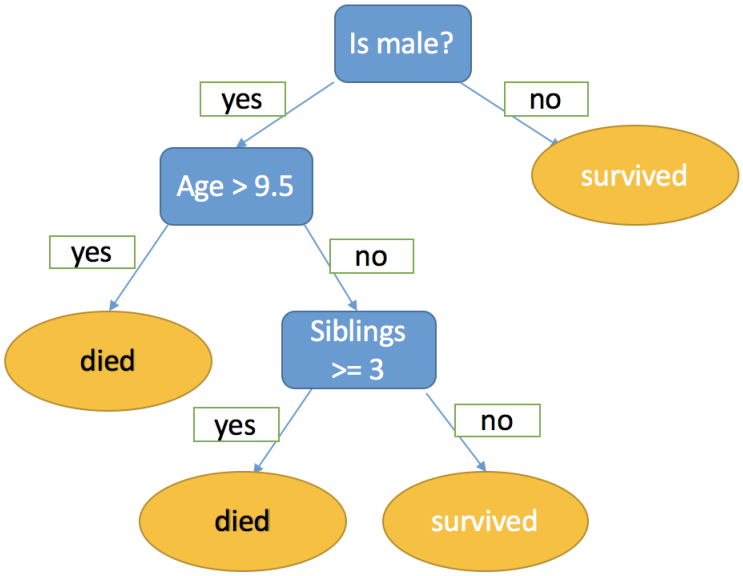

Figure 2.15: A Tree Structure on "Titanic"

The main components in CART are:

- Splitting rule: based on the value of one feature at each node
- Stopping rule: to terminate a branch
- Prediction: each leaf node has to have a prediction
- Greedy manner: choose the very best split point at each time

The splitting rule is based on the greedy manner: always choose the best split point at

[2] https://www.kaggle.com/c/titanic

each time. For stopping rule, it uses a minimum count on the number of training instances assigned to each leaf node. If the count is less than some threshold, then the split is not allowed, and the node is treated as a final leaf node.

For regression modeling, the cost function is designed to minimize the sum squared error (as shown in Equation 2.2).

$$loss = \sum (y - y_{prediction})^2 \tag{2.2}$$

The depth of the tree is very important for this algorithm. If the tree is very deep, that might cause overfitting; however, if the tree is very shallow, it might not be able to capture the features from the data. So, the strategy here is to grow a large tree first, and stop growing only when reaching the maximum number of nodes. After that, use "cost-complexity pruning" (as introduced in Ripley [41]).

For classification modeling, the Gini index is used as loss function (shown in Equation 2.3). The Gini index is a measure of the homogeneity (or "purity") of the nodes. If all data points at one node belong to the same class then this node is considered "pure". So, by minimizing the Gini index the decision tree finds the features that separate the data best.

$$Gini - Index = \sum (p_i * (1 - p_i)) \tag{2.3}$$

The splits in the CART are mainly binary splits because multiway splits decompose the data too quickly. That might lead to insufficient data for the next level since multiple binary splits can replace a multiway split; hence the binary split is preferred.

Then Quinlan [38] proposed the C4.5 and C5.0 algorithms. They apply a different pruning technique, error-based pruning, and use Shannon Entropy (shown in Equation 2.4) to pick

features with the greatest information gain as nodes.

$$Cross - Entropy = -\sum p_i log p_i \tag{2.4}$$

The problem for single-tree algorithms is high-variance and overfitting the training data, so many advanced methods were invented.

The bagging method was exploited by Breiman [14]; the authors proposed to produce "bagged" classification trees. The key idea is averaging over the results from a large number of bootstrap trees. This method can generalize easily to a wide variety of classifiers beyond classification trees.

Then, the authors went ahead and proposed the Random Forest algorithm [15]. It is very similar to bagging, but it uses a random sample of predictors before each node is split. For example, if there are fifty predictors, the bagging method would choose a random ten as candidates for constructing the best split. Repeat this process for each node until the tree is large enough. And as in bagging, do not prune.

It achieved excellent performance because it solved a big problem in tree-based algorithms: high variance. The trees in Random Forest are more independent due to the combination of bootstrap samples and random draws of predictors. It is also able to reduce bias because there are a large number of predictors that can be considered, so more information might be brought in to reduce bias.

Boosting algorithms, like bagging, are another good approach in improving prediction results for a variety of machine learning methods. They apply a sequential approach: first, use subsets of the original data to produce a series of weakly performing models, then "boost" their performance by combining them using a particular cost function (majority vote). Boosting

is a strong classifier, using a combination of week classifiers $h_t(x)$:

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{2.5}$$

Each tree in a boosting algorithm is grown based on previously grown trees. Instead of using a single decision tree to learn from the data, which may lead to overfitting, the boosting algorithm learns slowly. Give one decision tree, the new decision tree is fitted to the residual of the previous model. Then, add this new decision tree to the fitted model to update. By this way, we can slowly improve in the places where the single model does not perform well.

Chapter 3

Datasets

Table 3.1: Diabetic Retinopathy (DR) Severity Scale

Disease Severity Level	**Findings**
Grade 0: No apparent retinopathy	No visible sign of abnormalities
Grade 1: Mild NPDR	Presence of microaneurysms only
Grade 2: Moderate NPDR	More than just microaneurysms but less than Severe NPDR
Grade 3: Severe NPDR	Any of the following: • >20 intraretinal hemorrhages • Venous bleeding • Intraretinal microvascular abnormalities • No signs of PDR
Grade 4: PDR	Either or both of the following: • Neovascularization • Vitreous/pre-tinal hemorrhage

According to severity levels, diabetic retinopathy can be classified into five level, from NPDR to PDR (shown in Table 3.1). NPDR is the early stage of the disease where symptoms will be mild or non-existent. Blood vessels in the retina are weakened in NPDR. Fine bulges in the blood vessels called microaneurysms may leak fluid into the retina. This kind of leakage may lead to swelling of the macula. PDR is the more advanced type of the disease. At

this stage, circulation problems deprive the retina of oxygen. As a result, new and fragile blood vessels begin to grow in the retina and into the vitreous, the gel-like fluid that fills the back of the eye. The new blood vessels may leak blood into the vitreous, and cause cloudy vision [1].

Figure 3.1 shows some (sub)types of diabetic retinopathy [2].

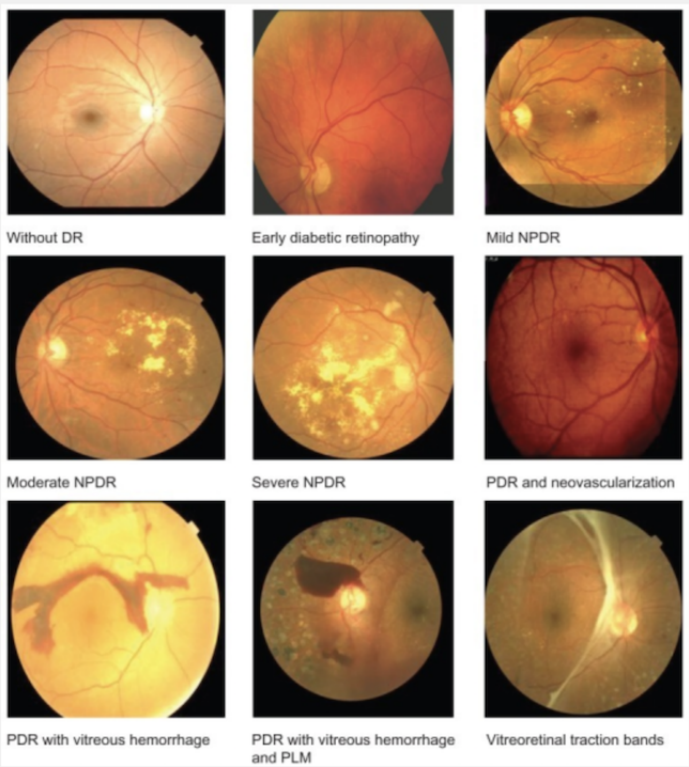

Figure 3.1: Types of Diabetic Retinopathy

In our experiments, we use data from 3 sources.

The first source is from sub-challenge 2 [3] of the ISBI 2018 challenges [4]. Retina image data

[1] https://www.aoa.org/patients-and-public/eye-and-vision-problems/glossary-of-eye-and-vision-conditions/diabetic-retinopathy

[2] https://openi.nlm.nih.gov/detailedresult.php?img=PMC3284208_opth-6-269f3&query=&it=xg&req=4&qimg=0.775042001524893499a&npos=1

[3] https://idrid.grand-challenge.org/grading/

[4] http://biomedicalimaging.org/2018/challenges/

from the IDRiD (Indian Diabetic Retinopathy Image Dataset) database was provided, which is the first database representative of an Indian population. Each retina image is labeled by severity level. Initially, 80% of the data (training data) with the ground truth was released on January 20, 2018; then the remaining 20% (test data) was provided on the day of challenge workshop (April 4, 2018). See Table 3.2 regarding the released data.

Table 3.2: Proportion of Images Per Grade in IDRiD

Diabetic Retinopathy (No. of Images)			
Severity Level	**Total Number**	**Training Set**	**Testing Set**
Grade 0	168	134	34
Grade 1	25	20	05
Grade 2	168	136	32
Grade 3	93	74	19
Grade 4	62	136	13

The second source is from the Kaggle website; there are 35,126 high-resolution color retina images taken under diverse imaging conditions that were released in the 2015 diabetic retinopathy detection competition (https://www.kaggle.com/c/diabetic-retinopathy-detection). Images of a left and a right eye are provided for every patient. Images are labeled with a patient ID as well as either left or right (e.g., 1_left.jpeg means the left eye of patient with ID 1). The quantity and proportions are shown in Table 3.3.

Table 3.3: Proportion of Images Per Grade in Kaggle

Severity Level	**Training Set**	**Percentage**
Grade 0	25810	73.48%
Grade 1	2443	6.96%
Grade 2	5292	15.07%
Grade 3	873	2.48%
Grade 4	708	2.01%

For the last source, we use the DRIVE (Digital Retinal Images for Vessel Extraction) database from its website (https://www.isi.uu.nl/Research/Databases/DRIVE/), which was established to enable comparative studies on segmentation of blood vessels in retinal images

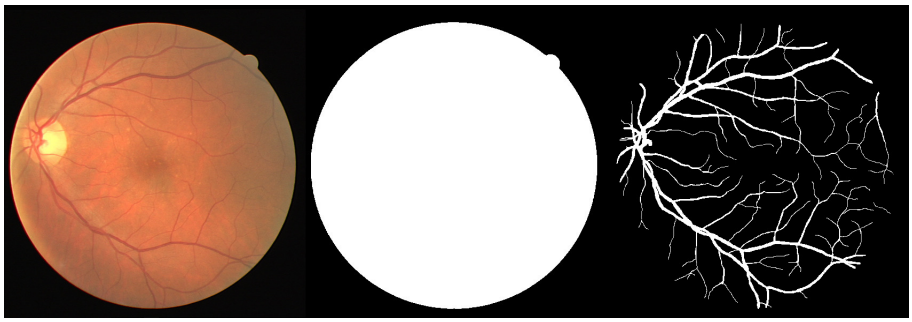

Figure 3.2: DRIVE Data Sample Image

(refer to Staal et al. [46] for details). The images in the DRIVE database were obtained from a diabetic retinopathy screening program in the Netherlands. Each image has been JPEG compressed. The screening population consisted of 400 diabetic patients, 25-90 years of age. 40 photographs have been randomly selected, 33 of them do not show any trace of diabetic retinopathy and 7 of them show signs of mild early diabetic retinopathy. Examples are shown in Figure 3.2.

Chapter 4

Methodology

This chapter is our methodology part: how we have conducted the whole experiment.

4.1 Pipeline Overview

The whole pipeline for our experiment is illustrated in Figure 4.1

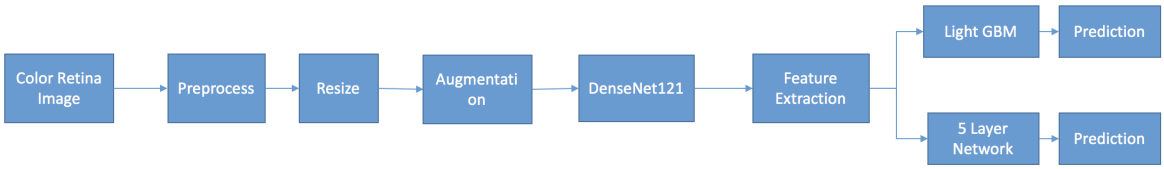

Figure 4.1: Pipeline

First, we input color retina images and use various methods to preprocess images. Then, we resize and augment images. After that, we apply DenseNet 121 to pre-train a model on the dataset. We do not directly use the output from DenseNet 121; instead, for each image, 50 times augmentations are applied and fed into the pre-trained model, forward all the way

to the last second fully connected layer. We then get outputs from that layer and calculate the mean vector and standard deviation vector as features for that image. Finally, we use extracted features to train and test on a Light GBM model and a 5-layer neural network model, respectively.

4.2 Data Preprocessing

In the medical image area, researchers applying similar network architectures may have different results. That is because feature engineering and expert knowledge available in an image are often overlooked by people. In this research experiment, many image preprocessing methods have been applied.

We first apply standardization, to have zero mean and unit norm. Basically, our raw image has three channels, and in each channel, pixel values range from 0 to 255. Our goal is to squash the range of values for all the pixels in the three channels to a small range. Using Equation 4.1 below can transform each image to have zero mean and unit variance:

$$x_{new} = \frac{x - \mu}{\sigma} \sim N(0, 1) \tag{4.1}$$

(μ represents the mean, and σ represents the standard deviation).

The reasons we conduct standardization are as follows:

1. Make training less sensitive to the scale of features: because our data are collected from different cameras, and under different conditions, so the scale of values in image channels is different. In that case, a very large distance from one variable may dominate the response variable in high dimensional space. But unit variance can help us ensure proportional contributions from all features.

2. Consistency for comparing results across models: providing the same scaling to compare among methods.

3. Make optimization well-conditioned: We use stochastic gradient descent to optimize during training, and the speed of convergence depends on the scaling of features (or more specifically, the eigenvalues of $X^T X$). Normalization makes the problem better conditioned, improving the convergence rate of gradient descent.

Then, we also apply a brightness-preserve operation. The preservation of mean brightness of the input image is essential for medical image analysis, so a morphological closing operation and contrast enhancement technique are employed. According to Lachure et al. [30], exudates and microaneurysms in the retina are easier to detect after applying morphological operations, like closing. The morphological closing operation, essentially, is obtained by the dilation of an image followed by an erosion operation. It is efficient in closing small holes on the objects. Examples are shown in Figure 4.2 [1]. The left one is the original image, and the right one is the image after the morphological closing operation. Contrast enhancement, basically, is using the difference in visual properties to make an object distinguishable from other objects and the background. It is able to preserve the mean brightness satisfactorily and also produce better quality images. As a result, according to Datta et al. [18], it indeed improves the overall MA detection. An example of contrast is shown in Figure 4.3. The left one is the original image, and the right one is the image after contrast enhancement.

[1] https://docs.opencv.org/trunk/d9/d61/tutorial_py_morphological_ops.html

Figure 4.2: Morphological Closing Example

Figure 4.3: Image Contrast Enhancement Example

According to Reimers et al. [40], evaluating DR severity from a simulated red-free (green) channel is comparable to using color images – perhaps slightly more sensitive for some lesions. And according to Sinthanayothin et al. [45], the color band chosen for recognition of hemorrhages and microaneurysms (HMA) was green as it contained more information and greater contrast for red features. We thus extract the green channel only for each image, which not only makes it easier to observe lesions in retina image but also reduces storage space by 2/3.

Since the original images are fairly large (say, 4000*25000 pixels on average), in order to make data suitable for deep learning training and also reduce computation, all input images

are resized to 256*256. Figure 4.4 shows the original image, brightness-preserved image, and green only image.

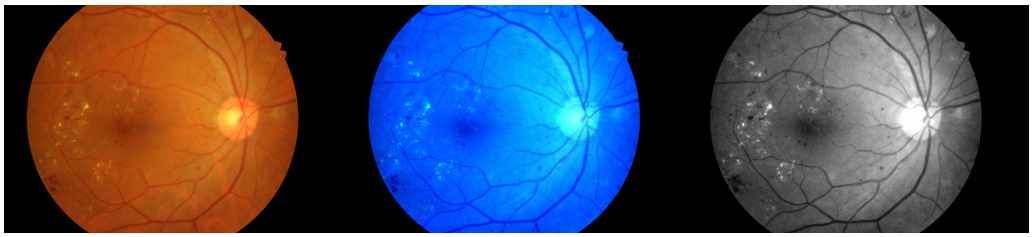

Figure 4.4: Original, Bright Preserved and Green Only Image

Segmentation also has been tried in the preprocessing. Diabetic retinopathy can induce blood vessels in the retina to leak fluid, distorting eyesight. In its most mature stage, new abnormal blood vessels expand (increase in number) on the surface of the retina, which can lead to scarring and cell loss in the retina. So, the shape of blood vessels also is an indicator of DR. The DRIVE dataset is used to segment blood vessels from retina images. Because this is a binary classification task – the neural network predicts if each pixel in the fundus image is either a vessel or not – we apply sequential approaches. First, we apply gray-scale conversion for each image. Then, standardization has been performed to have zero mean and unit norm. In order to avoid the over-brightness problem, adaptive histogram equalization is used. But images may be divided into small blocks and histograms may restrict to a small region (unless there is noise); if noise is there, it will be enlarged. To alleviate this problem, contrast limiting is applied. After that, sub-images of the preprocessed full image are randomly selected; each patch's size is 48*48. 200,000 patches are randomly selected: 10,000 patches from each of 20 DRIVE datasets. Sample input images are shown in Figure 4.5, and sample input masks are shown in Figure 4.6. 10% of them are selected as validation, and 90% are selected as training. Then we feed those as input into our U-Net model (Ronneberger et al. [42]). Rectifier Linear Unit (ReLU) is used as an activation function after each convolutional layer, and a dropout of 0.2 is used between two consecutive convolutional layers; cross-entropy and stochastic

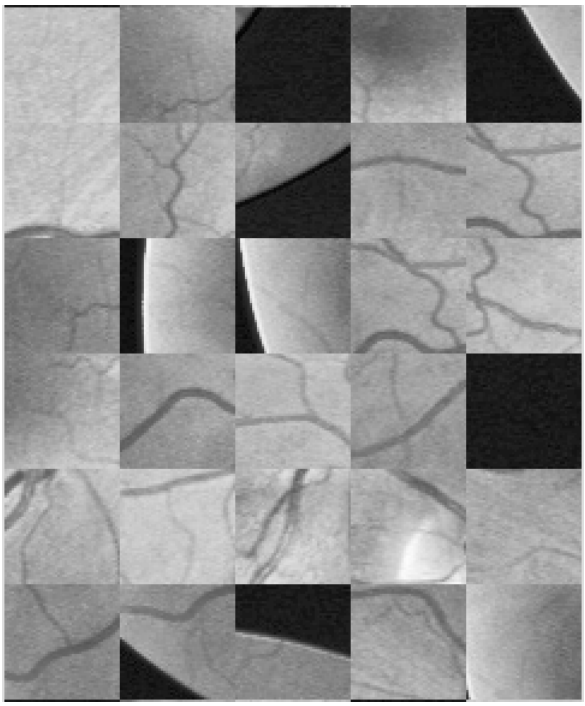

Figure 4.5: Sample U-Net Input Image

gradient descent have been applied for optimization. Our u-net structure is shown in Figure 4.7.

For computational resources, 2 GeForce GTX 1070 GPUs are used for u-net training; the whole process requires around 15 hours. Results are shown in Table 4.1 and an example segmentation result is shown in Figure 4.8.

Table 4.1: Blood Vessels Segmentation Results on DRIVE dataset

Metrics	**Score**
Accuracy	0.9560
Sensitivity	0.7671
Specificity	0.9835
Precision	0.8717
AUROC	0.9790
F1 score (F-measure)	0.8160

Figure 4.6: Sample U-Net Input Mask

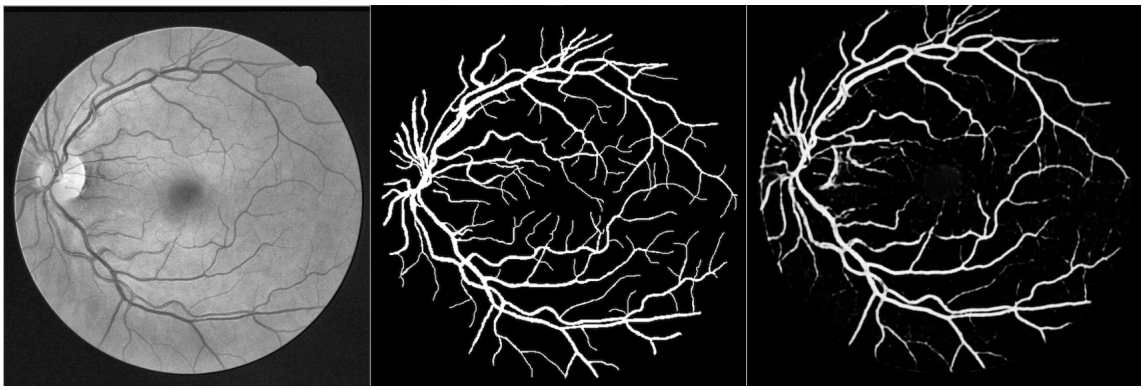

Figure 4.8: Original - Ground Truth - Prediction

Because there is no mask image in the Kaggle and IDRiD datasets, so we have not included it in our pipeline. Nevertheless, this is a very good method for retinal blood vessel segmentation.

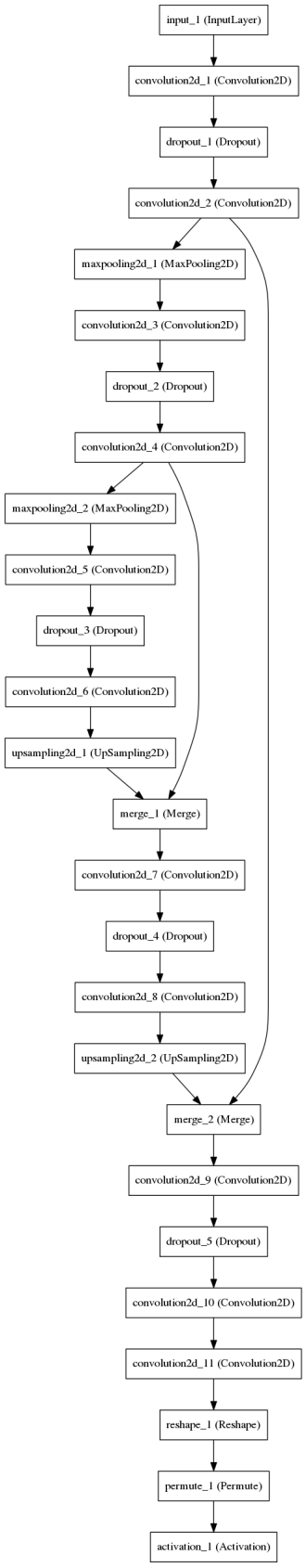

Figure 4.7: U-Net Architecture

4.3 Data Augmentation

After preprocessing, image datasets are computationally more feasible to allow many transformations. To make the model more robust, not overfitting, data augmentation methods have been applied to train the model. The main reason we augment the training set is to increase the number of training examples and make the pipeline more robust. Like with many other medical image databases – due to different medical facility conditions, like different models and types of cameras, or different racial groups – image quality or visual appearance may be affected. Examples are shown in Figure 4.9; most of the images like the leftmost image are very normal, but some images like the middle image are too bright, while some images like the right image are too dark. Hence, the following augmentations (transformations) are applied in our experiments:

- Cropping images to 224*224
- Flipping
- Shearing
- Rotating images by [0, 360] degrees
- Zooming (equal cropping on x and y dimensions)

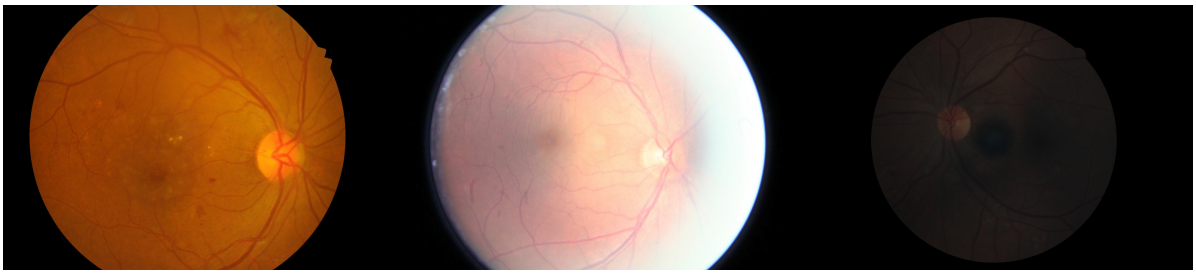

Figure 4.9: Different Image Quality

4.4 Benchmark

We applied deep learning as a method to predict diabetic retinopathy severity level from color retina images. Specifically, we construct a CNN model based on the DenseNet 121 architecture. DenseNet yields the state of the art in many image classification tasks. So we apply DenseNet 121 in our experiment as benchmark method. 121 is calculated by 5+(6+12+24+16)*2, where 5 is Conv+Pooling+3Transition, and 6,12,24,16 is for each dense block, and we multiply by 2 because each block has 2 layers (1*1 conv and 3*3 conv).

As controlled experiments, we use VGG19 and ResNet101 architectures, to test which CNN architecture works best in the DR grading task.

Retina images are fed into this network to pre-train the model; we apply cross-entropy as loss function and stochastic gradient descent as optimization; we stop training after 200 epochs.

An important problem in our experiment is that the distribution among the levels in the dataset is highly imbalanced. For example, grade0 describes 73.48 percent of the images, while grade3 and grade4 each have around 2 percent.

In order to alleviate the problem, where the model prefers to predict image level according to the higher proportion level in the training set, a crucial strategy we apply here is, during training, to resample the weights of loss that yields balanced classes. Accordingly, we use 1.3609 for grade 0, 14.3782 for grade 1, 6.6375 for grade 2, 40.2359 for grade 3, and 49.6129 for grade 4. In this case, images in small proportion are able to get heavier loss weights during training, and all classes are balanced. See in Table 4.2.

Table 4.2: Weights of Loss for Each Level

Severity Level(A)	**Training Set(B)**	**Percentage(C)**	**Loss Weight(D)**	**C * D**
Grade 0	25810	73.48%	1.3609	0.9989
Grade 1	2443	6.96%	14.3782	1.0007
Grade 2	5292	15.07%	6.6375	1.0002
Grade 3	873	2.48%	40.2359	0.9978
Grade 4	708	2.01%	49.6129	0.9972

We conduct our first experiment on the Kaggle dataset only. To make sure the training set and test set have the same distribution, we select 80% of the images of each level for training, and test on the other 20%. The result of this experiment is used as the benchmark on Kaggle dataset.

In our second experiment, we use the whole Kaggle dataset for training first. Then we also split 80% of the IDRiD dataset for fine-tuning and another 20% is used for testing. Also, we make sure that training set and test set have the same distribution. The same loss function and optimization are used, and we stop it after 50 epochs when tuning. We set the learning rate of this DenseNet 121 to 0.0005, and it decreases by 0.1 every 30 epochs, which is very low because we observe that a large rate can lead to poor convergence in our experiments. The result of this experiment is used as the benchmark on IDRiD dataset.

4.5 Feature Extraction

Feature extraction is the key and time-consuming part of the workflow. Previously, the feature engineering method was in domination. Recently, deep learning techniques have started to demonstrate superior performance on extracting features, much better than hand-

crafted feature detectors.

Many CNNs has been shown to be good feature extractors because hierarchical convolutional layers inside are able to provide high-level features and patterns. In our experiments, we use the DenseNet 121 model, trained on the Kaggle and IDRiD datasets, to extract features. Specifically, we use the last second fully connected layer of our model to extract features. In order to make extracted features more stable, we apply 50 random augmentations for each image to get 50 outputs from the last second fully connected layer (size of 50*1024), then the mean vector and standard deviation vector are calculated (size of 2*1024), and provided as features.

4.6 Feature Blending

We experiment on several algorithms with extracted features, including a gradient boosting decision tree algorithm and neural network.

Decision tree algorithms in machine learning have demonstrated amazing performance when features are well extracted. For example, the Xgboost algorithm has achieved top performance in many Kaggle competitions. In our experiment, we also apply a gradient boosting decision tree algorithm: Light Gradient Boosting Decision Tree (Light GBM).

Light GBM is a relatively new algorithm, built based on decision tree algorithms, and it is used mainly for classification, ranking, and other machine learning tasks [2]. It differs from other tree algorithms in many ways. First, Light GBM grows tree leaf-wise, while other algorithms grow tree level-wise. It will choose the leaf with max delta loss to grow. Figure 4.10 and Figure 4.11 [3] show the difference between Light GBM and other boosting

[2] https://github.com/Microsoft/LightGBM
[3] https://github.com/Microsoft/LightGBM/blob/master/docs/Features.rst

algorithms.

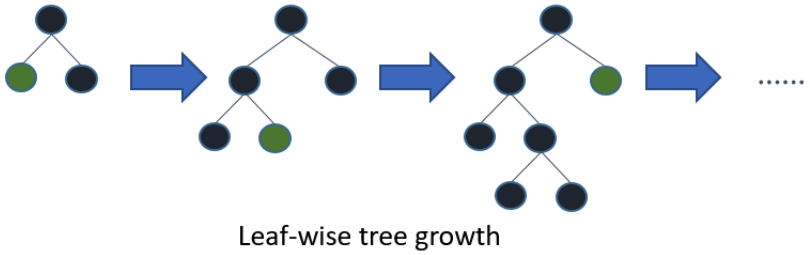

Figure 4.10: How LightGBM works

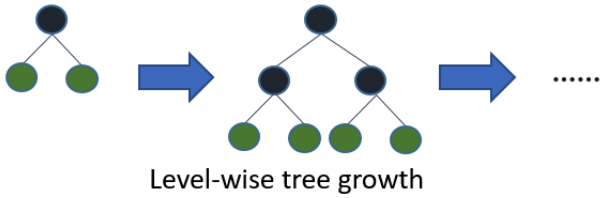

Figure 4.11: How other boosting algorithm works

Leaf-wise splits can lead to increase in complexity and finally may cause overfitting when the dataset is small, but "max_depth" is the parameter we can set to limit depth of tree and avoid overfitting.

Compared to Xgboost, Light GBM outperforms in many aspects:

- Fast training speed and higher efficiency: Light GBM buckets continuous feature values into discrete bins to accelerate the training procedure

- Low memory usage: it replaces continuous values using discrete bins to reduce memory usage

- Higher accuracy: it can produce much more complex trees by following a leaf-wise split approach, which is the main reason for achieving higher accuracy

- Parallel learning: it supports both feature parallel and data parallel

- GPU support: makes training even faster
- Deals with large-scale data
- Corporate support

For hyperparameters in the Light GBM algorithm, we apply random search. For random search, essentially, we need to initialize some random values for each hyperparameter, then randomly select values for hyperparameters, several times. From the final results, we compare and choose the best group of parameters for our model in our final prediction tasks.

In our experiment, we use extracted features, whose dimension is 2048, as input, and feed into our Light GBM model; randomly select hyperparameters and train 7 times; and select the best group of parameters. We choose multi_logloss as loss function, GBDT (gradient boosting decision tree) and DART (Dropouts meet Multiple Additive Regression Trees) as the random choice of boosting types. The output of Light GBM is the severity level of a retina image.

As a comparison with Light GBM, we construct a five-layer fully connected neural network, with details shown in Figure 4.12.

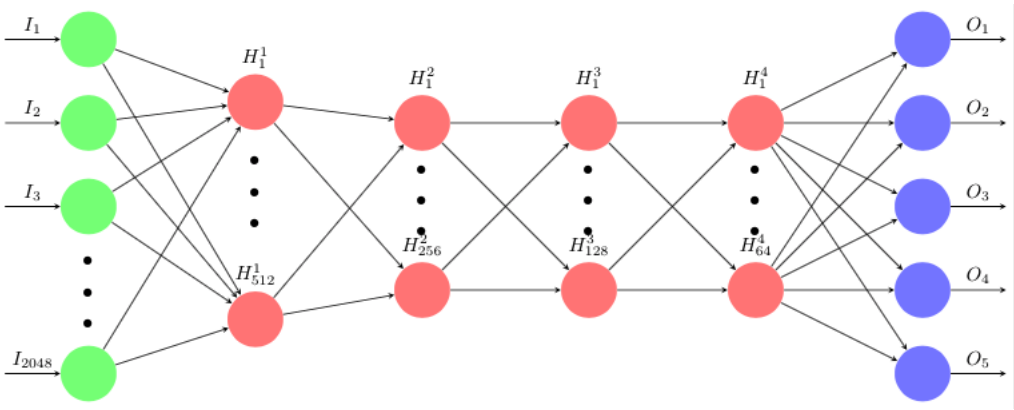

Figure 4.12: Five-layer Network

For each retinal image, we extract features from DenseNet 121 (dimension is 2048) and feed into this 5-layer neural network. For hidden layers, dimensions are 512, 256, 128, and 64, respectively. Leasky ReLU (0.01) has been applied as activation function (as shown in Equation 4.2), with Adam optimizer (where the learning rate is 0.0001).

$$f(x) = \begin{cases} x & (if \quad x > 0) \\ 0.01x & (otherwise) \end{cases} \tag{4.2}$$

The output of this network is the probability of each of the five levels, so the prediction is chosen as the level with the highest probability.

Chapter 5

Results

5.1 Metrics

In our experiments, we use accuracy, quadratic weighted kappa score, precision, recall, and F1 score to evaluate the performance of our model.

The accuracy metric is required by the Grand-Challenge, which is defined in Equation 5.1:

$$Accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n-1} 1 \quad (\hat{y}_i = y_i) \tag{5.1}$$

The quadratic weight kappa metric is a measurement which evaluates inter-rater agreement for qualitative (categorical) items. It is generally treated as a more robust measure than simple percent agreement calculation, as it takes into account the possibility of the agreement occurring by chance [1]. Its value usually is between 0 and 1, where 0 means random agreement

[1] https://en.wikipedia.org/wiki/Cohen%27s_kappa

and 1 means complete agreement between raters [2]. Only when there is less agreement between the raters than expected by chance, will this metric go below 0 [1].

We also report performance on the widely used metrics, namely precision, recall, and F1 score, defined in equations 5.2, 5.3, and 5.4, respectively.

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{5.3}$$

$$F1 = \frac{2 * TP}{2 * TP + FN + FP} \tag{5.4}$$

(TP represents True Positive; TN represents True Negative; FP represents False Positive; FN represents False Negative.)

5.2 Results

We conduct two experiments on the Kaggle and IDRiD datasets. In order to prove our pipeline works for general retina images, we first test on the Kaggle dataset. Then, we also fine-tune the model from the first experiment on the IDRiD dataset, to prove our pipeline is able to grade the IDRiD dataset.

[2] https://www.kaggle.com/c/diabetic-retinopathy-detection#evaluation

5.2.1 Results on Kaggle Dataset

In the first experiment, we randomly select 80% of the images at each level in the Kaggle training data to train the CNN, and the remaining 20% as the test set. We first conduct an experiment without any image preprocessing on DenseNet 121, VGG 19, and the ResNet 101 architecture. The experiment results are shown in Table 5.1.

Table 5.1: Controlled Experiment on Kaggle Dataset

Experiment	**Accuracy**	**Kappa Score**
VGG 19	0.35	0.49
ResNet 101	0.32	0.44
DenseNet 121	**0.38**	**0.54**

Because the DenseNet architecture has superior performance compared to the other architectures, we conduct further experiments based on DenseNet: we compare the influence from different pre-processing and post-processing methods based on DenseNet. The experiment results are shown in Table 5.2.

Table 5.2: Experiment Reults on Kaggle Dataset

Experiment	**Accuracy**	**Kappa Score**
DenseNet 121 (Benchmark)	0.38	0.54
Green-channel-only DenseNet 121	0.54	0.72
Bright-preserved DenseNet 121	0.60	0.77
5-layer NN + DenseNet 121	0.50	0.70
5-layer NN + Green-channel-only DenseNet 121	0.60	0.80
5-layer NN + Bright-preserved DenseNet 121	0.63	0.84
LightGBM + DenseNet 121	0.49	0.71
LightGBM + Green-channel-only DenseNet 121	0.61	0.82
LightGBM + Bright-preserved DenseNet 121	**0.65**	**0.84**

We compare our best model, which uses Light GBM on features which were obtained from DenseNet trained on the bright-preserved image, with the benchmark, considering precision, recall, and F1-score. Results are shown in Table 5.3.

Table 5.3: Classification Report on Kaggle dataset

DR Level	Precision		Recall		F1 Score	
	Benchmark	Best Model	Benchmark	Best Model	Benchmark	Best Model
0	0.88	**0.92**	0.99	0.96	0.93	**0.94**
1	0.56	**0.70**	0.23	**0.31**	0.32	**0.43**
2	0.08	**0.64**	0.17	**0.54**	0.10	**0.58**
3	0.35	**0.67**	0.10	**0.76**	0.15	**0.71**
4	0.20	**0.72**	0.19	**0.74**	0.19	**0.73**

5.2.2 Results on IDRiD Dataset

We conduct the second experiment by using the whole Kaggle dataset for training and randomly select 80% of each grade in the IDRiD dataset for fine-tuning. The remaining 20% are used to test the model performance. We also conduct a controlled experiment on the VGG19, ResNet101, DenseNet121 architectures. Experiment results are shown in Table 5.4.

Table 5.4: Controlled Experiment on IDRiD Dataset

Experiment	**Accuracy**	**Kappa Score**
Fine-tuned VGG 19	0.43	0.56
Fine-tuned ResNet 101	0.44	0.60
Fine-tuned DenseNet 121	**0.49**	**0.69**

On the IDRiD dataset, DenseNet is still the best architecture for this pipeline, so we also compare the influence from different pre-processing and post-processing methods based on DenseNet. The experiment results are shown in Table 5.5.

Table 5.5: Experiment Reults on IDRiD Dataset

Experiment	**Accuracy**	**Kappa Score**
Fine-tuned DenseNet 121 (Benchmark)	0.49	0.69
Green-channel-only Fine-tuned DenseNet 121	0.56	0.76
Bright-preserved Fine-tuned DenseNet 121	0.58	0.79
5-layer NN + Fine-tuned DenseNet 121	0.55	0.76
5-layer NN + Green-channel-only Fine-tuned DenseNet 121	0.59	0.78
5-layer NN + Bright-preserved Fine-tuned DenseNet 121	0.64	0.83
LightGBM + Fine-tuned DenseNet 121	0.55	0.77
LightGBM + Green-channel-only Fine-tuned DenseNet 121	0.62	0.81
LightGBM + Bright-preserved Fine-tuned DenseNet 121	**0.66**	**0.84**

We also compare our best model, where we use Light GBM after features that are obtained from the fine-tuned DenseNet trained on the bright-preserved image, with the benchmark, considering in precision, recall, and F1-score. Results are shown in Table 5.6.

Table 5.6: Classification Report on IDRiD dataset

DR Level	Precision		Recall		F1 Score	
	Benchmark	Best Model	Benchmark	Best Model	Benchmark	Best Model
0	0.93	0.92	0.99	0.95	0.95	0.93
1	1.00	0.67	0.17	**0.40**	0.29	**0.50**
2	0.10	**0.62**	0.36	**0.71**	0.16	**0.66**
3	0.26	**0.36**	0.33	0.28	0.29	**0.31**
4	0.16	**0.42**	0.15	**0.42**	0.15	**0.46**

5.2.3 Computational Resources

For computational resources, we use 4 GeForce GTX 1070 GPUs, each loaded with 16G memory, spending 4 days in total to train the DenseNet 121 model and extract features. This process can be accelerated if we use green channel only images as input.

5.2.4 Achievement

This deep learning based pipeline (described in Figure 4.1) achieved third place in the ISBI 2018 Diabetic Retinopathy Segmentation and Grading Challenge. Our group has three members: Jeff Wu, from Cleerly Inc.; Ting Zhou, from University at Buffalo; and the author of this thesis, who led the effort discussed herein. We participated in two sub-challenges (Diabetic Retinopathy grading and Diabetic Macular Edema grading) of the Diabetic Retinopathy Segmentation and Grading Challenge. Jeff mainly focus on providing computational resources, environment configuration, and training DenseNet 121 with the author. Ting Zhou mainly focus on the whole DME prediction and he also contributed on finding image preprocessing methods. The author mainly contributed to the whole DR prediction, including data collection, data preprocessing (e.g., contrast enhancement, green channel extraction), blood vessels segmentation, DenseNet 121 training, Light GBM training and random search on Light GBM. For the DR grading challenge, Table 5.7 shows the detailed effort by members

of the group.

Table 5.7: Detailed Effort

Work	**Assignee**
Data Information Collection	Yu
Data Download	Yu
Image Morphological Closing Operation	Ting
Image Contrast Enhancement	Yu
Image Green Channel Extraction	Yu
Blood Vessel Segmentation	Yu
U-Net Training	Yu
Data Augmentation	Yu
DenseNet 121 Training	Wu &Yu
Environment Delopment	Wu
Feature Extraction	Yu
Light GBM Training	Yu
Random Search Optimization	Yu
5-Layer Neural Network Training	Yu
Computational Resources Providing	Wu

Chapter 6

Discussion

In this research, we propose a deep learning based pipeline for image grading of diabetic retinopathy. Results outperform the baseline method by a large margin. From our study, we observe the following.

First, preprocessing is commonly used in the medical image area, and is very important for feature extraction in diabetic retinopathy. Medical images have often deteriorated with high noise due to various interference sources in the measurement processes of the imaging and data acquisition systems. Improvement in appearance and visual quality of the images may assist in the interpretation of medical images, and may also affect the diagnostic decision. Preprocessing retina image can help us suppress unwanted (non-object) information and enhance wanted (object) information. Preprocessing retina images makes feature extraction easier and achieves better performance. From the experiment results in Chapter 5, we can observe that preprocessing has a huge impact on DR detection. In our experiment results on the IDRiD dataset, with preprocessing, our model can achieve 0.66 in accuracy and 0.84 in quadratic weighted kappa score; without much preprocessing, we can only achieve 0.55 in accuracy and 0.77 in quadratic weighted kappa score. In our experiment results on the

Kaggle dataset, with preprocessing, our model can achieve 0.65 in accuracy and 0.84 in quadratic weighted kappa score; without much preprocessing, our model can only achieve 0.49 in accuracy and 0.71 in quadratic weighted kappa score. In all our experiments, we clearly see a boost in performance with the preprocessing techniques.

When we compare experiment results on the IDRiD dataset with the Kaggle dataset, we find that in the Kaggle dataset, preprocessing work is even more important. That can be mainly explained by the relative order-of-magnitude difference between the Kaggle and IDRiD datasets in quantity. Additionally, the Kaggle dataset exhibits more diversity. In contrast, the IDRiD dataset contains images that are mostly collected under similar conditions.

Second, like other CNNs, DenseNet 121 is a good feature extractor, and works well especially for our pipeline. In the deeper layer of DenseNetn 121, it can provide us with a higher level of features of input images, which enable us to capture features related to DR. In our experiments, DenseNet 121 as the benchmark achieves 0.69 quadratic weighted kappa score for the IDRiD dataset, and 0.54 in Kaggle dataset, which is much better than other CNN architectures. Also, from our experiments, with some important preprocessing steps like brightness preserve and green channel extraction, DenseNet 121 can achieve even better performance than those without preprocessing. That means DenseNet 121 itself can learn some features, but with some expert knowledge about retinal images (as implemented through preprocessing), it is able to learn better features.

Third, lesions in the green channel are well represented. From our experiment, extracting only the green channel of a retina image after bright preserve in the diabetic retinopathy grading task still can achieve good performance. However, it may lose some information we need to detect diabetic retinopathy. In our experiment results, extracting only the green channel hurts the performance a little comparing to those keep all the channels. The results

between experiments (i.e., extracting green channel vs. using all of the channels) shows the information entropy that we may have lost.

Finally, the boosting decision tree algorithm works better in which features are well extracted than the 5-layer neural network, because it trains fast and gets better results. In all of our experiments, Light GBM gives better results as compared to the neural network. On the other hand, the neural network may be overfitting in small datasets. When training on the IDRiD dataset, there is a big difference with Light GBM. However, with the dataset becoming large, it is able to achieve performance comparable with Light GBM. When training on the Kaggle dataset, it generates a much better result, even a similar kappa score with Light GBM.

Chapter 7

Conclusions and Future Work

7.1 Summary

In this research, we build an integrated pipeline to automatically grade diabetic retinopathy. We use a dataset from the Kaggle diabetic retinopathy competition and the IDRiD database. First, we preprocess the image data to make it efficient to train, including contrast enhancement, standardization, and morphological closing operation. Then we augment the data with various transformations.

After that, we feed data into the DenseNet 121 architecture, which is the state-of-art architecture in many image classification tasks, to train a model as a benchmark. A resampling method is applied to alleviate problems caused by imbalanced data. For each image, features are extracted by randomly augmenting images 50 times, then extracting the outputs from the last second fully connected layer. The mean vector and standard deviation vector are calculated as features for each image. Then we apply a 5-layer neural network and LightGBM to train the model based on extracted features. Results show that our integrated

workflow outperforms the baseline method by a large margin in not only the IDRiD dataset but also the Kaggle dataset. This pipeline achieved third place in the ISBI 2018 Diabetic Retinopathy Grading challenge on April 4, 2018 in Washington, D.C.

7.2 Conclusions

In this work, we propose a novel deep learning based pipeline for grading diabetic retinopathy. Previous work in diabetic retinopathy detection and grading work mainly relies on expert knowledge and hand-made features.

As main contributions of this study, we propose a novel and efficiently integrated pipeline composed of the following steps:

- Retina image preprocessing
- Deep learning based feature extraction
- Light GBM and 5-layer neural network

Data preprocessing methods, like standardization, morphological closing operation, and contrast enhancement, are very useful to highlight the lesions of the retina and enable making retina images efficient for training. Compared to other pipelines without preprocessing work, those pipelines with preprocessing work achieved much better performance. Preprocessing considerably reduced noise in the retina image.

DenseNet as the state-of-art image classification architecture is a good fit for our pipeline. It uses hierarchical convolutional layers to describe some kinds of visual patterns and is able to strengthen feature propagation, and encourage feature reuse. Thus it achieves better performance compared to other CNN architectures on both the Kaggle dataset and the

IDRiD dataset.

Light GBM is a well optimized boosting decision tree and is able to learn the model from features to predict DR severity level. It uses a leaf-wise optimization strategy and is able to generate more complex trees to predict more accurate scores. A 5-layer neural network also works well when training and test sets are large enough.

We believe this whole pipeline is able to help people in real life to detect diabetic retinopathy, thus effectively preventing or delaying DR.

7.3 Future Work

We believe this pipeline can be further improved, when:

- More expert knowledge in DR can be applied.
- More image preprocessing methods can be applied.
- More labels on the retinal image can be used, like lesions segmentation labels.

In the future, we plan to learn more expert knowledge in DR, apply various image preprocessing methods, and get other labels or hire experts to manually label lesions for retina images. In that way, we believe we can extract better features, thus providing better results for people, and prevent or delay DR.

Bibliography

[1] (2015). Quadratic Weighted Kappa Score From Kaggle. https://www.kaggle.com/c/diabetic-retinopathy-detection#evaluation.

[2] (2015). Reliant Medical X-ray Equipment. http://reliantmed.com/x-ray/.

[3] (2015). What is a CT scan, Bupa.co,uk. https://www.bupa.co.uk/health-information/directory/c/ctscan.

[4] (2016). Siemens Magetom Symphony MRI Machine. https://www.indiamart.com/proddetail/siemens-magnetom-symphony-mri-machine-11881558248.html.

[5] (2017). Stanford CS231N 2017 CNN Architectures. http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture9.pdf.

[6] (2018). Diabetic Retinopathy Segmentation and Grading Challenge. https://idrid.grand-challenge.org/.

[7] (2018). Diabetic Retinopathy Segmentation and Grading Challenge Sub-challenge 1. https://idrid.grand-challenge.org/segmentation/.

[8] (2018). Diabetic Retinopathy Segmentation and Grading Challenge Sub-challenge 2. https://idrid.grand-challenge.org/grading/.

[9] (2018). Difference between 2D, 3D, 4D and latest 5D Ultrasound machines. https://enterpriseultrasound.com/difference-between-2d-3d-4d-and-latest-5d-ultrasound-machines/.

[10] Abràmoff, M. D., Lou, Y., Erginay, A., Clarida, W., Amelon, R., Folk, J. C., and Niemeijer, M. (2016). Improved automated detection of diabetic retinopathy on a publicly available dataset through integration of deep learning. *Investigative ophthalmology & visual science*, 57(13):5200–5206.

[11] Anavi, Y., Kogan, I., Gelbart, E., Geva, O., and Greenspan, H. (2015). A comparative study for chest radiograph image retrieval using binary texture and deep learning classification. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 2940–2943. IEEE.

[12] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007a). Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160.

[13] Bengio, Y., LeCun, Y., et al. (2007b). Scaling learning algorithms towards AI. *Large-scale kernel machines*, 34(5):1–41.

[14] Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

[15] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

[16] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). Classification and regression trees (cart) wadsworth. *Pacific Grove, CA*.

[17] Brosch, T., Tam, R., Initiative, A. D. N., et al. (2013). Manifold learning of brain mris by deep learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 633–640. Springer.

[18] Datta, N. S., Dutta, H. S., and Majumder, K. (2016). Brightness-preserving fuzzy contrast enhancement scheme for the detection and classification of diabetic retinopathy disease. *Journal of Medical Imaging*, 3(1):014502.

[19] Delalleau, O. and Bengio, Y. (2011). Shallow vs. deep sum-product networks. In *Advances in Neural Information Processing Systems*, pages 666–674.

[20] Fu, H., Xu, Y., Lin, S., Wong, D. W. K., and Liu, J. (2016a). Deepvessel: Retinal vessel segmentation via deep learning and conditional random field. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 132–139. Springer.

[21] Fu, H., Xu, Y., Wong, D. W. K., and Liu, J. (2016b). Retinal vessel segmentation via deep learning network and fully-connected conditional random fields. In *Biomedical Imaging (ISBI), 2016 IEEE 13th International Symposium on*, pages 698–701. IEEE.

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

[23] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

[24] Graham, B. (2015). Kaggle diabetic retinopathy detection competition report. *University of Warwick*.

[25] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[26] Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

[27] Huang, G., Liu, Z., Weinberger, K. Q., and van der Maaten, L. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, volume 1, page 3.

[28] Klein, R., Klein, B. E., Neider, M. W., Hubbard, L. D., Meuer, S. M., and Brothers, R. J. (1985). Diabetic retinopathy as detected using ophthalmoscopy, a nonmyciriatic camera and a standard fundus camera. *Ophthalmology*, 92(4):485–491.

[29] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[30] Lachure, J., Deorankar, A., Lachure, S., Gupta, S., and Jadhav, R. (2015). Diabetic retinopathy using morphological operations and machine learning. In *Advance Computing Conference (IACC), 2015 IEEE International*, pages 617–622. IEEE.

[31] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

[32] Li, R., Zhang, W., Suk, H.-I., Wang, L., Li, J., Shen, D., and Ji, S. (2014). Deep learning based imaging data completion for improved brain disease diagnosis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 305–312. Springer.

[33] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88.

[34] Montufar, G. F., Pascanu, R., Cho, K., and Bengio, Y. (2014). On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pages 2924–2932.

[35] Pascanu, R., Gulcehre, C., Cho, K., and Bengio, Y. (2013). How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*.

[36] Poultney, C., Chopra, S., Cun, Y. L., et al. (2007). Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems*, pages 1137–1144.

[37] Prasanna Porwal, Samiksha Pachade, R. K. M. K. G. D. V. S. and Meriaudeau, F. (2018). Indian diabetic retinopathy image dataset (idrid). IEEE Dataport.

[38] Quinlan, J. R. (1993). C4. 5: Programming for machine learning. *Morgan Kauffmann*, 38:48.

[39] Ravishankar, S., Jain, A., and Mittal, A. (2009). Automated feature extraction for early detection of diabetic retinopathy in fundus images. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 210–217. IEEE.

[40] Reimers, J., Gangaputra, S., Esser, B., Harding, T., Thayer, D., Peng, D., Hubbard, L., and Danis, R. (2010). Green channel vs. color retinal images for grading diabetic retinopathy in dcct/edic. *Investigative Ophthalmology & Visual Science*, 51(13):2285–2285.

[41] Ripley, B. D. (2007). *Pattern recognition and neural networks*. Cambridge university press.

[42] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for

biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

[43] Shiraishi, J., Katsuragawa, S., Ikezoe, J., Matsumoto, T., Kobayashi, T., Komatsu, K.-i., Matsui, M., Fujita, H., Kodera, Y., and Doi, K. (2000). Development of a digital image database for chest radiographs with and without a lung nodule: receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. *American Journal of Roentgenology*, 174(1):71–74.

[44] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

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

[46] 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. *IEEE transactions on medical imaging*, 23(4):501–509.

[47] Suk, H.-I., Lee, S.-W., Shen, D., Initiative, A. D. N., et al. (2015). Latent feature representation with stacked auto-encoder for ad/mci diagnosis. *Brain Structure and Function*, 220(2):841–859.

[48] Suk, H.-I. and Shen, D. (2016). Deep ensemble sparse regression network for alzheimers disease diagnosis. In *International Workshop on Machine Learning in Medical Imaging*, pages 113–121. Springer.

[49] Walter, T., Klein, J.-C., Massin, P., and Erginay, A. (2002). A contribution of image

processing to the diagnosis of diabetic retinopathy-detection of exudates in color fundus images of the human retina. *IEEE transactions on medical imaging*, 21(10):1236–1243.

[50] Wang, C., Elazab, A., Wu, J., and Hu, Q. (2017). Lung nodule classification using deep feature fusion in chest radiography. *Computerized Medical Imaging and Graphics*, 57:10–18.

[51] Yang, W., Chen, Y., Liu, Y., Zhong, L., Qin, G., Lu, Z., Feng, Q., and Chen, W. (2017). Cascade of multi-scale convolutional neural networks for bone suppression of chest radiographs in gradient domain. *Medical image analysis*, 35:421–433.