APTOS Blindness Detection

Report of Kaggle competition for Project Computer Vision

S M Abdul Ahad, 23034454, MSc in Artificial Intelligence, Friedrich-Alexander-Universität Erlangen-Nuremberg.

***Abstract*—Millions of people suffer from diabetic retinopathy, the leading cause of blindness among working aged adults. Aravind Eye Hospital in India hopes to detect and prevent this disease among people living in rural areas where medical screening is difficult to conduct. Successful entries in this competition will improve the hospital's ability to identify potential patients. Further, the solutions will be spread to other Ophthalmologists through the 4th Asia Pacific Tele-Ophthalmology Society (APTOS) Symposium**

***Index Terms*—Blindness, Deeplearning , Models , etc.**

I. Introduction

THE goal of this Kaggle-hosted APTOS 2019 Blindness Detection competition was to treat diabetic retinopathy, an acute global health concern. If this left untreated, diabetic retinopathy a serious eye ailment that can affects people with diabetes and will result in blindness. In order to manage this disorder and prevent the vision loss, early detection and proper diagnosis are essential.

Participants in this competition has to develop machine learning models that could categorize retinal images into five groups, from proliferative diabetic retinopathy to no diabetic retinopathy. The dataset that have been available for this task has labels which indicates the severity of the disease with an addition to high-resolutional retinal images.

Healthcare professionals' diagnostic abilities were to be improved through the competition by utilizing AI and machine learning. The organizers aimed to promote creativity and expedite the development of automated diagnostic tools by offering a comprehensive dataset and cultivating a competitive atmosphere.

This competition is significant not just because it has the potential to enhance patient outcomes but also because it adds to the larger area of medical image analysis. The influence of the solutions produced by this competition may be expanded if developments in this field result in the creation of instruments that assist in the identification of other ailments.

This study explores the specifics of the competition, including the information, implemented approaches, evaluation metrics, and results. The findings' consequences for the field of medical imaging in general and the future of diabetic retinopathy detection.

II. Literature Review

In ”Convolutional neural networks for mild diabetic retinopathy detection: an experimental study” Rubina Sarki, Sandra Michalska, K. A. H. W. Y. Z. worked on the objective of to identify cases of mild diabetic retinopathy (DR), which are difficult to diagnose because convolutional neural networks (CNNs) frequently overlook minor signals. Experiments were carried out using 13 CNN architectures pre-trained on ImageNet utilizing transfer learning, using annotated fundus photos from public sources. To enhance performance, methods like volume growth, data augmentation, and fine-tuning were used. The best accuracy of 86% was obtained by the fine-tuned ResNet50 model on the No DR/Mild DR classification assignment. The work suggests a system for mild DR detection and highlights the significance of early DR identification. Through the utilization of deep learning and performance enhancement methodologies, the system exhibits resilience and flexibility in real-world scenarios, optimizing eye-screening processes and functioning as a diagnostic tool.

In the paper ”Deep Learning Approach to Diabetic Retinopathy Detection” Tymchenko, B., Marchenko, P., & Spodarets, D. describes an automated deep learning technique that uses single fundus imaging to detect the stage of diabetic retinopathy. By using a multistage transfer learning strategy, the method ranked 54 out of 2943 competing methods on the APTOS 2019 Blindness Detection Dataset, with a sensitivity and specificity of 0.99. The approach exhibits stability and resilience by fine-tuning on the target dataset and integrating an ensemble of three CNN architectures. Prospective enhancements encompass comprehensive ensemble SHAP computations, enhanced hyperparameter optimization, and investigation of pretrained encoders for associated assignments. The paper emphasizes how deep learning can improve the diagnosis of diabetic retinopathy and calls for more research on meta-learning strategies.

And Hagos, M. T., & Kant, S. wrote ”Transfer Learning based Detection of Diabetic Retinopathy from Small Dataset” With the goal to overcome the problem of limited annotated training data in medical picture classification, this work investigates the effectiveness of transfer learning utilizing a pre-trained Inception-V3 model for Diabetic Retinopathy (DR) detection. Through subsampling a smaller dataset from the Kaggle DR challenge, the authors outperform existing methods in binary classification. Their method, which combines a cosine loss function and an ascending learning rate with stochastic gradient descent, demonstrates how deep learning can effectively learn from tiny datasets in the medical field. The results point to the wider applicability of such methods to solve the lack of labeled data in other medical picture classification tasks. For a thorough assessment, more testing with different pre-trained convolutional networks is advised.

III. Data Exploration

The given dataset for APTOS 2019 Blindness Detection competition on Kaggle includes images of retina and associated lables to develop machine learning models to classify the stages of diabetic retinopathy in patients, which is one of the leading cause of blindness globally.

A. Dataset Description

1. Images:

The dataset includes high-resolution images of eyes captured using fundus photography, which is a specialized form of medical imaging that provides a view of the retina. These images are color represents the interior surface of the eye, including the retina, optic disc, macula, and posterior pole, also includes the arteries, veins, and fundus.

2. Labels:

Images in the dataset is labeled with a corresponding severity stages of diabetic retinopathy. The labels are integer values representing different stages of the disease:
0: No diabetic retinopathy
1: Mild diabetic retinopathy
2: Moderate diabetic retinopathy
3: Severe diabetic retinopathy
4: Proliferative diabetic retinopathy

3. Files:

"train.csv": This file contains the training data with two columns: id_code (the unique identifier for each image) and diagnosis (the severity grade of diabetic retinopathy). "test.csv": This file contains the testing data with one column: id_code, which can be used to match with the images for making predictions. "train_images": A folder containing the training images with filenames matching the id_code in "train.csv". "test_images": A folder containing the testing images with filenames matching the id_code in "test.csv".

B. Data Composition

1. Training Data Shape: (3662, 4) Columns: id_code: Unique identifier for each image. diagnosis: Severity grade of diabetic retinopathy (0 to 4). file_path: Path to the image file. file_name: Name of the image file.

2. Testing Data Shape: (1928, 3) Columns: id_code: Unique identifier for each image. file_path: Path to the image file. file_name: Name of the image file.

C. Class Distribution in Training Data

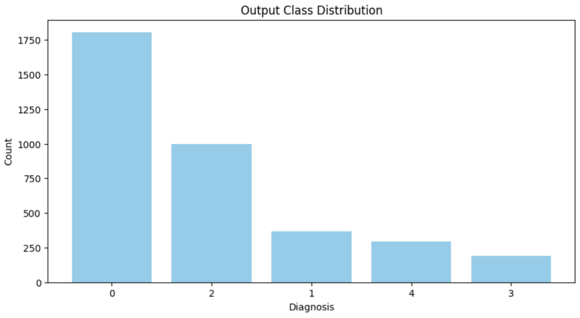

Fig. 1. Class distribution of training data.

The training dataset consists of 3662 images, each labeled with a specific DR severity grade. The distribution of the classes is as follows:

No DR (Class 0): 1805 images Mild DR (Class 1): 370 images Moderate DR (Class 2): 999 images Severe DR (Class 3): 193 images Proliferative DR (Class 4): 295 images This distribution highlights a significant class imbalance, with the majority of images falling into the 'No DR' category.

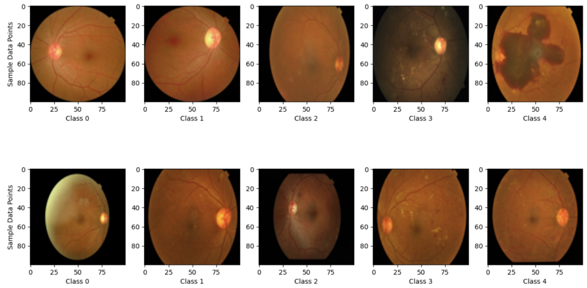

Fig. 2. Retina Images of different class in the training dataset.

IV. Pre-Porcessing

The images above demonstrate that they were not all taken simultaneously, as technicians visited the different locations to take pictures. As a result, the photos are in non-uniforemd form is evident. Pre-processing the photos is crucial to ensure uniformity and improve the effectiveness of machine learning models, as the APTOS 2019 Blindness Detection dataset has variable image capture. The two primary phases in this pre-processing are resizing photographs without altering their aspect ratio and cutting off unwanted borders. To further guarantee that the preprocessing is effective and scalable for big datasets, multiprocessing has been used.

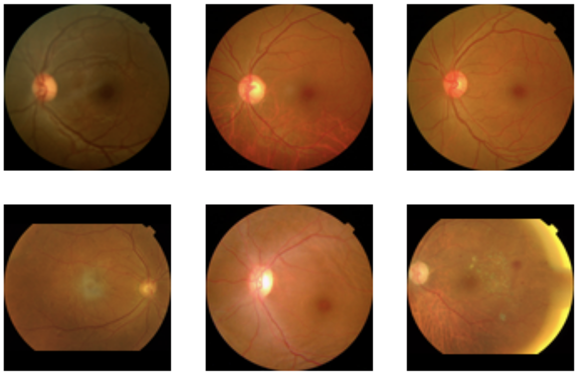

Fig. 3. Retina Images after pre-processing.

1. Cropping Insignificant Borders: Since the photos in the dataset were gathered from rural regions, they frequently include black borders as a result of uneven capture techniques.

2. Resizing to Consistent Size While Preserving Aspect Ratio: In order to avoid distortion, photos are cropped and then resized to a constant size while keeping their aspect ratio. To match the required dimensions, the image is resized and padded using the "resize_maintain_aspect" function.

3. Saving the processed Image: The previously mentioned routines process each image and save it to the designated directory. One image's cropping, scaling, and saving are handled by the function "save_single".

4. Batch Processing with Multiprocessing: Multiprocessing is used to process a large dataset in an effective manner. "Fast_image_resize" is a function that speeds up preprocessing greatly by dividing the task among several CPU cores.

Pre-processing ensures that every image has the same dimensions and that, in order to prevent distortion, their aspect ratios are maintained. Every image has the retinal area centered within it, removing any extraneous black borders while preserving the important details(Fig. 3). As a result, the dataset is more reliable and consistent, which makes it easier to train algorithms that detect diabetic retinopathy.

V. Model Implmentation Methodologies

This projects main goal was to use deep learning techniques to develop a consistent model that could identify diabetic retinopathy (DR) from retinal pictures. Initally, I opted for ResNet-18 and ResNet-50, two distinct pre-trained models. These models belong to the ResNet family, which is renowned for its performance in application of image classification. To optimize the pre-trained models for this particular task, a new layer that produces five classes(stages) matching to the DR severity levels was substituted for the final fully connected layer.

1. Data Pre-processing for Model Implementaiton:
For the training I used previousely processed (IV. PRE-PORCESSING) Images and the images were subsequently scaled to 224x224 pixels and normalized to meet the input specifications of pre-trained deep learning models before being fed into the model. The pre-trained models were first trained on the ImageNet dataset, whose mean and standard deviation values were used to normalize the images.

2. Preparing the Dataset for Model Implementaiton:
Custom Dataset Class: To manage the loading and alteration of pictures and labels, a custom PyTorch Dataset class called BlindnessDataset was made. This class imports the photos, applies the designated transformations, then reads the image URLs and names from a CSV file.

3. Data Splitting:
To guarantee that every class was fairly represented, the dataset was divided into training and validation sets using stratified sampling. By keeping the balance across various DR severity levels, our method made sure that the model was trained on a representative and varied subset of the data.

4. Training Process:

A. *ResNet18 & ResNet50:*

For ResNet18 and ResNet50, the Cross-Entropy loss function, which works well for multi-class classification tasks, was used to train the models. The model weights were updated using the Adam optimizer, which has a learning rate of 0.001. After 13 epochs of training, the models' performance on the training set was assessed to make sure they were picking up new information efficiently and avoiding overfitting.

The performance of the ResNet18 and ResNet50 model on the diabetic retinopathy classification task was further assessed when a more sophisticated training strategy was put into place, which included the addition of weight decay(e.g 1e-3), a learning rate scheduler, and sticking to the initial number of epochs (13). The subsequent adjustments were implemented:

Weight Decay(WD): By penalizing excessive weights in the model, a weight decay parameter was added to the optimizer to stop over-fitting. This regularization method helps the model to more effectively generalize to unknown input.

Learning Rate Scheduler: To progressively lower the learning rate during training, a learning rate scheduler was implemented. Every five epochs, the learning rate was lowered by a factor of 0.1. By adjusting the model's parameters, this technique aids in its convergence to a better answer.

B. *EfficientNet's:*

In order to explore the possibilities of more sophisticated architectures, I switched to the EfficientNet models, which is a renowned cutting-edge convolutional neural network(CNN) for its effectiveness in maintaining a balance between computing cost and accuracy. To take the advantage of the information gathered from extensive image datasets, the model is implemented with pre-trained weights. I improved the model by changing the last classification layer to produce predictions for all five severity classes required for the blindness detection test.

The training procedure was structured similarly to earlier studies, but with further methods to reduce overfitting:

Loss Function: To make sure the model's predictions were nearly in line with the right labels, I implemented CrossEntropyLoss, which is common for multi-class classification tasks.

Optimizer and Learning Rate Scheduling: To avoid overfitting, the Adam optimizer was used with a weight decay of 1e-3 and an initial learning rate of 0.001. I used a StepLR scheduler to adjust the learning rate over time. This allows the model to converge more smoothly by lowering the learning rate by a factor of 0.1 every five epochs.

Early Stopping: I added an early stopping mechanism to better prevent overfitting and prevent pointless training epochs. Training would end early if the validation score, which is determined by the Quadratic Weighted Kappa, or QWK, did not increase for three consecutive epochs.

The model was continuously assessed on a validation set to monitor its performance during training. To track how well the model was learning to classify the images in a fashion that matched human judgment, the QWK score was computed at each epoch. In order to maintain the model in its ideal state, the top-performing model weights were stored for later use.

I tried to further improve the EfficientNet-B0 model's performance by using more advanced data augmentation techniques during training after the model produced good results. The intention was to imitate real-world variability in medical imaging by subjecting the model to a wider range of picture alterations, hence improving its generalization.

Extension to Data Pre-Processing here I included Data Augmentatin and Balanced Sampling for EfficientNet.

1.1 Data Augmentation:
I made use of a number of transformations, such as RandomRotation (up to 20 degrees), to mimic various camera viewpoints. To get the model to focus on different areas of the image, RandomResizedCrop was used to crop and resize the images at random, with a scale variation between 80% and 100%. Flipping is a new source of variety introduced by RandomHorizontalFlip. In order to increase the model's resilience to various lighting scenarios, ColorJitter introduced random variations in brightness, contrast, saturation, and hue. The pipeline that was created from these augmentations made sure that every image that was used to train the model was a distinct variation. In particular, for real-world settings when image circumstances fluctuate greatly, this step was essential to boosting the model's generalization capabilities.

1.2 Balanced Sampling:
In order to rectify the imbalance of classes in the dataset, I devised a training technique that involved equal sampling of each class. To ensure that the model received an equal amount of data from each class, the maximum class count was determined and then applied to underrepresented classes using augmentations.

But, looking at the results this changes does not impacted the result significantly. Thus, for further exploration I did not considered this data augmentation and balanced sampling techniques.

C. Ensamble:

As, from the above implemented methods we saw that ResNet18 and EfficientNet-B0 outputs better validation score individually. Thus, I tried EfficientNet-B0, B1, B2, B3, B4, B5, B6, B7. While training all the EfficientNet I noticed that EfficientNet-B5 gives a very good Keppa Score. Thus, I ensambled EfficientNet-B5 & our previous best performed ResNet18 (with weight decay: 1e-3) and I got a very good validation keppa score. But for the test dataset this combination also does not performed well.

Thus, I was exploring for more better models and I found out that 'seresnext50_32x4d' and 'seresnext101_32x4d' works better for this dataset in ensambling with other models. Thus, I trained this two model with our previously implemented pre-processing techniques. Also, "inception_v4", and "inception_resnet_v2" also works better for medical image classification, thus, I trained this two models too. Individually, though above mentioned methods are performing very well with a very good validation score, but when I ensamble them for the test evaluation along with "ResNet18" & "EfficientNet-B5" and submit into the competition, the test score becomes too much low compared to submitting just one single model for test data evaluation. But, as we know ensamble usually outperform the individual performance, thus, I made multiple combination out of all the models for ensambling and after trying multiple combination of ensamble submission, upon submitting with 'resnet18', "seresnext101_32x4d",and "seresnext50_32x4d", I got the highest test. This time I also used weighted softmax according to validation score for scoring.

VI. Results & Evaluation

The evaluation was centered on tracking the Quadratic Weighted Kappa (QWK) score because this measure is very important for determining the ordinal nature of the severity levels of diabetic retinopathy.

A. ResNet18 and ResNet50 Base Model:

Training Performance of the ResNet18 Model: During training, the ResNet18 model showed consistent improvement, with the loss falling from 0.75 in the first epoch to 0.11 in the last epoch. This suggests that the model was successfully picking up the characteristics required to differentiate between various degrees of DR severity.

Validation Performance of the ResNet18 Model: By the thirteenth epoch, the QWK score on the validation set had increased from 0.75 in the first epoch to 0.86. With predictions that almost match the labels of the ground truth, the model appears to be operating well, as indicated by its score of 0.86.

Training Performance of the ResNet50 Model: After 13 epochs, the ResNet50 model also demonstrated a steady decline in training loss, beginning at 0.79 and ending at 0.24. But there were some variations in the model's learning curve, especially between epochs 6 and 8, which might indicate overfitting or difficulties fine-tuning the deeper architecture.

Validation Performance of the ResNet50 Model: The QWK score peaked at 0.83 after varying over epochs from a starting point of 0.75. Compared to ResNet18, the validation performance of the ResNet50 model was marginally more unstable, despite its overall good performance. ResNet50's increased complexity may be the cause of this, necessitating further regularization or more cautious tweaking.

While ResNet18 produced significantly more stable findings, both ResNet18 and ResNet50 models demonstrated outstanding performance in predicting the severity of diabetic retinopathy from retinal pictures. Effective and accurate predictions were made possible by the combination of comprehensive evaluation, substantial data pre-processing, and the use of pre-trained models.

B. ResNet18, ResNet50 with enhanced training strategy

Training & Validation of the ResNet18 Model with WD=1e-3: A learning rate scheduler that lowered the learning rate by a factor of 0.1 every five epochs and set weight decay to 1e-3 were used to test the ResNet18 model over a period of 13 epochs. The outcomes showed that the improvements were quite successful:

First Five Epochs (1-5): The loss dramatically reduced, and the QWK score increased from 0.68 to 0.79.

Mid Epochs (6–10): The model successfully adjusted its parameters by lowering the learning rate, which resulted in a significant decrease in loss and an improvement in the QWK score to 0.86.

Final Epochs (11–13): In epoch 11, the model showed strong and stable performance by maintaining a low loss and reaching its best QWK score of 0.88.

0.8875 is the best validation QWK for ResNet18.

Training & validation of the ResNet50 Model with WD=1e-3: In contrast to ResNet18, the ResNet50 model had a different trajectory after being trained for 13 epochs with the same parameters (weight decay of 1e-3 and a learning rate scheduler). This is how the instruction went:

First Five Epoches (1-5): In the first epoch, the loss was greater at 0.87 and had a QWK score of 0.70. Because of the intricacy of the model and its initial high learning rate, the performance varied slightly, with the QWK decreasing to 0.65 by epoch 5.

Mid-Epochs (6–10): Following the decrease in learning rate, the ResNet50 started to stabilize. The loss kept going down, and by epoch 10, the QWK score had increased to 0.83.

Final Epochs (11–13): ResNet50 kept getting better in the concluding epochs. The best QWK score attained 0.85 in epoch 13, suggesting that the model had at last perfected its capacity to discriminate between the severity levels. 0.8454 is the best validation QWK for ResNet50.

With the improved training procedures, ResNet18 and ResNet50 both demonstrated notable gains, although their learning curves differed.

ResNet18's performance improved steadily and consistently as it was trained. With rapid data adaptation, it reached a higher peak QWK score of 0.88. This indicates that ResNet18 was easier to train and had better task generalization due to its more simplistic architecture. On the other hand, the training procedure for ResNet50 was more complex. As is typical with deeper networks with more parameters, it took longer to stabilize at first. But ResNet50 started to catch up as training went on, especially once the learning rate was lowered, and it eventually achieved a strong QWK score of 0.85. ResNet50's deeper design showed promise for improved performance, even if it needed additional fine-tuning—especially in instances with larger or more complicated datasets.

C. EfficientNet:

As early as the first epoch, EfficientNet-B0 showed promising performance, achieving a QWK score of 0.81. The model improved further throughout training, readily adjusting to the subtleties of the dataset. The model peaked at the fourth epoch, showing a high degree of agreement with the ground truth labels with a QWK score of 0.8993.

With the use of sophisticated data augmentations during training, the EfficientNet-B0 model demonstrated a notable improvement over the course of 13 epochs. It had a great start with a QWK score of 0.74, which was encouraging. Over the course of training, the model's accuracy increased gradually, peaking at 0.90 by the 10th epoch. This suggests that the supplemented data was a useful source of learning and generalization for the model. In order to keep the model operating at peak efficiency and avoid overfitting, early stopping was initiated after the 13th epoch. The model's resilience and applicability for actual medical image processing in real-world scenarios are highlighted by its final QWK score of 0.90. But, If I compare with the initial EfficientNet-B0 performance this data augmentation techniques does not improved the validation accuracy significantly.

D. Ensamble:

After trying multiple combination of models with or without pre-processing for ensamble submission, upon submitting with 'ResNet18', "seresnext101_32x4d",and "seresnext50_32x4d", I got the highest Test Score of 0.848385.

No	Model	Validation Score	Test Score
1	ResNet18	0.8875	0.830
2	ResNet50	0.8454	-
3	EfficientNet-B0	0.8993 , 0.9038 (Augmented Data)	-
4	EfficientNet-B5	0.9044	-
5	InceptionResNet_V2	0.7537	-
6	InceptionV4	0.8376	
7	sersnext-101_32	0.9043	-
8	sersnext-50_32	0.6524	-
8	Ensamble	-	0.848385

TABLE I

RESULTS OF DIFFERENT MODELS.

VII. DISCUSSION AND SUMMARY

The highlight of the competition was to find out potential of machine learning to detect diabetic retinopathy to avoid blindness as early as possible. Despite chalnges like image variablity and class imbalances I used transfer learning techniques to develop an efficient models and finally ensambled models ('ResNet18', "seresnext101_32x4d",and "seresnext50_32x4d") which gives keppa score (QWK): 0.848385, detecting the blindness. However, the issue with model interpretability and real-world generalization remain. But, ultimately it shows that Artificial Intelligence/Machine Learning/Deep Learning is very useful tool for medical image processing.

REFERANCES