Electronic Letters on Computer Vision and Image Analysis 0(0):1-10, 2021

Detection and Grading of Diabetic Retinopathy from Fundus Images using Transfer Learning

Shravan Chandra[*], Anand Vinekar[+] and Gowri Srinivasa[*]

[*] *PES Center for Pattern Recognition, PES University, Bengaluru, India*
[+] *Department of the Pediatric Retina Service at Narayana Nethralaya Eye Institute, Bengaluru, India*

Received TBD; revised TBD; accepted TBD

Abstract

Diabetic retinopathy is a progressive disease caused by uncontrolled diabetes that, if not treated, can lead to complete loss of vision. In this work, we focus on a system that is capable of detecting the disease and grading the severity of the disease with high accuracy for a publicly available data set of fundus images collected in India. In particular, we present a transfer learning paradigm that involves pre-training with a large, publicly available data set that is annotated to be able to achieve a high accuracy and weighted kappa score with minimum effort for retraining on the limited data available from the Indian subcontinent. Our experiments have demonstrated that the best of the deep learning models we have explored achieves a promising accuracy of 92.95% and a high quadratic weighted kappa score of (κ^2)=0.96 with less than ten epochs of retraining.

Key Words: Diabetic Retinopathy, Retina, transfer learning, ResNet, Xception.

1 Introduction

The incidence of diabetes in India is growing at a rapid rate, with over seventy seven million people affected by the disease. [11] Uncontrolled and chronic high blood sugar levels is the chief cause for the onset and progress of Diabetic Retinopathy (DR), a complication that affects the eye. There are 4 grades of DR based on the severity of the disease: Mild DR, Moderate DR, Severe DR (all three are nonproliferative) and Proliferative DR. Even the most severe form of the disease has a treatment protocol available, if detected while it can still be treated. [12] The disease is presently detected through an independent visit to the ophthalmologist [5]. Since DR is a progressive disease, early detection and staging is imperative in being able to protect a patient's vision. This necessitates routine screening or monitoring for the at-risk population. Routine screening in the traditional manner, however, may impose a burden on ophthalmologists and retinal specialists, since only a fraction of the patients screened may need their intervention. This makes a compelling case for an automated pre-screening system for DR that is affordable and available to those at risk for DR. Such a pre-screening system can help in the early detection of DR, diminishing the chances of total blindness due to DR. An additional benefit is that the time and expertise of ophthalmologists and retinal specialists can be focused on those who do need intervention.

Correspondence to: <E-mail>

Recommended for acceptance by <name>
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

There have been several efforts made to automate the detection and staging of DR. In particular *Hatanaka and Nakagawa* proposed an automatic classification of the different stages of DR based on six features extracted from the retinal images using the neural network. [1] *Yun and Acharya* described an improved method for detecting hemorrhages in fundus images. [3] The overall detection scheme consisted of six stages - image digitization, image normalization, extraction of optic nerve head, detection of hemorrhages candidates, elimination of false positives (FP) in blood vessels at the stage of vessel segmentation, and elimination of FPs by feature analysis at the classification stage. Although an effective detection of hemorrhage candidates was not dealt with in this work, it demonstrated the mapping between diagnostically relevant features and numerical descriptors and paved way for an automated classification approach.

In the foregoing papers, considerable effort was invested in engineering numerical descriptors of the pathology. The performance of the classification relies heavily on the descriptive and discriminative efficacy of these hand-crafted features. However, with the advent of deep learning architectures, CNN-based networks among others, have exhibited capabilities that far exceed traditional feature engineering in numerous image recognition tasks. [2] This can be ascribed to the ability of such networks to automate the process of feature extraction. The features extracted by CNN-based architectures may not be as amenable for interpretation as hand-crafted features; all the same, these are statistical patterns that optimize an objective, which could be minimizing loss or FP's or maximizing accuracy, etc., more effectively than numerical descriptors have managed to. Data is an important aspect of deep learning architectures, since the basis of the implicit extraction of features and subsequent classification, is the input data.

2 Data

We use two datasets in this study: the EyePACS dataset and the Asia Pacific Tele-Ophthalmology Society (APTOS) 2019 dataset. The EyePACS dataset includes $35,126$ images of various dimensions, with sufficient variability for each class, and was obtained from Kaggle: EyePACS. The APTOS contains 5590 annotated images of dimension 2136x3216. This dataset is representative of the Indian population, captured and graded by retinal specialists at the Aravind Eye Hospital, made available on Kaggle: APTOS. Since both datasets follow similar imaging protocols and convention for grading the severity of DR, we can combine the data sets. Table 1 presents a summary of the numerical categories, the label they correspond to and the number of images available in each class. We note that the 'No DR' class (or the class of images that is considered 'healthy' for the purpose of this study) is overwhelmingly over-represented compared to the other categories. Among the four categories that represent different grades of DR, we note the largest number of images for Moderate DR in both data sets.

Class	Label	EyePACS	APTOS
0	No DR	25810	1805
1	Mild DR	2443	370
2	Moderate DR	5292	999
3	Severe DR	873	193
4	Proliferate DR	708	295

Table 1: Dataset overview: Number of images in each class

3 Methodology

In this study, we attempt to use transfer learning to achieve a high classification accuracy and an automated method for grading of DR, with minimum effort in retraining the network. We begin with CNN-based architectures that are initialized with the ImageNet data. This helps us leverage the implicit understanding of generic image patterns of the deep learning architecture. Next, we train these using the relatively larger number of images available in the EyePACS data (split using the 80:20 rule, with 20% of the images used to validate the model and ensure there has been no overfitting). This is our training phase. To networks trained in this manner, we then input the relatively fewer APTOS images (also split using the 80:20 rule, with 80% of the images used to retrain the network and 20% being hold-out data that is not presented to the network at this stage). This is our 'retraining' phase. The best of models at the end of this process constitutes the feedforward network that can be used to classify any input image or 'test data'. The output of this automated screening system is a class label that corresponds to one of the category labels listed in Table 1. A schematic diagram of the pipeline is shown in Fig. 1. Each block of the pipeline is explained in detail in the subsections that follow.

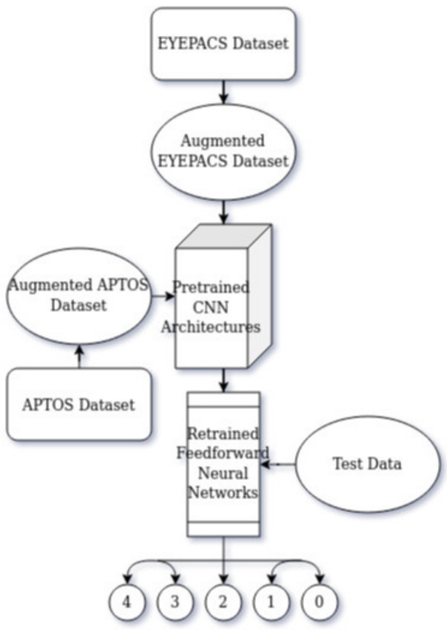

EYEPACS Dataset
Augmented EYEPACS Dataset
Augmented APTOS Dataset
Pretrained CNN Architectures
APTOS Dataset
Retrained Feedforward Neural Networks
Test Data
4
3
2
1
0

Figure 1: A schematic diagram of the solution approach

3.1 Data Preprocessing and Augmentation

As noted in Section 2, both the datasets we use in this study present the challenge of imbalance between the 'No DR' and 'DR' classes. To overcome this imbalance, the data was augmented synthetically.

The EyePACS dataset comprises a substantial number of fundus images with diverse resolutions. We selected 300 images from each class and resized them to 256 $\times$ 256 pixels with the visible region of the fundus image centered in every image. To eliminate the boundary effect, the field of view (FOV) was cropped to 80% diameter of the original FOV. We computed local averages of the intensity for each pixel in a window of 4×4 around the pixel and subtracted these local averages to overcome the variation in lighting across the images in similar .

Due to the inadequate number of training images available per class in relation to the complexity of CNN architectures, we employed real-time data augmentation, where the original data is artificially transformed to generate new data for training. Data augmentation also bypasses overfitting by utilizing various, randomly transformed images for training in every epoch, as compared to the original images. We applied the following random transformations to produce synthetic data in real-time:

(i) Rotation - This can help in replicating the various positions of retinopathy lesions which can appear in any orientation.

(ii) Flipping - To ensure positional invariance of retinal features, such as the Optic Disc, in determining the severity.

(iii) Zooming - Aids in capturing features that may have been missed due to the brevity of feature.

(iv) Shearing - Lesions develop in various shapes and sizes, and sheering can help to replicate some of these.

A few sample images have been presented in Figure 2.

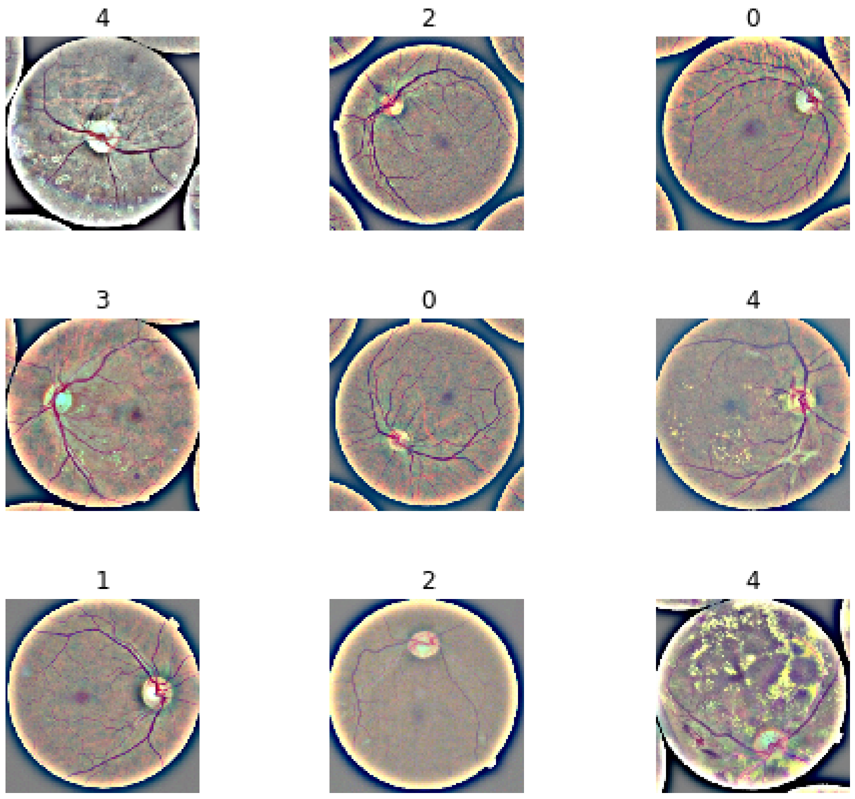


Figure 2: Preprocessed and augmented images of the retina (the number atop each image indicates the category (i.e., the DR severity) to which the image belongs).

The next step in the pipeline comprises CNN-based deep learning architectures.

3.2 CNN

Convolutional Neural Networks (CNN) comprise a series of correlation filters whose dimensions can be set empirically. [13] These facilitate detecting highly representative, data-driven, stratified image characteristics (such as lines, curves, textures, etc., with each progressive layer) from adequately comprehensive data. Nevertheless, gathering a dataset that is as extensive and comprehensively annotated as the ImageNet dataset for the detection and grading of DR severity is challenging. Therefore, the first step was to create a vanilla CNN model to use as a baseline comparison for the rest of our models. For this model, both the datasets were used together for training.

3.2.1 Vanilla CNN

CCN served as our baseline model. This comprised of 7 convolution blocks with MaxPooling, and the ReLU activation function. The embedding was then passed through a small feedforward neural network layer, finally classifying the image. Also, it has to be noted that all the architectures had early stopping and reduced learning rate callback employed to prevent overfitting and to facilitate the fine-tuning of models.

3.3 Pretrained CNN-based Architectures

It was our idea of leveraging CNN-based architectures pretrained with imagenet weights for improved performance was inspired by the promising results for a similar problem setting, among others. [4] [15] Similar to these references, we have explored multiple models, including VGG16, Xception and ResNet50, in this framework. The embeddings from the pretrained models were flattened out to a single vector using the GlobalMaxPooling layer. GlobalMaxPooling was chosen, since it can capture features that have high value, rather than just averaging them all out. During training, the pretrained weights were first frozen, and only the neurons after the embeddings were trained. Next, the pretrained weights were unfrozen for a more careful fine-tuning of the model. Progressive resizing is a technique developed by Jeremy Howard as part of the fast.ai class. The approach is that we train with smaller images at the beginning and retrain with larger images which will have added information to learn from, which could be very helpful for dealing with small datasets. Introducing this technique boosted the accuracy by 4% and the kappa score by 2%. The batch size was set to 32, and the optimizer selected was *Adam*, with an initial learning rate of 0.001. While the overall accuracy in *Khalifa et al.* is impressive, we have expended considerable effort in ensuring there is no overfitting in our models and have made all the code used for this study available in the public domain (see the Section on Reproducible Research) to facilitate reproducing the results and improving upon these. [15]

3.3.1 VGG16

VGG16 is a convolutional neural network architecture proposed by *Simonyan and Zisserman* from the University of Oxford. [10] This architecture presents enormous width with around 138 million trainable parameters but a relatively limited embedding size of 512.

3.3.2 Xception

Xception stands for ”Extreme version of Inception,” was developed by *Chollet* in Google. [9] With a modified depthwise separable convolution, it is much better than Inception-v3, which was also developed by Google. In the version of the architecture we used, there were no skip connections.

3.3.3 Residual Networks

Residual Networks, commonly known as, ResNets was developed by *He and Zhang*. [8] The essence of ResNet is the introduction of “identity shortcut connection” that skips one or more layers. This circumvents degradation of performance with the stacking of more layers. By simply stacking identity mappings, the resulting architecture would perform equivalent to one which includes layers that are not doing anything. This symbolizes that the deeper model should not present a training error higher than its shallower equivalents. We used a version of ResNet called ResNet50 (with 50 indicating the number of layers of the residual network) and provides an embedding size of 2048. Among the CNN-based architectures, it is ResNet50 that yielded the highest accuracy and highest kappa score on hold-out data and the least standard deviation across multiple folds of cross validation (details of which are elucidated in the subsequent section). Consequently, this architecture is our method of choice.

3.4 XGBoost

XGBoost is a decision-tree-based ensemble Machine Learning algorithm proposed by *Chen and Guestrin* that employs a gradient the boosting framework. [7] In classification tasks comprising unstructured data like images and text, feedforward neural networks manage to outperform all other algorithms or frameworks. Nevertheless, when it comes to small-to-medium sized structured data, decision tree-based algorithms are regarded as the best-in-class. We wanted to test the performance of XGBoost, considered one of the best available ML algorithms, against the transfer-learning based pipeline proposed in this paper. Since the focus of our work is on achieving the highest accuracy and kappa scores for APTOS images and the XGBoost algorithm works well with limited data, we limited the training to images from the APTOS training set. We resized the images to $128 \times 128 \times 3$, and flattened this into a single vector of size $49,152 \times 1$. As we will see in the subsequent section, the XGBoost algorithm outperformed the vanilla CNN architecture.

4 Experiments AND Results

The performance of XGBoost and the various CNN-based architectures is shown in Table 2. We rank the performance of a model by the quadratic weighted kappa score (κ^2). Although DR severity are indicated by distinct classes, there is an underlying order, from 0 (No retinopathy) to 4 (Proliferative retinopathy). In this case, accuracy is a harsh yardstick as it seeks to match the labels verbatim. For instance, an image that is predicted to show features of DR grade 3 but is in fact annotated DR grade 2 as per the ground truth, is considered a misclassification (given a weight of 0) by accuracy. In practice, however, the algorithm has performed reasonably in recognizing the presence of the disease and calling out the higher grade of severity ('erring on the side of caution') and should be ranked over a method that predicts a label of 0 or 4. The quadratic weighted kappa score, that takes into account such nuances, is thus a more reliable performance indicator. Misclassifications, as explained in the foregoing example, can be weighted differently when calculating κ^2. Nevertheless, we strive to achieve as high an accuracy as possible, since misclassification is not desirable. Misclassifying an image with a higher severity of DR as one with a lower severity, for instance, can lead to negligence, which can cause serious damage in later stages.

A summary of the validation loss, accuracy and kappa score is presented in Table 2.

Model	Loss	Accuracy	Kappa Score
`XGBoost`	0.820	55.60%	0.4449
`CNN`	0.772	34.34%	0.6447
`VGG16`	0.671	57.46%	0.9209
`Xception`	0.446	59.34%	0.9584
`ResNet50`	**0.210**	**92.95%**	**0.9609**

Table 2: Performance of Various Architectures

We observe that Vanilla CNN, XGBoost and VGG16 performed rather poorly. Surprisingly, XGBoost outperformed Vanilla CNN. For Vanilla CNN, we conjecture the poor outcome is on account of the limited representation. While XGBoost is a limited framework in comparison to CNN-based architectures, in the case of VGG16, it is possible that the width of the architecture and limited embedding size of 512, unlike other architectures which have 2048, may have been insufficient to represent nuances of features for better prediction. Finally, due to the sheer magnitude of the architecture, VGG16 took the longest to train, around 30% longer than the other architectures.

The Xception architecture has a promising kappa score, though the accuracy rather poor (see Table 2). This can be explained as a consequence of misclassification of the gradations of severity of proximate categories. To verify the results, we modified the model into a five output softmax activated network. By this modification, the accuracy shot up to 80%, but the kappa score fell to 62%. This may be on account of the system being pushed to achieve nearly perfect recall for the most represented class, yielding low weights on the kappa score for other categories. It is possible that the addition of residual /skip connections can improve the results.

ResNet50 yielded the highest accuracy and kappa score on our hold-out data. These scores were verified by computing the standard deviation across five folds of cross-validation, which resulted in a standard deviation of

$2.5e-3$ for accuracy and $1.8e-3$ for kappa score. We believe the more comprehensive embedding of length 2048 along with the residual connections helped with achieving this high score.

Since ResNet50 yielded the highest accuracy and kappa scores, we analyze the results of this architecture further.

4.1 ResNet50: Training and Performance Analysis

Figure 3 shows a plot of the loss during the training for each batch of the training process through eight epochs in blue and a plot of the loss computed at the end of each epoch for the validation set in orange. We notice that

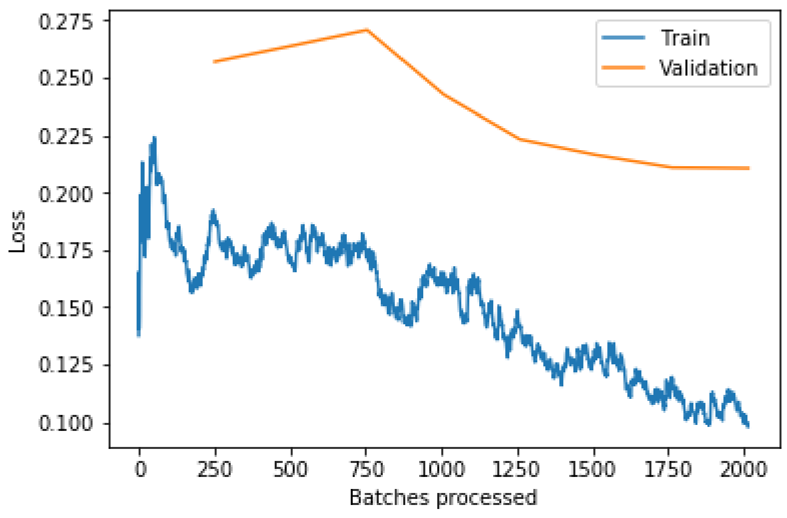


Figure 3: Loss Plot during Progressive Resizing Training of ResNet50.

the training loss, though not monotonic, progressively decreases. The validation loss is higher than the training loss, but, as can be expected, this too decreases with epochs. This is a confirmation that there is no overfitting in the training phase.

Table 4 shows the last 8 epochs - that correspond to the training of APTOS dataset - with progressive resizing. The reduction of validation loss and increase in accuracy and quadratic kappa on the validation set is a confirmation of there being no overfitting in the training process. We stop with reporting the results after eight epochs as the change in accuracy and kappa score for further epochs is not deemed statistically significant to be reported.

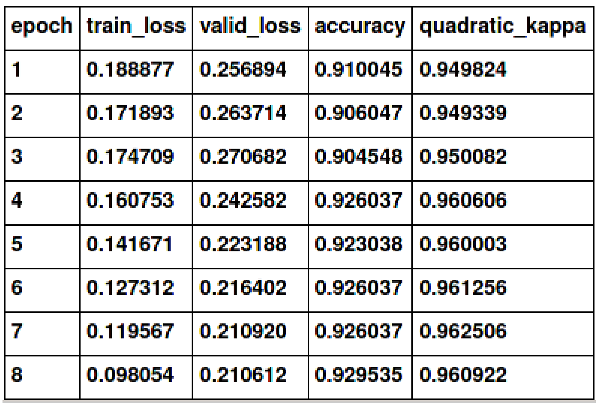
Figure 5 represents the confusion matrix for the test data.

There was no overlap of the test images with the training set and no image from the test set was presented to the model at any stage prior to running the tests.

Class Label	Accuracy (%)
No DR	98.41
Mild DR	95.81
Moderate DR	86.55
Severe DR	91.43
Profliferate DR	92.02

Table 3: Performance of Various Architectures

We note that the accuracy of the classes is highest for ‘No DR’ as it happens to be the most represented class. It is possible the model has picked up nuanced features of ‘DR’ through the various stages that has facilitated



epoch	train_loss	valid_loss	accuracy	quadratic_kappa
1	0.188877	0.256894	0.910045	0.949824
2	0.171893	0.263714	0.906047	0.949339
3	0.174709	0.270682	0.904548	0.950082
4	0.160753	0.242582	0.926037	0.960606
5	0.141671	0.223188	0.923038	0.960003
6	0.127312	0.216402	0.926037	0.961256
7	0.119567	0.210920	0.926037	0.962506
8	0.098054	0.210612	0.929535	0.960922

Figure 4: Accuracy & Kappa Score Plot during Progressive Resizing Training of ResNet50 (in blue). The validation loss at the end of each epoch has also been presented (in orange).

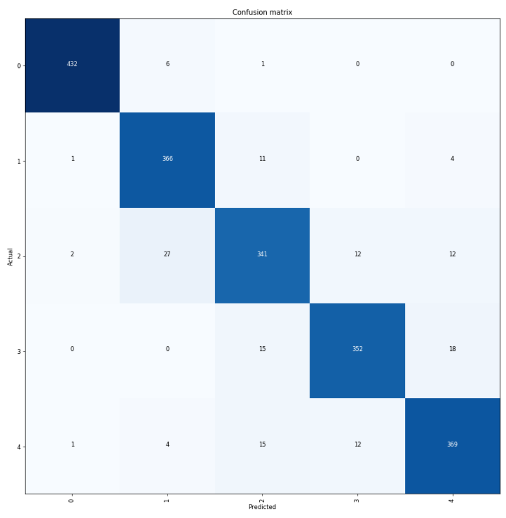


Figure 5: Confusion Matrix of the APTOS Validation Datset trained using ResNet.

the classification of ‘No DR’. It is interesting to note that the second most represented category viz., Moderate DR (severity grade 2) happens to have the lowest classification accuracy. We surmise this is on account of the similarity in features with severity grades 1 and 3. Moving forward, we would endeavor to reduce the misclassification to a lower grade.

While most typical cases are classified correctly with no special effort, Fig. 7 presents samples of images that were classified correctly by ResNet50 but failed to be classified correctly by Xception, the model closest to ResNet50 in the overall performance. The labels for the images presented in Fig. 7 as predicted by Xception (moving from left to right) are: 1, 4 and 3 respectively, when the actual labels happened to be 0, 2 and 4, respectively.

No model is perfect, and, for the sake of completeness, we present the images with the highest loss from ResNet50 in Figure 6,

The label predicted by ResNet50 is presented here with the ground truth in parenthesis (moving from left to right): Row 1: 4(2), 3(2), 0(2), Row 2: 3(2), 0(2), 2(1) and Row3: 2(0), 1(2), 3(2). As we note, there are more errors classifying an image to a higher grade than a lower one and, in the current scenario of automated screening, where it is better to err on the side of caution, such an error is acceptable. However, it is the errors of classifying an image to a lower grade of severity that needs to be revisited.

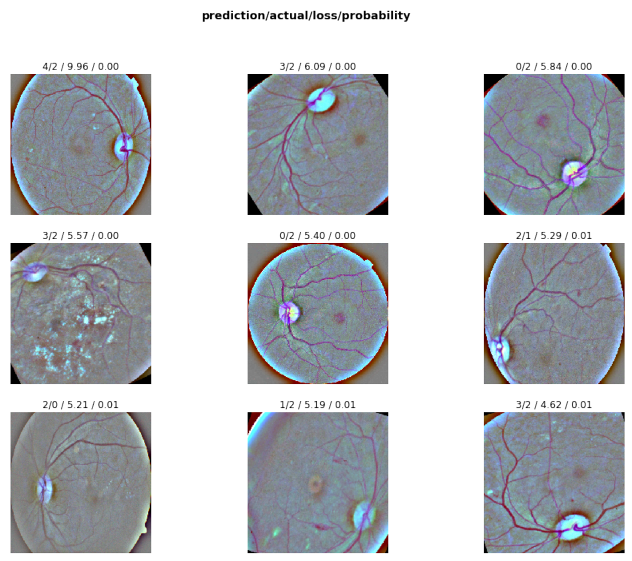


Figure 6: Top Losses by the ResNet50 Model during Prediction.

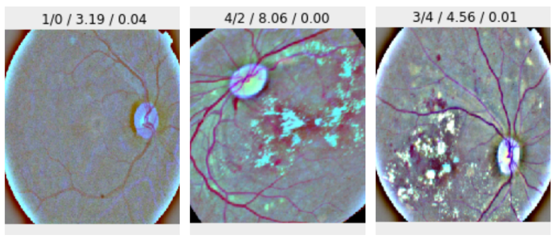


Figure 7: A few samples where ResNet was right and Xception was wrong.

5 Reproducible Research

In the spirit of reproducible research, we have used publicly available data (EyePACS and APTOS) and made all the code used in this study available here.

6 Conclusion

We have presented a framework for the automated detection and grading of the severity of diabetic retinopathy. The proposed method leverages the power of CNN-based architectures pretrained with ImageNet and trained using EyePACS data, to achieve high accuracy and quadratic kappa scores on hold out data, with minimum effort in retraining the images from an Indian database. We have ensured there is no overfitting in the training phase through tracking the validation loss at the end of each epoch and computing the standard deviation across five-fold cross-validation. We have worked on publicly available data and made the code used in this study available to facilitate reproducing the results and, hopefully, improving upon these, towards making effective automated pre-screening for DR affordable and accessible.

References

[1] Y. Hatanaka, T. Nakagawa, Y. Hayashi, Y. Mizukusa, A. Fujita, M. Kakogawa, K. Kawase, T. Hara, H. Fujita (2007), "CAD scheme to detect hemorrhages and exudates in ocular fundus images", *(2007 SPIE*

Symposium, vol. 65142M, 2007)

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton (2012), ”Imagenet classification with deep convolutional neural networks, *(Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., pp. 1097–1105. Curran Associates, Inc., 2012.)*

[3] L. W. Yun, U. R. Acharya, Y. V. Venkatesh, C. Chee, L.C. Min, and E.Y.K. Ng (2007), ”Identification of different stages of diabetic retinopathy using retinal optical images”, *(Information Sciences, vol. 178, pp. 106-121, 2008)*

[4] S. B. Hathwar and G. Srinivasa (2019), ”Automated Grading of Diabetic Retinopathy in Retinal Fundus Images using Deep Learning”, *(2019 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), Kuala Lumpur, Malaysia, 2019)*

[5] Ankita Gupta, and Rita Chhikara (2018), ”Diabetic Retinopathy: Present and Past”, *(Procedia Computer Science, https://doi.org/10.1016/j.procs.2018.05.074.)*

[6] Koch, G., Zemel, R., and Salakhutdinov, R. (2015), ”Siamese Neural Networks for One-shot Image Recognition”, *()*

[7] Tianqi Chen, and Carlos Guestrin (2016), ”XGBoost: A Scalable Tree Boosting System”, *(785-794. 10.1145/2939672.2939785.)*

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015), ”Deep Residual Learning for Image Recognition”, *(arXiv:1512.03385v1 [cs.CV])*

[9] François Chollet (2017), ”Xception: Deep Learning with Depthwise Separable Convolutions”, *(1800-1807. 10.1109/CVPR.2017.195.)*

[10] Karen Simonyan, and Andrew Zisserman (2014) ”Very Deep Convolutional Networks for Large-Scale Image Recognition”, *(arXiv:1409.1556v6 [cs.CV])*

[11] International Diabetes Federation (2020) ”Members: IDF South-East Asia Region - India”, *(https://idf.org/our-network/regions-members/south-east-asia/members/94-india.html (Last accessed on 03-31-2020.)).*

[12] Wong TY, Sun J, Kawasaki R, et al. ”Guidelines on diabetic eye care: The International Council of Ophthalmology recommendations for screening, follow-up, referral, and treatment based on resource settings”, *(Ophthalmology. 2018;125(10):1608-1622.)*

[13] O’Shea, Keiron & Nash, Ryan. (2015), ”An Introduction to Convolutional Neural Networks”, *(ArXiv e-prints.)*

[14] Cuadros, Jorge, and George Bresnick. (2009), ”EyePACS: an adaptable telemedicine system for diabetic retinopathy screening”, *(Journal of diabetes science and technology vol. 3,3 509-16. 1 May. 2009, doi:10.1177/193229680900300315)*

[15] Khalifa NEM, Loey M, Taha MHN, Mohamed HNET (2019), ”Deep Transfer Learning Models for Medical Diabetic Retinopathy Detection”, *(Acta Inform Med. 2019;27(5):327-332. doi:10.5455/aim.2019.27.327-332)*