Diabetic Retinopathy Detection through EfficientNet and Transfer Learning: Final Report

()

Abstract

Diabetic retinopathy(DR) is one of the most severe retinal disease that could lead to permanant blindness. DR has five stages including No DR, Mild DR, Moderate DR, Severe DR and Proliferate DR. The most effective way to avoid DR is early detection of DR stages, which is extremely important for treatment. However, traditional way of DR identification requires manual intervention, which relies on the clinician's experience and expertise and is time-consuming due to complexity of DR images. Therefore, an automatic and reliable detection method is needed. Convolutional neural network(CNN) has been the state-of-art in image classification problems. In this report, we use EfficientNet, which is a recently proposed CNN, combined with transfer learning as well as data preprocessing and weighted random sampling as the deep learning method. After comparing multiple networks, EfficientNet-B4 outperforms all the other networks and obtains a quadratic weighted kappa score(QWK) of 0.925 on validation set from APTOS DR dataset.

1. Introduction

Diabetic retinopathy(DR) is a devastating retinal disease as well as a dangerous complication of diabetes, which has become one of the major sources of permannat blindness. Regular check is suggested by medical experts since early detection and diagnosis is very important to avoid and cure DR(Benbassat & Polak, 2009).

Currently, the diagnosis of DR is performed by manual evaluation of retinal images by trained professional clinicians. Clinicians diagnose DR into 5 classes: normal, mild, moderate, severe and proliferative(Wilkinson et al., 2003) by detecting the features such as microaneurysms, exudates, hemorrhages and their relative position, which is shown in Fig. 1(Abdullah et al., 2016). However, this process relies too much on the clinician's experience and is time-consuming due to the complexity of structure of the lesions.

Therefore, the motivation of our work is to find an effective method to automatically diagnose DR into five classes by analysing the retinal images with high efficiency and reliability.

Many researchers have made contribution to this problem using traditional machine learning methods. For example, (Priya & Aruna, 2012) proposes SVM to classify DR images and obtains an accuracy of 97.608%. (Chetoui et al., 2018) uses Local Energy-based Shape Histogram (LESH) combined with SVM and achieves an accuracy of 90.4%.

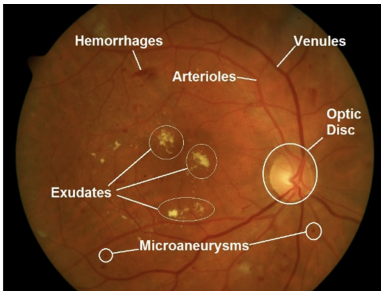


Figure 1. Important features in retinal fundus image.

Over the past decade, breakthrough in Deep Learning(Fukushima & Miyake, 1982) has made tremendous contribution to the area of computer vision especially in classification tasks. Convolutional neural networks (CNN)(Krizhevsky et al., 2012) are proved to outperform classical machine learning methods as well as other neural networks in deep learning. Many CNNs architectures have been proposed in recent years such as VggNet (Simonyan & Zisserman, 2014), Densenet(Iandola et al., 2014) and ResNet (He et al., 2016). More and more researchers focus on CNN to solve this problem. (Mookiah et al., 2013) shows that the applications of CNN are particularly useful in classifying DR images into different classes with encouraging results. (Pratt et al., 2016) develops a deep CNN and obtains an accuracy of 75% on 5000 validation images. (Hagos & Kant, 2019) adopts transfer learning using pretrained InceptionNet V3 and achieves an accuracy of 90.9%.

In our project, we adopt a recent CNN architecture EfficientNet proposed by(Tan & Le, 2019) to detect DR in publicly available datasets on kaggle: one is APTOS(APTOS, 2019) and another is EyePACS(Bhaskaranand et al., 2015). EfficientNet gains popularity because it is 8.4x smaller and 6.1x faster than other existing CNN archtectures(Chetoui & Akhloufi, 2020). However, APTOS is a small dataset including only 5590 images, which limits the performance of our network. Therefore, instead of training directly on (APTOS, 2019), we use transfer learning to pretrain the

model on another public DR dataset EyePACS, which has 35126 images and is much larger than APTOS. Then we use the pretrained model to train on APTOS. The evaluation metric that we choose is Quadratic Weighted Kappa coefficient(QWK)(Brenner & Kliebsch, 1996). Combined with preprocessing, data augmentation and weighted random sampling, our best model is EfficientNet-B4 which achieves a QWK of 0.925 on validation set.

Our contributions are summarized as follows:

- We take effective preprocessing steps such as cropping dark area, converting the images to grayscale, blurring the Gaussian filter and uniforming the size. With preprocessing, the QWK of our baseline is improved from 0.887 to 0.891 and training is much faster.
- We compare the whole family of EfficientNets as well as many other CNN models and obtain the best model to be EfficientNet-B4.
- We use transfer learning to pretrain model on EyePACS and apply the the information on APTOS. The QWK is improved from 0.910 to 0.915.
- We use weighted random sampling to resample the dataset and largely alleviate the problem of class imbalance without hurting the performance.
- We use Nelder-mead method and improve the QWK from 0.915 to 0.925.

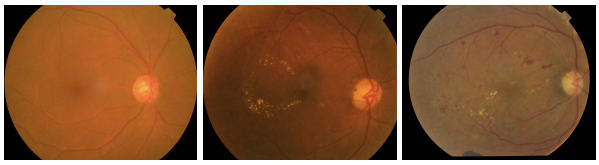
2. Data set and task

2.1. Datasets

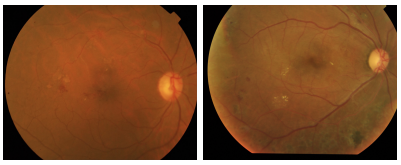
There are many publicly available datasets for detecting DR from the retina. We use two data sets with high-resolution colour images.

EyePACs was released in Kaggle Diabetic Retinopathy Detection Challenge 2015, which consists of 35126 fundus images. The images are divided into five classes (0, 1, 2, 3, 4) by the severity of DR according to the International Clinical Diabetic Retinopathy severity scale(Wilkinson et al., 2003):

- 0: Normal, no DR
- 1: Mild DR, in which only microaneurysms might happen;
- 2: Moderate DR, where blood transportation by bloode vessels stops working.
- 3: Severe DR, in which continuous blocking of blood vessels leads to impaired ability of blood supply to the retina;
- 4: Proliferative DR, where tiny blood vessels proliferate from the retina surface.



(a) Normal (b) Mild DR (c) Moderate DR

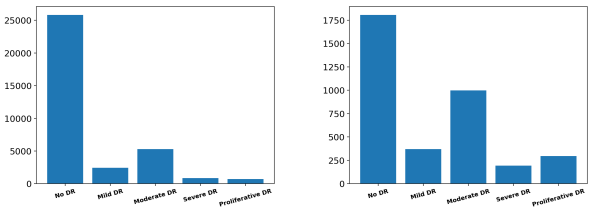


(d) Severe DR (e) Proliferative DR

Figure 2. Classes of diabetic retinopathy with increasing severity

The example of each class image is displayed in Fig. 2.

APTOS was released in 2019 by the Asia Pacific Tele-Ophthalmology Society, which includes 5590 fundus images including 3662 training images and 1928 test images. The clinicians have rated the presence of DR in each image on a scale of 0 to 4 which is the same as the EyePACS dataset. And the class distribution of these two datasets is shown in Fig. 3. Our dataset suffers from class imbalance.



(a) EyePACs in 2015 (b) APTOS in 2019

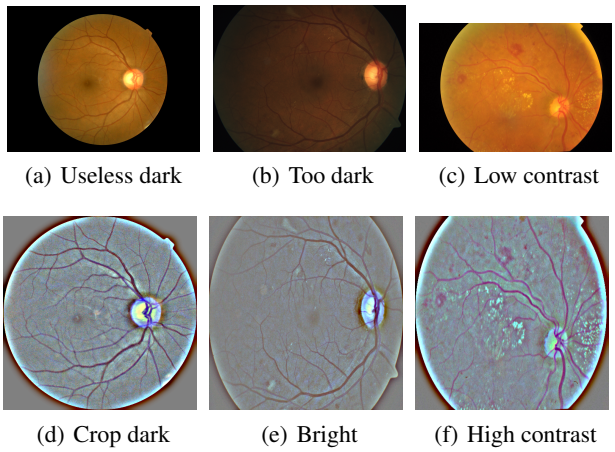
Figure 3. Classes distribution in datasets

2.2. Data Preprocessing

There is no doubt that the visible features in Fig. 1 are essential to our task. However, the images in the datasets come from different types of cameras, operated by specialists who have varying levels of experience and under a variety of lighting conditions, which results in a large variation in image quality(Rakhlin, 2018). In detail, several images with low quality are shown in the first row of Fig. 4. There are many dark areas around retina, which are uninformative to our task and shown in Fig. 4(a). As shown in Fig. 4(b), due to bad light conditions, the image is too dark to view the features. And in Fig. 4(c), it is difficult to distinguish the features because of low contrast. Also, the images are not uniform in size.

To address these issues, we use the following preprocessing methods:

(1) Cropping the dark area, and the cropping boundary is determined by judging the pixels in a column that are



(a) Useless dark (b) Too dark (c) Low contrast

(d) Crop dark (e) Bright (f) High contrast

Figure 4. Examples of image after preprocessing.

all black. Since the RGB of black is 000, we remove the columns of pixels where the values are all 000 in the image data.

(2) Converting the images to gray scale. It reduces the amount of data we process since the grayscale has only one layer image from 0-255 whereas the RGB has three layers.

(3) To reduce the correlations between the features in many images, we use the Gaussian blur(Hummel et al., 1987). We let the image blurred by Gaussian function and combine the output with the original image as our new image. In detail, the Gaussian standard deviation is set to 51.2(Image size/10). The weight added for the original image is 4 while the weight for the image blurred by Gaussian function is set to -4, and the scalar added to each sum is set to 128(half of the max grayscale).

(4) Resizing all the images to the same size. In this case, we set the size as 512×512 pixels.

(Sisodia et al., 2017) found that the images after preprocessing have a higher mean value and standard deviation. And we can find that Fig. 4(d) has cropped the dark area that contains useless information compared with Fig. 4(a). Fig. 4(e) have better the light condition than Fig. 4(b) and the features of Fig. 4(f) are more clear than Fig. 4(c) because of high contrast.

2.3. Evaluation metric

We use Quadratic Weighted Kappa coefficient(QWK)(Brenner & Kliebsch, 1996) as the evaluation metric. It is calculated between the labelled results and predicted results and defined as 1:

$$\kappa = 1 - \frac{\sum_{i=1}^{k} \sum_{k}^{j=1} w_{ij} o_{ij}}{\sum_{i=1}^{k} \sum_{k}^{j=1} w_{ij} e_{ij}} \tag{1}$$

where κ is the number of categories, o_{ij} are the predicted labels and e_{ij} are the expected labels. w_{ij} is calculated as 2:

$$w_{ij} = \frac{(i-j)^2}{(k-1)^2} \tag{2}$$

The performance of this metric will be good when dealing with imbalanced datasets and useful for measuring inter-rater agreement for categorical classification where the raters are the human-labeled dataset and the neural network predictions(Ben-David, 2008).

2.4. Task

Our task is to develop a reliable and effective CNN to classify DR images from APTOS automatically instead of manual evaluation. To be more specific, our task can be divided into many sub-tasks. The first one is to develop a robust baseline with proper parameter settings and compare it with many other different models to find the model that outperforms the others. The second one is to apply transfer learning to our dataset and find whether it helps improve the performance. The third one is to resample our images to alleviate the problem of class imbalance.

3. Methodology

3.1. Data augmentation

It is well known that a machine learning algorithm can be effective with rich data. (Buslaev et al., 2020) has shown that data augmentation can increase the size of training sets by applying input transformations that obtain the corresponding output values, which can improve the performance of the classification task. We use augmentations methods including flipping, rotating and zooming the input images(Fastai, 2021). In detail, before training the images, we flip the images vertically or rotate the images by 90 degrees with the same probability 0.5. And we use the random zoom by scaling between 1 and 1.2, where the focal point is set to the center of original images.

3.2. Weighted random sampling

Class imbalance is a typical problem in image classification problems(Buda et al., 2017). Our dataset is also extremely imbalanced. Many methods have been developed to solve this problem. The most common methods include oversampling and undersampling. However, these two methods both need to change the distribution of classes. Instead we use weighted random sampling, which does not change the dataset. To use weighted random sampling, we first need to calculate weight of each class, which equals 1 divided by the count of images belonging to this class. Then take the weights and number of training samples as the parameters of multinomial distribution and simulate a new training dataset which is more balanced. The algorithm of weighted random sampler is presented in algorithm 1.

Algorithm 1 Weighted random sampling

Input: $class\_sample\_count, num\_sample$
$weights = \frac{1}{class\_sample\_count}$
$new\_sample = multinomial(weights, num\_sample)$
train on the new sample

3.3. Transfer learning

(Yosinski et al., 2014) has stated that when we train the neural network, the first layer features appear not to be specific to a particular dataset or task, and they can be transferred to train other different dataset or pictures. They also pointed out that if the base data is much bigger than the target data, transfer learning can perfectly work in training target network with preventing the target network from overfitting.

As we can see from the distribution of data 2019 in 2, class 0 accounts for the majority, which shows that our data is distributed unevenly. There are too few labels especially for class 1, 3 and 4. In addition, the dataset of 2019 is small which might easily lead to overfitting. Therefore We decided to pretrain the network with larger 2015 dataset which contains 35126 pictures. Transfer learning can work because the features of the diabetic retinopathy are not dependent from different data sets and bigger datasets can alleviate the instrument noise. In addition, the dataset of 2019 is not big enough to train complex networks and we need to apply transfer learning to prevent it from overfitting.

3.4. EfficientNet

We choose the EfficientNet as our neural network to balance the newwork depth,width and resolution. (Tan & Le, 2019) has proved that model EfficientNet-B7 has achieved the best accuracy on 84.38.4 times smaller and 6.1 times faster than the best traditional ConvNet. As we own limited computing resources and the dataset is relatively large, we choose to use this network to both train the network efficiently and achieve high classification accuracy.

The paper stated that only scaling up one factor of width, depth and resolution does not necessarily improve the accuracy and the accuracy gain may disappear when the model get bigger. For example, when we scale up the depth of network, the gradient decent will occur and the accuracy gain will diminish when we train very deep network. This paper then concluded that we can only get better performance when we balance all dimensions of width, depth, and resolution in a principled way.

The efficientNet firstly proposes a baseline called EfficientNet-B0. Efficient-B0 y is architectured by leveraging a multi-objective neural architecture search that optimizes both accuracy and FLOPS. It consists of multiple mobile inverted bottleneck MBConv with mobile inverted bottleneck optimization. It then scales up network Depth,width and resolution with the compound method below. The coefficient varies from different efficientNet. α, β, γ are constants which can be achieved by a small grid search. Then it scales up the network with different ϕ to get EfficientNet B0 to B7.

$$\begin{array}{ll} \text{depth: } d = \alpha^\phi & \\ \text{width: } w = \beta^\phi & \text{s.t. } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\ \text{resolution:} w = \gamma^\phi & \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \end{array}$$

Intuitively, the number of parameters of the networks increase from B0 to B7, which means it will take longer time to train and it is more difficult to train.

3.5. Cyclical learning rates

Learning rate is one of the most important parameters in tuning deep learning networks. Too small learning rate will lead to slow converge of training while too big learning rate can break the training and make it diverge(Smith, 2015). Choosing a proper learning rate usually requires a variety of "trials and errors". To avoid pesky guessing and trials of learning rate, we choose cyclical learning rates(CLR) which is firstly proposed by (Smith, 2015). Instead of adopting a fixed learning rate or exponentially decreasing learning rate, CLR enables the learning rate to cyclically vary between two boundaries. To apply CLR to the training, we need to set the minimum and maximum boundaries and a *stepsize*. A cycle includes two steps. In the first step, the learning rate increases linearly while in the second step, the learning rate decreases linearly. $Stepsize$ refers to the number of iterations in each step. The number of iterations of an epoch can be calculated by *batchsize* dividing number of images. According to (Smith, 2015), a good *stepsize* could be 2 to 10 times the number of iterations for each epoch. To get reasonable minimum and maximum boundaries, we use "LR range test"(Smith, 2015). In LR range test, we initiate the learning rate to be a very small number close to 0 and increase the learning rate slowly and linearly to explore how the accuracy changes with a range of learning rates. When we start increasing the learning rate from beginning, the accuracy increases. However, when the learning rate becomes too large, the accuracy starts to decline. The learning rate at the peak of accuracy is set to be the maximum boundary and the minimum boundary is set to be one third or one fourth of the maximum boundary.

3.6. One cycle policy

In LR range test by(Smith, 2015), the accuracy increases with a small learning rate while decreases as learning rate increases to a relatively large number. This is true when the range of learning rates is small. However, according the experiment by(Smith & Topin, 2017), the accuracy remains high consistently when the order of magnitude of learning rate is much larger than typical learning rate. Besides, the model is trained much faster with large learning rate. This phenomenon is called super convergence(Smith & Topin, 2017), which combines both speed and accuracy and is the goal that we pursue for training process. We use a method

named "one cycle policy"(Smith & Topin, 2017) which only includes two steps that are smaller than the total number of iterations, one of which increases from minimum to maximum and another decreases from maximum to minimum and even lower than the initial learning rate for the rest of the iterations(Smith & Topin, 2017).

4. Experiments

This section consists of a few significant experiments.

First experiment is performed on training before and after data preprocessing to see whether and how data preprocessing can improve training. In the second experiment, we expect to explore the whole family of EfficientNets and compare them with other models such as DenseNet, ResNet, VGG_11 and VGG_16 and find the best model. Moreover, we explore whether transfer learning from a larger dataset can produce a better result. Then we experiment on whether weighted random sampling works in alleviating class imbalance of our dataset. Lastly, we also explore how Nelder-mead method could help improve the network.

4.1. Baseline and parameter setting

Inspired by (Tymchenko et al., 2020), we set the baseline to be EfficientNet-B4. We initialize the weights of baseline from pretrained CNN on ImageNet. We split the dataset into training set with 2929 images and validation set with 733 images. The image size is 256×256 and the batch size is 32. We us LR range test to select the learning rate with steepest gradient as suggested by fastai from the plot of learning rate versus loss. Then we use this learning rate as the maximum learning rate *lr_max* and use "one cycle policy" to calculate the starting learning rate as $\frac{lr\_max}{div}$ and minimum learning rate as $\frac{lr\_max}{div\_final}$. *div* is initiated to be 25 and div_final is initiated to be 100000. The method of parameter setting applies to all the models discussed later.

4.2. Training before and after data preprocessing

In this experiment, we compare the training of the baseline before and after data preprocessing to learn whether and how data preprocessing can help improve training process and results. We start by training our baseline without data preprocessing. After LR range test, we find the suggested maximum learning rate to be 1.00×10^{-2}. We set the number of training epochs to be 20 and use early stopping to stop the training when network converges. Then we use the same training method to apply to baseline after data preprocessing. The result is presented in table 1. It's clear that after preprocessing, the baseline is trained faster and QWK also increases.

After preprocessing, some useless features in the former images have been filtered which will prevent the network from being affected by the noise, which can also make the training process faster. In addition, the high contrast makes it possible to learn important features. For example, (Foracchia et al., 2004) has pointed out that the Optic Disc is a very important feature and it will be easier to be detected when the image is of high contrast.

Preprocessed	Training loss	Validation loss	QWK	Training time(min:sec)
No	0.275	0.357	0.887	22:16
Yes	0.313	0.278	0.891	02:52

Table 1. Baseline before preprocessing vs after preprocessing

4.3. EfficientNets and other networks

In this experiment, we explore an extensive range of networks within the family of EfficientNets as well as the other networks to compare them with the baseline and find the best network. The models that we explore include EfficientNet-B0 to EfficientNet-B6, DenseNet121, ResNet18, VGG_11 and VGG_16. We initialize the weights of all these networks using pretrained parameters from ImageNet. For each model, we use the same method as before to select the learning rate. All training images are preprocessed in this experiment. The results are presented in table 2. The best model is EfficientNet-B4 with highest QWK of 0.910.

As we can see from table2, what is interesting is that the QWT value does not increase when the efficientNet get deeper and wider from EfficientNet-B0 to EfficientNet-b6. Intuitively, the wider, deeper networks can learn more complex features and achieve higher accuracy. But the results show that we can not achieve even higher accuracy when the network get more complex(the final QWK of EfficientNet-B5 and EfficientNet-B6 does not achieve better performance than EfficientNet-B4). It proves that we can not fully solve the problem of scaling networks with compound scale methods. It can only get relatively high results than networks which are scaled with only one dimension.

Models	Training loss	Validation loss	QWK
EfficientNet-B0	3.063	2.823	0.000
EfficientNet-B1	0.220	0.285	0.895
EfficientNet-B2	0.325	0.343	0.883
EfficientNet-B3	0.089	0.263	0.909
Baseline	0.120	0.277	**0.910**
EfficientNet-B5	0.322	0.313	0.884
EfficientNet-B6	0.370	0.317	0.878
ResNet18	2.362	2.261	0.487
DenseNet121	0.413	0.424	0.853
VGG_11	0.406	0.373	0.861
VGG_16	0.379	0.377	0.859

Table 2. Results of different models

4.4. Transfer learning from EyePACs

The dataset APTOS is not large enough and class imbalanced. Therefore, we explore whether pretraining the larger dataset EyePACS and using transfer learning to train on APTOS would help improve the performance. To pretrain EyePACS, we set the same parameters as we did on APTOS. Then we apply the pretrained weights on the EfficientNet-B4, which the is best model that we get. The result shows that the QWK of the model is 0.915, which is higher than the QWK of EfficientNet-B4. The training loss is 0.084 and validation loss is 0.251.

The transfer learning can work because the dataset of EyePACS is very similar to our dataset. The features the network learns from the first few layers can also be applied to our dataset. In addition, learning from much larger dataset can prevent our target network from overfitting.

4.5. Finding Optimized Threshold

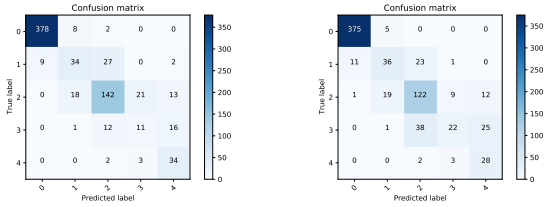
After we train the network, we get the regression results range from 0 to 4. It is necessary to find the optimal threshold to map the regression results to certain class.We try to search the best threshold values using nelder-mead optimization. Nelder-mead method helps to find the minimum of the loss function. We initiate the thresholds as(0.5,1.5,2.5,3.5) and search the optimized threshold value near the initiated value.

In our experiment, The final threshold values we got was (0.530074 1.495352 2.464067 3.267254) which finally improved the QWK value from 0.915 to 0.925.

4.6. Before and after weighted random sampling

Class imbalance is a big problem in our dataset, which is shown in figure 2 where numbers of class 0 and 2 are overwhelmingly larger than the other classes. We use weighted random sampling to resample the dataset and explore whether it can help alleviate this problem. The weight of a class is calculated by 1 divided by the number of images in the class. Then we use the weights to resample the dataset into a more balanced one. We present the confusion matrix in figure 5 before and after weighted random sampling. Before sampling, $\frac{378}{388}$ of images in class 0, $\frac{142}{194}$ of images in class 2 and $\frac{34}{39}$ of images in class 4 are correctly labeled. In contrast, the ratio is $\frac{34}{72}$ in class 1 and only $\frac{11}{40}$ in class 3. The confusion matrix shows that our classification is imbalanced between different classes. After sampling, the ratios of correctly labeled images in class 0, 2 and 4 all get lower. However, for class 1 and 3, the ratios have increased significantly especially in class 3, where the ratio increases from $\frac{11}{40}$ to $\frac{22}{40}$. Therefore, after weighted random sampling, problem of class imbalance is alleviated. The reason why weighted random sampling helps is that this method ensures that the large classes have a lower probability to be chosen while small classes have higher probability to be selected based on the weights of all the classes. We can interpret that weighted random sampling balance the classification by sacrificing large classes and helping small classes.

Moreover, we wonder whether weighted random sampling would affect QWK. Surprisingly, we train the model after sampling and find that the QWK remains unchanged. Therefore, we can conclude that weighted random sampling can alleveiate class imbalance without hurting QWK in our network.



(a) Before resampling (b) After resampling

Figure 5. Before resampling vs after resampling

5. Related work

Many researchers have been contributing to diabetic retinopathy detection.

Before deep learning becomes popular, many researchers have been focusing on traditional machine learning methods and computer vision techniques to deal with this problem. (Priya & Aruna, 2012) proposes a machine learning and computer vision based method to classify diabetic retinopathy using Probabilistic Neural network(PNN) and Support vector machine(SVM). In comparison of these two models, SVM achieves an accuracy of 97.608%, outperforming PNN with an accuracy of 89.60%. Another research (Chetoui et al., 2018) uses Local Energy-based Shape Histogram (LESH) combined with SVM with a Radial Basis Function kernel and achieves an accuracy of 0904 and AUC of 0.931. Although these are all traditional methods, many of them actually achieve high accuracy and are still worth exploring.

As deep learning gains increasing popularity recent years, many deep learning methods have been proposed by researchers on diabetic retinopathy detection. For example, (Pratt et al., 2016) develops a deep convolutional neural network combined with data augmentation and preprocessing and obtains an accuracy of 75% and a sensitivity of 95% on 5000 validation images. (Hagos & Kant, 2019) adopts transfer learning using pretrained InceptionNet V3 on ImgaeNet and obtains an accuracy of 90.9%, which is inspiring for us to think about transfer learning in our network.

A recent successful and inspiring research is done by (Tymchenko et al., 2020), which uses deep ensemble to combine 3 CNN models including EfficientNet-B4, EfficientNet-B5 and SE-ResNeXt50. Besides, they propose transfer learning using the pretrained network on EyePACs and apply it on APTOS. This is a robust and effective network which can learn meaningful features even in a small and noisy

dataset. The best model they achieved is the ensemble of 20 models with a high QWK of 0.986 on the validation set. The network ranked 54 of 2943 competitors with a high QWK of 0.925 on kaggle. Many ideas in the paper are really inspiring for our work such as transfer learning from larger similar dataset. Deep ensemble is an attractive method which has not been applied in our work and deserves to be explored in our future work.

6. Conclusions

In this report, we successfully build a deep learning model to automatically detect the stages of diabetic retinopathy on APTOS DR dataset without manual intervention. The baseline that we use is EfficientNet-B4 and the evaluation metric is Quadratic Weighted Kappa score(QWK). We start by data preprocessing and prove that it helps increase QWK and make training faster. Then we make extensive comparisons between other models and find that EfficientNet-B4 outperforms all the other models with a QWK of 0.910. Due to small size of APTOS, we use transfer learning by pretraining EfficientNet-B4 on a larger dataset EyesPACS and applying the information to APTOS. With transfer learning, the QWK increases from 0.910 to 0.915. Moreover, we use weighted random sampling to alleviate the problem of class imbalance in our dataset without hurting QWK. Finally, to minimize the effect of threshold, we use the Nelder-mead method to find thresholds value to minimize the overall loss and successfully increase the QWK from 0.915 to 0.925.

There is still much room for improvement in our future work. For example, we can explore ensembling multiple networks to further improve our network. What's more, using hyperparameter search to find the optimal parameter setting is also a research direction that deserves to be explored. Finally, although weighted random sampling can help balance the classification, class imbalance is still a serious problem. We expect to try more methods to address class imbalance such as using more advanced classifier.

References

Abdullah, Muhammad, Fraz, Muhammad Moazam, and Barman, Sarah A. Localization and segmentation of optic disc in retinal images using circular hough transform and grow-cut algorithm. *PeerJ*, 4:e2003, 2016.

APTOS. Asia pacific tele-ophthalmology society. Accessed 20 March 2021. Available: link, 2019.

Ben-David, Arie. Comparison of classification accuracy using cohen's weighted kappa. *Expert Systems with Applications*, 34(2):825–832, 2008.

Benbassat, Jochanan and Polak, Bettine CP. Reliability of screening methods for diabetic retinopathy. *Diabetic medicine*, 26(8):783–790, 2009.

Bhaskaranand, Malavika, Cuadros, Jorge, Ramachandra, Chaithanya, Bhat, Sandeep, Nittala, Muneeswar G, Sadda, Srinivas R, and Solanki, Kaushal. Eyeart+ eyepacs: automated retinal image analysis for diabetic retinopathy screening in a telemedicine system. 2015.

Brenner, Hermann and Kliebsch, Ulrike. Dependence of weighted kappa coefficients on the number of categories. *Epidemiology*, pp. 199–202, 1996.

Buda, Mateusz, Maki, Atsuto, and Mazurowski, Maciej A. A systematic study of the class imbalance problem in convolutional neural networks. *CoRR*, abs/1710.05381, 2017. URL http://arxiv.org/abs/1710.05381.

Buslaev, Alexander, Iglovikov, Vladimir I, Khvedchenya, Eugene, Parinov, Alex, Druzhinin, Mikhail, and Kalinin, Alexandr A. Albumentations: fast and flexible image augmentations. *Information*, 11(2):125, 2020.

Chetoui, M., Akhloufi, M. A., and Kardouchi, M. Diabetic retinopathy detection using machine learning and texture features. In *2018 IEEE Canadian Conference on Electrical Computer Engineering (CCECE)*, pp. 1–4, 2018. doi: 10.1109/CCECE.2018.8447809.

Chetoui, Mohamed and Akhloufi, Moulay A. Explainable diabetic retinopathy using efficientnet. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 1966–1969. IEEE, 2020.

Fastai. Lists of transforms for data augmentation in cv. Accessed 20 March 2021. Available: link, 2021.

Foracchia, Marco, Grisan, Enrico, and Ruggeri, Alfredo. Detection of optic disc in retinal images by means of a geometrical model of vessel structure. *IEEE transactions on medical imaging*, 23(10):1189–1195, 2004.

Fukushima, Kunihiko and Miyake, Sei. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pp. 267–285. Springer, 1982.

Hagos, Misgina Tsighe and Kant, Shri. Transfer learning based detection of diabetic retinopathy from small dataset. *CoRR*, abs/1905.07203, 2019. URL http://arxiv.org/abs/1905.07203.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hummel, Robert A, Kimia, B, and Zucker, Steven W. Deblurring gaussian blur. *Computer Vision, Graphics, and Image Processing*, 38(1):66–80, 1987.

Iandola, Forrest, Moskewicz, Matt, Karayev, Sergey, Girshick, Ross, Darrell, Trevor, and Keutzer, Kurt. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Mookiah, Muthu Rama Krishnan, Acharya, U Rajendra, Chua, Chua Kuang, Lim, Choo Min, Ng, EYK, and Laude, Augustinus. Computer-aided diagnosis of diabetic retinopathy: A review. *Computers in biology and medicine*, 43(12):2136–2155, 2013.

Pratt, Harry, Coenen, Frans, Broadbent, Deborah M, Harding, Simon P, and Zheng, Yalin. Convolutional neural networks for diabetic retinopathy. *Procedia computer science*, 90:200–205, 2016.

Priya, R. and Aruna, P. SVM and Neural Network based Diagnosis of Diabetic Retinopathy. *International Journal of Computer Applications*, 41(1):6–12, March 2012. doi: 10.5120/5503-7503.

Rakhlin, Alexander. Diabetic retinopathy detection through integration of deep learning classification framework. *bioRxiv*, pp. 225508, 2018.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Sisodia, Dilip Singh, Nair, Shruti, and Khobragade, Pooja. Diabetic retinal fundus images: Preprocessing and feature extraction for early detection of diabetic retinopathy. *Biomedical and Pharmacology Journal*, 10(2):615–626, 2017.

Smith, Leslie N. No more pesky learning rate guessing games. *CoRR*, abs/1506.01186, 2015. URL http://arxiv.org/abs/1506.01186.

Smith, Leslie N. and Topin, Nicholay. Super-convergence: Very fast training of residual networks using large learning rates. *CoRR*, abs/1708.07120, 2017. URL http://arxiv.org/abs/1708.07120.

Tan, Mingxing and Le, Quoc. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105–6114. PMLR, 2019.

Tymchenko, Borys, Marchenko, Philip, and Spodarets, Dmitry. Deep learning approach to diabetic retinopathy detection, 2020.

Wilkinson, CP, Ferris III, Frederick L, Klein, Ronald E, Lee, Paul P, Agardh, Carl David, Davis, Matthew, Dills, Diana, Kampik, Anselm, Pararajasegaram, R, Verdaguer, Juan T, et al. Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales. *Ophthalmology*, 110(9):1677–1682, 2003.

Yosinski, Jason, Clune, Jeff, Bengio, Yoshua, and Lipson, Hod. How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*, 2014.