Project Report: Study on Machine Learning for Diabetic Retinopathy Disease Grading

Hani Ramadhan (201793254), Asep Muhammad Awaludin (201799141), and Fachriannoor (201883655)

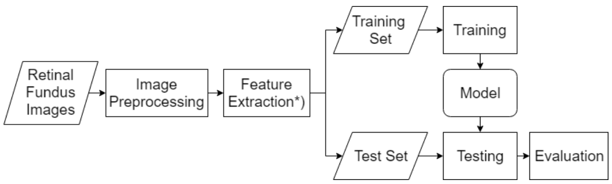

Fig. 1. Basic implementation flow for proposed method

I. INTRODUCTION

This report summarizes the result of the experiments conducted based on the project proposal of Study on Machine Learning for Diabetic Retinopathy Disease Grading. In this study report, we found that a method for retinal fundus image preprocessing and different machine learning classifiers affects the performance of Diabetic Retinopathy Grading in terms of error reduction and the time to predict the severity of diabetic retinopathy.

II. PROPOSED METHOD

Our basic work flow for the implementation of machine learning for Diabetic Retinopathy disease grading is presented in Figure 1. However, deep learning models will not perform feature extraction, thus they will directly train and test using after the image preprocessing step. Each process will be explained in the following subsections.

A. Image Preprocessing

Based on research that conducted in [1], we will use the Fractional Max-Pooling to obtain retinal fundus image that used for feature extraction. These are the steps that we perform in preprocessing phase:

1) Rescale the images to have the same radius (300 pixels or 500 pixels)
2) Subtract the local average color and map into 50% gray
3) Clip the images to 90% size to remove the boundary

B. Feature Extraction

After the image preprocessing step, we will try to extract attributes that explains the image instead of raw pixels. In this case, we will use combination of Principal Component Analysis (PCA) [2], t-Distributed Stochastic Neighbor Embedding (t-SNE) [3], and their combination.

Both PCA and t-SNE will reduce the features from the full dimension of the images to only a handful meaningful features. Thus, these extracted features capture the characteristics of the diabetic retinopathy severity.

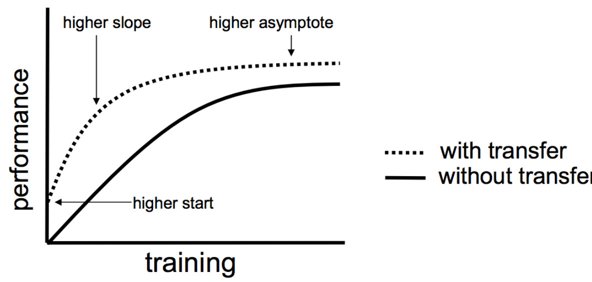

Fig. 2. Performance Comparison of Transfer Learning

C. Non-Deep Learning-Machine Learning Methods

We will perform ensemble machine learning method for the diabetic retinopathy using previously extracted features. We will use combination of two different classifiers: Support Vector Machine [4] and Decision Trees [5]. We extend those classifiers to be more robust using bagging style ensemble methods [6]. This ensemble method will assign the predicted label of each instance by the majority vote of its member classifiers.

D. Convolutional Neural Network

Finally we will examine the Diabetic Retinopathy dataset using Convolutional Neural Networks (CNN). It has been proved that CNN is the most suitable deep learning architecture for image classification.

Instead of training a deep network from scratch, we will use a model trained on a different domain (such as ImageNet) and adapt it for Diabetic Retinopathy dataset. Pre-trained model has learned to pick out features from images that are useful in distinguishing one image (class) from another. The network contains convolution blocks with activation on the top layer that defines complex functional mappings between inputs and response variable, followed by batch normalization after each convolution layer [7]. Figure 2 shows how transfer learning approach achieves a better performance compare to traditional model. Moreover, it reduces significantly the training process since we don't need to train all component within the network.

III. DATASET

The dataset used in this research is Indian Diabetic Retinopathy Image Dataset [8] of Diabetic Retinopathy grading. This dataset contains 516 images with 5 different severity of DR, which 0 stands for no DR and 4 for severe DR. Those 516 images are separated into 413 images of

TABLE I

DISTRIBUTION OF THE DATASET'S LABELS

Set	Total	DR Severity				
		0	1	2	3	4
Training	413	134	20	136	74	49
Test	103	34	5	32	19	13

TABLE II

PARAMETERS FOR NON-DEEP LEARNING METHODS

Parameters	Values
Dataset	Original,**Preprocessed**
Bagging	**Yes**, No
Base classifier	Decision Tree, **Support Vector Machine**
Feature Extraction	PCA (10), **t-SNE (100)**
Number of Features	2,10,20,50,100,200,400

training set and 103 test set. The distribution of the ground truth for the dataset is depicted in Table I.

IV. EXPERIMENTS

In the experiments, we measure the performance of the studied methods using two metrics: running time and Mean Squared Error (MSE). The running time is defined by the sum of feature extraction time, training time, and prediction time. However, because of our small dataset, we ignore the prediction time as it is very small. Then, MSE will measure how far a single predicted label $\hat{y}_j$ to its ground truth y_j are apart. The formula of MSE for dataset with N instances is described by Eq (1). We use MSE as the labels have the value from 0 to 4 where each value corresponds to severity in the term of rank.

$$MSE = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2 \qquad (1)$$

Table II and III summarizes the parameters for the non-deep learning and deep learning method experiments. The parameters typed in bold refers to default variation. Note that as we have 413 images for training and 103 images for testing, the PCA only extract up to 100 features because it cannot extract features exceeding the number of rows or columns in a dataset (in this case the each pixels is considered a single column). In deep learning setup, we use 0.01 learning rate with 1000 training steps which each step picks randomly 100 samples from training dataset. We will repeat the experiment of a single combination of the parameters 5 times, then show the averaged result.

A. Preprocessing result

Before processed in feature extraction, we preprocess the original JPG image using the Fractional Max-Pooling.

TABLE III

HYPERPARAMETER OF DEEP LEARNING METHODS

Hyperparameters	Values
Training Steps	1000
Batch Size	100 Images
Learning Rate	0.01
Training Setup	70% Training, 30% Validation

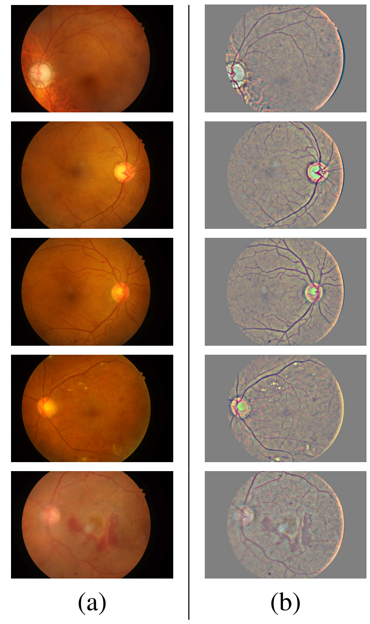

Fig. 3. Comparison of the original images (a) and preprocessed images (b)

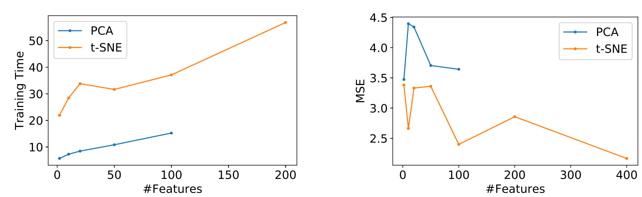

Fig. 4. The comparison between the training time (left-side) and the MSE (right-side) of feature extraction methods

Figure 3 shows the difference between images of different severity before and after preprocessing. The order of severity represented in the increasing fashion, where top row depicts the level 0 severity and the last row refers to level 4 severity. This preprocess takes 97.80 seconds to produce the image that used for the next phase.

B. Feature Extraction

We compare the performance of PCA and t-SNE in terms of the training time and the Mean Squared Error in Figure 4. We also include the training time of the transfer learning of Convolutional Neural Network. We see that the PCA has shorter training time but has higher MSE when compared to t-SNE. We omit the training time of t-SNE because it exceeds 200 seconds. We see that the MSE is not likely to be related to the number of extracted features for both PCA and t-SNE. Thus, we may safely set the number of features for t-SNE and PCA as 100 and 10 respectively as default for the other experiments' report. We do not choose t-SNE with 400 features as it needs long time to be extracted, which is not convenient for future use.

C. Effects of Preprocessing

We fix the number of features for each extraction method according to previous results, then measure the effects of

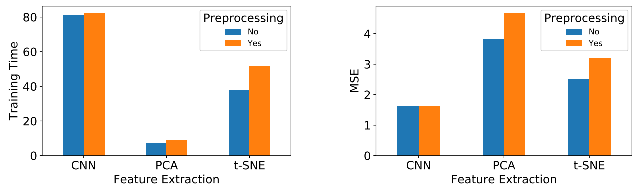

Fig. 5. The comparison between the training time (left-side) and the MSE (right-side) in effects of preprocessing

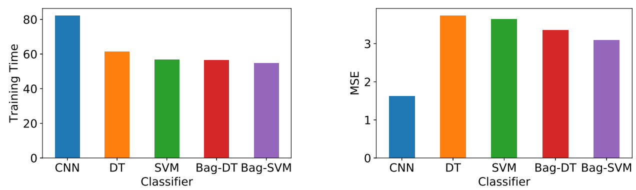

Fig. 6. The comparison between the training time (left-side) and the MSE (right-side) of different classifiers

proposed preprocessing by comparing the MSE of simple RGB-to-grayscale color conversion for the non-deep learning method and original RGB image for deep learning method to the preprocessed image for both classification methods as input. Figure 5 shows the result of the comparison. We can conclude that in non-deep learning, the proposed preprocessing method does not give significant improvement to the prediction quality. However, the CNN's prediction performance seems not very affected with the proposed preprocessing method. In addition, the preprocessed images lead to longer training time, which means makes the training harder to converge. Then, based on this experiment, Fractional Max-Pooling is not compatible to PCA, t-SNE, and convolution of CNN feature extraction method in such case of retinal fundus image.

D. *Comparison of the Classifiers*

Thus, in this part, we compare the MSE of different classifier scenario cases: Bagged Decision Tree (Bag-DT), Bagged Support Vector Machine (Bag-SVM), single Decision Tree (DT), Single SVM (SVM), and transfer learning of CNN (CNN) in Figure 6. It is very noticeable that transfer-CNN sharply outperformed other classifiers. Although CNN has worst training time, it is acceptable to have a highly accurate method to predict the severity of the diabetic retinopathy given the retinal fundus image dataset.

V. Conclusion

In this paper, we studied the effects of a preprocessing method, some non-deep learning classification method combined with several feature extraction scenarios, and pre-trained Convolutional Neural Network to predict the severity of diabetic retinopathy in retinal fundus images. This study is motivated by the high cost of experts to detect the diabetic retinopathy from retina screening of patient. Benefiting from the computing power, this cost can be highly reduced by the robustness of machine learning. In time less than a second, machine learning can predict the severity of diabetic retinopathy of a patient.

Based on the experiment result, the combination of Fractional Max Pooling and PCA/t-SNE feature extraction did not give beneficial improvement to prediction. This gives chance explore the compatibility of this preprocessing method to another feature extraction, such as Histogram of Gradient (HoG) or Histogram of Color Difference (CDH). While the training time of the non-deep learning method is much shorter than the deep learning method, the accuracy of deep learning method is still superior than the non-deep learning method. Even though so, the application of ensemble learning, e.g. the bagging method, improves the diabetic retinopathy prediction quality of non-deep learning cases.

References

[1] Gwenol Quellec, Katia Charrière, Yassine Boudi, Batrice Cochener, and Mathieu Lamard. Deep image mining for diabetic retinopathy screening. *Medical Image Analysis*, 39:178–193, 7 2017.

[2] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1-3):37–52, 8 1987.

[3] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[4] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 7 1998.

[5] Thomas G. Dietterich. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Machine Learning*, 40(2):139–157, 2000.

[6] Somasundaram S K and Alli P. A Machine Learning Ensemble Classifier for Early Prediction of Diabetic Retinopathy. *Journal of Medical Systems*, 41(12):201, 12 2017.

[7] Carson Lam, Darvin Yi, Margaret Guo, and Tony Lindsey. Automated Detection of Diabetic Retinopathy using Deep Learning. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, 2017:147–155, 2018.

[8] Prasanna Porwal, Samiksha Pachade, Ravi Kamble, Manesh Kokare, Girish Deshmukh, Vivek Sahasrabuddhe, and Fabrice Meriaudeau. Indian Diabetic Retinopathy Image Dataset (IDRiD): A Database for Diabetic Retinopathy Screening Research. *Data*, 3(3):25, 2018.