PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

The application of deep learning for diabetic retinopathy prescreening in research eye-PACS

Siliang Zhang, Huiqun Wu, Veda Murthy, Ximing Wang, Lin Cao, et al.

Siliang Zhang, Huiqun Wu, Veda Murthy, Ximing Wang, Lin Cao, John Schwartz, Jorge Hernandez, Gustavo Rodriguez, Brent J. Liu, "The application of deep learning for diabetic retinopathy prescreening in research eye-PACS," Proc. SPIE 10579, Medical Imaging 2018: Imaging Informatics for Healthcare, Research, and Applications, 1057913 (6 March 2018); doi: 10.1117/12.2296673

SPIE.

Event: SPIE Medical Imaging, 2018, Houston, Texas, United States

The application of deep learning for diabetic retinopathy prescreening in a research eye-PACS

Siliang Zhang[a], Huiqun Wu[a,b], Veda Murthy[a], Ximing Wang[a], Lin Cao[a], John Schwartz[a], Jorge Hernandez[a], Gustavo Rodriguez[a], Brent J Liu[a*]

[a]IPILab, Department of Biomedical Engineering, Viterbi School of Engineering, University of Southern California, Los Angeles, United States, CA 90089;

[b]Department of Medical Informatics, Medical School of Nantong University, Nantong, China, 22600

ABSTRACT

The increasing incidence of diabetes mellitus (DM) in modern society has become a serious issue. DM can also lead to several secondary clinical complications. One of these complications is diabetic retinopathy (DR), which is the leading cause of new cases of blindness for adults in the United States. While DR can be treated if screened and caught early in progression, the only currently effective method to detect symptoms of DR in the eyes of DM patients is through the manual analysis of fundus images. Manual analysis of fundus images is time-consuming for ophthalmologists and can reduce access to DR screening in rural areas. Therefore, effective automatic prescreening tools on a cloud-based platform might be a potential solution to that problem. Recently, deep learning (DL) approaches have been shown to have state-of-the-art performance in image analysis tasks. In this study, we established a research PACS for fundus images to view DICOMized and anonymized fundus images. We prototyped a deep learning engine in the PACS server to perform prescreening classification of uploaded fundus images into DR grade. We fine-tuned a deep convolutional neural network (CNN) model pretrained on the ImageNet dataset by using over 30,000 labeled image samples from the public Kaggle Diabetic Retinopathy Detection fundus image dataset[6]. We linked the PACS repository with the DL engine and demonstrated the output predicted result of DR into the PACS worklist. The initial prescreened result was promising and such applications could have potential as a "second reader" with future CAD development for next-generation PACS.

Keywords: deep learning, diabetic retinopathy, fundus image, PACS, ResNet

1. INTRODUCTION

Diabetes Mellitus is classified as a metabolism disorder that could be caused by varying risk factors, from genetic mutations to altered life styles. However, DM can cause several secondary complications as well. One of these complications is diabetic retinopathy (DR) which is the leading cause of new cases of blindness for adults in the United States. In 2010, approximately 7.69 million adults in the United States suffered from diabetic retinopathy, and the number of Americans with diabetic retinopathy is expected to double to approximately 14.6 million by 2050[5].

DR can be treated if screened and caught early in the clinical disease progression. Clinical guidelines recommend DR screening for patients with Type 1 diabetes every year starting five years after their diagnosis, and screening for patients with type 2 diabetes every year starting from the time of diagnosis. Unfortunately, only 60% of DM patients are screened for DR as recommended[7]. This is because the only current effective method to detect symptoms of DR is manual analysis of fundus images. Fundus photography is a reproducible, clinically validated technique used to assess the severity of DR. Analyzing fundus images is a time-consuming process for an ophthalmologist, and for patients in rural areas, access to trained medical professionals may be limited. Automated prescreening methods that analyze the severity of DR cases have the ability to increase the percentage of patients screened for DR as it will reduce the amount of time

Medical Imaging 2018: Imaging Informatics for Healthcare, Research, and Applications, edited by Jianguo Zhang, Po-Hao Chen, Proc. of SPIE Vol. 10579, 1057913 · © 2018 SPIE
CCC code: 1605-7422/18/$18 · doi: 10.1117/12.2296673

needed to diagnose a patient, and provide simple prescreening measures for patients who live in regions without access to trained medical professionals.

In this study, we seek to establish a software augmentation to a widely-used existing clinical ophthalmology software known as eye-PACS. Currently, eye-PACS allows ophthalmologists to view DICOM fundus images sent by fundus cameras. For our developmental work, we augmented an existing WADO viewer and data entry pipeline of eye-PACS with a module that will produce automated predictions of the grade of a fundus image. We built a Convolutional Neural Network pretrained on the ImageNet ILSVRC dataset by using 30,000 normalized labeled images from the Kaggle Diabetic Retinopathy Detection dataset[6]. We then integrated the WADO viewer system with our CNN and display predictions alongside fundus images displayed in eye-PACS. Thus, ophthalmologists can spend less time reviewing the images for a given patient to determine salient DR grade information.

In developing the Convolutional Neural Network, the prescreening is treated as a classification task, where an input fundus image is classified into one of five grades of severity of DR. To classify DR manually, an ophthalmologist looks for predefined pathologies in each image. Our CNN is trained to produce a similar output prediction as an ophthalmologist by training it on a dataset of 30,000 labelled fundus images from the Kaggle Diabetic Retinopathy Detection dataset[6]. We utilized ResNet CNN architecture, which achieved state of the art performance on the ImageNet Large-Scale Visual Recognition competition and the Microsoft Common objects in Context Challenge in 2015[10]. We initialized ResNet with pretrained weights trained for the ILSVRC challenge, then fine-tune the weights on the Kaggle DR dataset, as this has been shown from prior works to improve accuracy in both common and medical imaging tasks[17, 21]. While works have been done in the past for utilizing predefined features and machine learning methods such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) classifiers, our work is unique in integrating a state-of-the-art architecture CNN with existing PACS infrastructure. Finally, the trained CNN is integrated into a research eye-PACS as a computer aided detection module.

In our previous work, we developed a WADO viewer-based Research eye-PACS which stores fundus images of diabetes patients and allows users to manipulate those images through a web interface[22]. Moreover, CAD-structured report(CAD-SR) has been integrated into the system successfully[23]. In this study, we aim to incorporate the research eye-PACS with trained CNN model and use the detected result from computer to enhance the CAD-SR and refine the traditional eye-PACS workflow. We also redefined the workflow of the research eye-PACS with a deep learning module. In the future, our Research-PACS could be used to evaluate the feasibility of applying deep learning algorithm in the clinical setting and whether use of the deep learning engine could lead to improved care and outcomes compared with current ophthalmologic assessment.

2. MATERIAL AND METHODS

2.1 Research eye-PACS system development

The hardware specifications of our eye-PACS system are as follows: CPU: Intel i5 1.5GHz; RAM: 1024MB; Operating System: Ubuntu 14.04; Database MySQL 5.0; Hard disk: 40 GB. To build the website, PHP 5.5; Apache 2 and Kohana framework 3.2 were used.

The WADO viewer-based research eye-PACS was designed as browser/server architecture. Users can get access to it through a very simple website interface. After a successful registration, users can login to the website and upload fundus images by entering the name they want to use to anonymize the study. The eye-PACS is equipped with a module to DICOMize uploaded JPEG files, and uploaded images are stored and manipulated in DICOM format. Users can also fill the case report forms and download the complete report as a separate file in DICOM format.

2.2 Dataset and development environment

For the development of algorithm, the retinal fundus images were collected from the public Kaggle Diabetic Retinopathy Detection competition dataset[6]. As shown in Table 1, this dataset contains 35,126 fundus images from existing eye-PACS users as released by the California Healthcare Foundation. In terms of labeling, a clinician has rated the severity of diabetic retinopathy in each image on a scale of 0 to 4, according to the scale in Table 2. The images in the dataset come from different models and types of cameras, which can affect the visual appearance and resolution of the images. As shown in Figure 1, some images in the dataset contain artifacts, are out of focus, underexposed, or overexposed. Thus, our algorithm must be robust to variations in color and resolution. The algorithm was designed with the diversity of images of our dataset in mind. A portion of the total dataset of 10% of the total 35,126 training images was set aside for our training dataset for hyperparameter selection and validation.

Table 1. The number of images used for training, hyperparameter validation and testing, respectively.

Dataset	Number of Images
Training Dataset (Complete)	35,126
Training Images Used	32,005
Validation Images Used	3121
Testing Images Used	53,584

To train our CNN, the development environment is equipped with an NVIDIA GeForce Titan Xp GPU. Utilizing a GPU for training a CNN can accelerate computational speed, and as a result we were able to train our model in under 24 hours[19]. The GPU development environment contains CUDA 8 and CuDNN 6 for interfacing with the GPU. The deep learning package Keras was used with the TensorFlow as the backend[1, 3]. This was chosen because of good documentation and short calculation time. We use OpenCV to preprocess our input images[2].

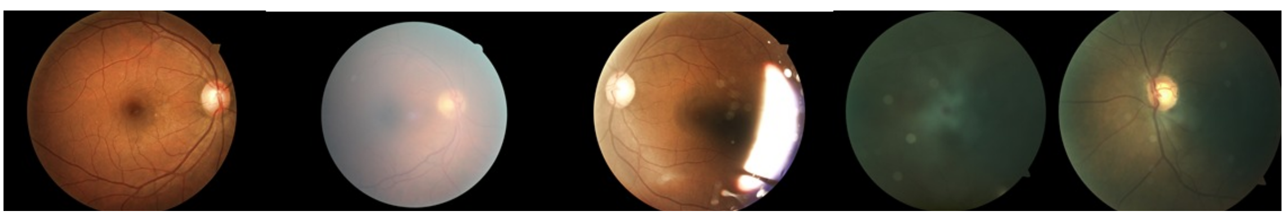

Figure 1. Example images from the Kaggle DR Detection training dataset Images were taken with varying cameras and under varying illuminations. Some images contain artifacts (third image) or are out of focus (fourth image), or vary in whether the image is macula-centered or optic disk centered.

2.3 Preprocessing

To remove some of the variation between images due to differing lighting conditions, camera resolution and so on, each image is preprocessed to be of a standard resolution and standard cropping. Similar to the procedure of the Kaggle Competition winner, the original data were preprocessed through the following steps. We rescaled the images to have the same radius (~300 pixels) and then subtracted the local average color which was mapped to 50% grayscale level. Finally, we clipped the images to 90% size to remove the "boundary effects" (Figure 2)[9]. This cropped image is then resized to 224 x 224 pixels. At runtime, the image is subtracted the VGG mean color channel intensities before feeding into the CNN[20].

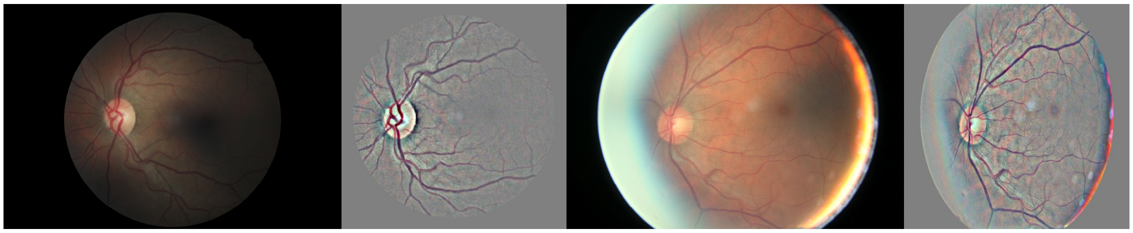

Figure 2. Two preprocessed images from training dataset.

Examples of our preprocessing method are shown in Figure 2. Original images are on the left and preprocessed images are on the right. The image above is under poor illumination, and the image below contains illumination artifacts. Note that our preprocessing method reduces the variations of the two images and corrects for variations in illumination through local normalization.

Table 2. The DR grades algorithm is trained to distinguish between levels of DR.

Rating	Grade
0	No DR
1	Mild DR
2	Moderate DR
3	Severe DR
4	Proliferative DR

2.3 Algorithm development

Due to the state of the art performance in medical image analysis and classification tasks, CNN is chosen for the basis of our algorithm. Many different architecture CNNs have been benchmarked recently. ResNet-50 was chosen as the structure of our neural network after studying the literature for other recognition tasks. ResNet won the 1st place in the ILSVRC classification competition and MS COCO segmentation task in 2015, and its fully convolutional design lacks the millions of parameters of previous VGG-16 and AlexNet architecture CNNs[11, 15, 20].

ResNet-50 has 50 layers. The first layer attempts to learn the abstract features like edges, and the deep convolutional layer attempts to learn high level features such as hemorrhages or exudates. All convolutions are performed with kernel size 3×3 and 1×1 except the first convolution layer which has a kernel size of 7×7. After the final convolutional block, the fully connected layer flattens the tensor into one dimension. The identity blocks of ResNet have RELU as the activation function and batch normalization after each convolution layer. To avoid overfitting, we added a dropout layer to the convolutional block (Figure 3). Inputs are dropped at a rate of 50%, and dropout is added on the non-skip connection pathway of the convolutional block of ResNet-50.

In order to get a good optimization result, we utilized transfer learning as it has been shown to increase accuracy on medical imaging tasks[17, 21]. First, the weights of ResNet-50 was initialized with weights trained for ImageNet ILSVRC classification. Then, the network was trained using stochastic gradient descent with Adam optimization algorithm[14]. The learning rate is 0.0001, beta-1 is 0.9 and beta-2 is 0.99. The learning rate will be adjusted along with the substantial 100 epochs of training on whole dataset. The loss function used to optimize was the widely used categorical cross-entropy function. We weight the loss of pathological classes (DR classes 1-4) by dividing the number of negative images by the number of images of the given pathological class. This weights our algorithm toward the correct classification of pathological classes, which is critically important in creating a prescreening tool.

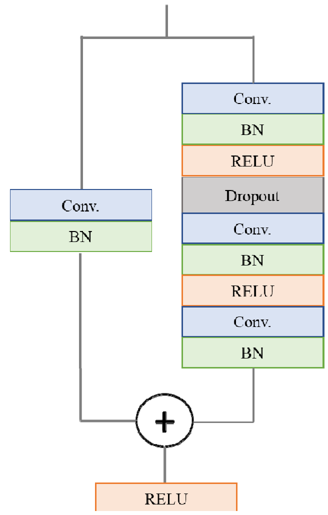

Figure 3. The convolutional blocks where we added dropout. ResNet-50 Convolutional blocks contain two pathways that are summed together at the end of the block; we apply 50% dropout to one of the pathways to regularize our model.

3. RESULTS

A total of 53,584 images were used to evaluate the performance of our model. When compared against the test dataset, our algorithm achieved a quadratic weighted kappa of 0.64. It would have achieved a top 6% position of submissions for the competition even with the minimal preprocessing and lack of model averaging found in other top scoring submissions[6]. In addition, the results are generated on all test images versus the subsets used for the Kaggle competition itself.

In addition, the ROC curve is shown below. We compare the predictions of our algorithm against the DR grade annotations of a clinician. The trained model achieves an area under the receiver operating curve (AuROC) of 0.83. Our trained model achieves 84%specificity and 61% sensitivity across the full test dataset, surpassing the result of Pratt et. al. in terms of sensitivity. In addition, our confusion matrix is similar to that reported in Pratt et al in that our model struggled to differentiate between some of the mild DR classes (DR label 0 versus 1)[18]. This may indicate that CNN was unable to learn fine differences between classes.

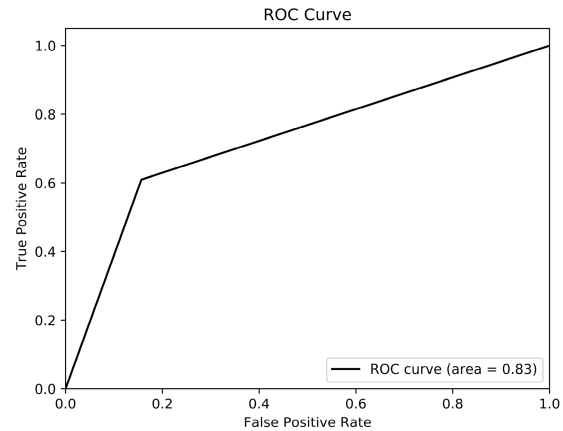

Figure 4. The receiver operating curve comparing our CNN's performance against the gold-standard labels created by a clinician.

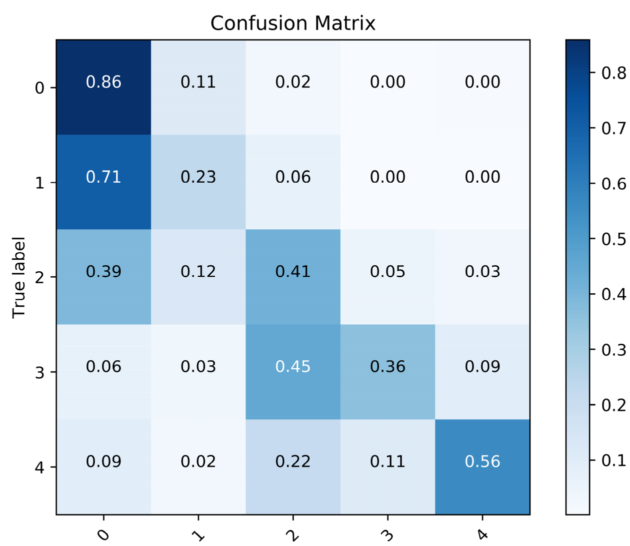

Figure 5. The confusion matrix representing our CNN's performance.

3.1 The integration of a deep learning engine with a research eye-PACS server

We integrate our CNN with the research eye-PACS system described previously after training was completed. The workflow of our research Eye-PACS system is shown in Figure 7. After the fundus images were captured from fundus camera, they are transferred to server and converted into DICOM files using Pydicom, a Python package. Then the DICOM files are stored in our eye-PACS server.

- Workflow Step 1 depicts the fundus camera. Fundus images can be acquired from remote fundus cameras, and sent to our eye-PACS server through a browser-based interface.
- Workflow Step 2 depicts our eye-PACS server. It retrieves and stores uploaded fundus images from distant fundus cameras.

- Workflow Step 3 depicts the GUI. The GUI retrieves images from the eye-PACS server to show on screen. In addition, it sends entered user data back to the eye-PACS server for storage.

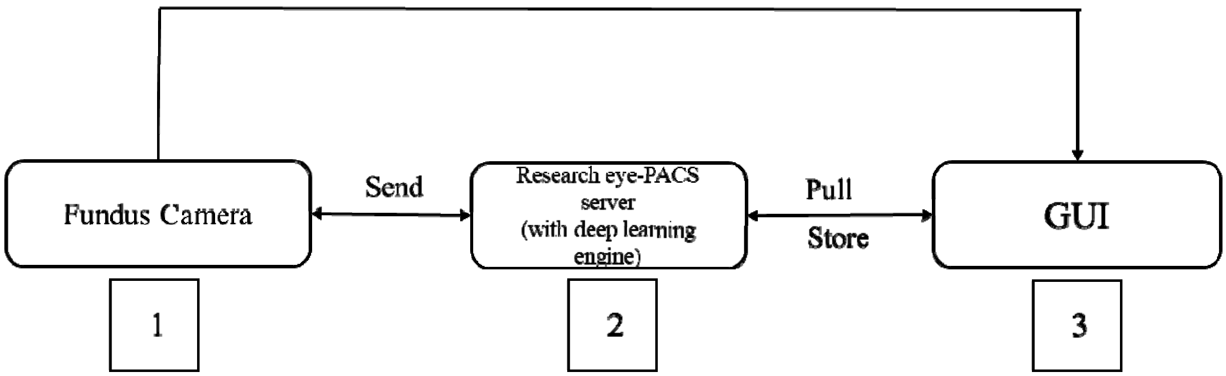

Figure 6. The workflow of eye-PACS system.

The research eye-PACS server was augmented to contain a deep learning engine to handle processing image inputs using neural network, and transferring the classification results to the eye-PACS database. The deep learning engine can fetch DICOMized files, and save the predictions into the database. Predictions can be displayed on the user web-based interface of our eye-PACS server where medical professionals can view the result alongside the original input fundus image as shown in Figure 8. Then, based on the result of prescreening the ophthalmologists can filter out the patients whose fundus images have been classified as not having DR and only pull images from patients flagged as having DR from the research eye-PACS server. Our system thus allows ophthalmologists to prioritize writing CAD-SR reports for patients who have been flagged as having DR versus all patients equally.

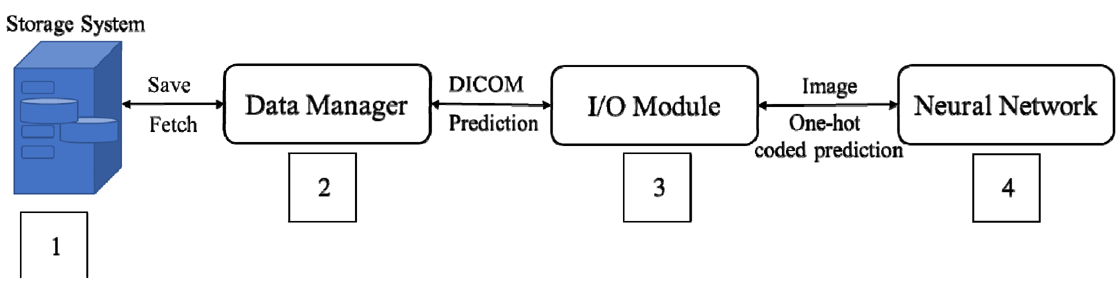

Figure 7. Depicts our workflow for the deep learning engine and will be described in more detail below.

- Workflow Step 1 depicts our storage system, consisting of a SQL database on the research eye-PACS machine. It stores DICOMized fundus images along with patient data entered by clinicians on the browser interface.
- Workflow Step 2 depicts our data manager. It DICOMizes input images and allows our software to access stored data on the server side. It sends image data between the I/O module and the GUI. Based on predictions that have been created by the neural network and stored in the storage system, it is able to transfer predicted grades to the GUI for display on screen.
- Workflow Step 3 depicts our I/O manager. It extracts the image data from the DICOMized fundus image, and sends the image data to the neural network for prediction. When the prediction is created, it sends the data back to the Data manager so that it can be stored in our storage system. This mechanism prevents repeat predictions for the same image, saving computational time.

- Workflow Step 4 depicts our neural network. It has been designed to operate on our eye-PACS server. It generates an output predicted DR grade for each input image, then sends the prediction back to the I/O manager for storage and display on the GUI.

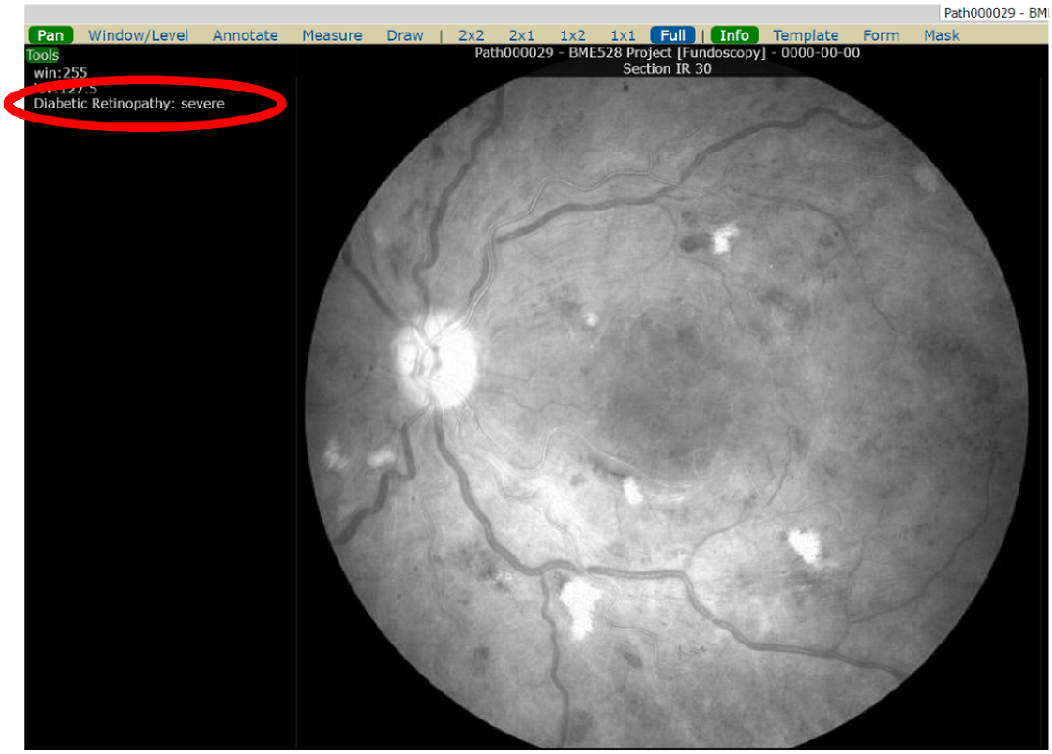

Figure 8. A Screenshot of the web-based GUI. The severity of diabetic retinopathy is displayed alongside the image for more effective prescreening built-into the GUI.

4. DISCUSSION AND CONCLUSION

Our study has shown that problem for pre-screening of DR can be approached using CNN. Our initial prototype model has shown the ability of learning enough features to grade the level of DR of fundus images. Furthermore, our results produced comparable results to the previous published methods. Although our model struggled with differentiating between mild DR and no DR. We believe this may be due to the low resolution at which images are processed. Fine details, such as exudates and hemorrhages may be missed at a lower resolution. The potential benefit of our research eye-PACS system is that it can be used to increase the efficiency and reproducibility of screening program and provide simple prescreening measures for patients who live in regions without access to trained medical professionals. It could also improve patient outcomes by providing early detection and treatment.

In the future, we would like to expand our CNN to give more detailed predictions. Specifically, we would like to localize pathology locations versus generating an output classification for each pathology. This will enable more generalizability between classes as all pathological classes may share pathologies in some form. In addition, localization-level data will provide medical professionals with more detailed analysis of fundus images. We also plan on using our augmented research eye-PACS server to evaluate the efficacy of the deep learning algorithm under clinical conditions and determine whether our prescreening algorithm is effective in improving the ophthalmology workflow.

In conclusion, CNN has shown the potential to identify the level of diabetic retinopathy in fundus images. The research eye-PACS system has the potential to maximize the clinical utility of automated grading and be useful to ophthalmologists with an integrated CNN platform. As the method and dataset improves, so will the potential to offer more support in the clinical environment.

ACKNOWLEDGEMENTS

This work was supported by a GPU grant from NVIDIA Corporation.

REFERENCES

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., et al., "TensorFlow: A System for Large-Scale Machine Learning," 12th USENIX Symp. Oper. Syst. Des. Implement. (OSDI '16), 265–284 (2016).

[2] Bradski, G. "The OpenCV Library," (2000). https://github.com/opencv/opencv

[3] Chollet., F., et al., "Keras," https://github.com/keras-team/keras

[4] Ciresan, D. C., Giusti, A., Gambardella, L. M. and Schmidhuber, J., "Mitosis Detection in Breast Cancer Histology Images using Deep Neural Networks," Proc Med. Image Comput. Comput. Assist. Interv., 411–418 (2013).

[5] "Diabetic Retinopathy," https://nei.nih.gov/eyedata/diabetic

[6] "Diabetic Retinopathy Detection," https://www.kaggle.com/c/diabetic-retinopathy-detection

[7] "Diabetic Retinopathy PPP," https://www.aao.org/preferred-practice-pattern/diabetic-retinopathy-ppp-updated-2016

[8] "Eye-PACS System," http://www.eyepacs.com/eyepacssystem/

[9] Graham, B. "SparseConvNet Diabetic Retinopathy Detection," https://github.com/btgraham/SparseConvNet/ tree/kaggle_Diabetic_Retinopathy_competition

[10] Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., Kim, R., Raman, R., Nelson, P. C., Mega, J. L. and Webster, D. R., "Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs," Jama 316(22), 2402 (2016).

[11] He, K., Zhang, X., Ren, S. and Sun, J., "Deep Residual Learning for Image Recognition," 2016 IEEE Conf. Comput. Vis. Pattern Recognit., 770–778 (2016).

[12] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. R., "Improving neural networks by preventing co-adaptation of feature detectors," 1–18 (2012).

[13] Ioffe, S. and Szegedy, C., "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Proc. Mach. Learn. Res. (2015).

[14] Kingma, D. P. and Ba, J., "Adam: A Method for Stochastic Optimization," 1–15 (2014).

[15] Krizhevsky, A., Sutskever, I. and Geoffrey E., H., "ImageNet Classification with Deep Convolutional Neural Networks," Adv. Neural Inf. Process. Syst. 25, 1–9 (2012).

[16] LeCun, Y., Bengio, Y. and Hinton, G., "Deep Learning" (2016).

[17] Oquab, M., Bottou, L., Laptev, I. and Sivic, J., "Learning and transferring mid-level image representations using convolutional neural networks," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 1717–1724 (2014).

[18] Pratt, H., Coenen, F., Broadbent, D. M., Harding, S. P. and Zheng, Y., "Convolutional Neural Networks for Diabetic Retinopathy," Procedia Comput. Sci. 90(July), 200–205 (2016).

[19] Shi, S., Wang, Q., Xu, P. and Chu, X., "Benchmarking State-of-the-Art Deep Learning Software Tools" (2016).

[20] Simonyan, K. and Zisserman, A., "Very Deep Convolutional Networks for Large-Scale Image Recognition," Int. Conf. Learn. Represent., 1–14 (2014).

[21] Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B. and Liang, J., "Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?," IEEE Trans. Med. Imaging 35(5), 1299–1312 (2016).
[22] Wu, H. *et al.,* "The Design and Integration of Retinal CAD-SR into Diabetes Patient EPR System," SPIE Med. Imag., (2017).
[23] Wu, H. *et al.,* "The development of an ophthalmologic imaging CADe structured report for retinal image radiomics research," SPIE Med. Imag., (2018).