DEEP LEARNING LAB - WINTER TERM 21/22
DIABETIC RETINOPATHY DETECTION

A PREPRINT

Peter Bauer[*]
Electromobility
University of Stuttgart
st148923@stud.uni-stuttgart.de

Kai Pan[†]
Electromobility
University of Stuttgart
st171112@stud.uni-stuttgart.de

February 14, 2022

ABSTRACT

This work addresses the automatic detection of diabetic retinopathy (DR), an eye disease caused by diabetes, using state-of-the-art deep learning technologies. Although there are many interesting aspects when developing a deep learning pipeline, we limit ourselves to presenting some preprocessing steps, different model architectures, and the implemented deep visualization techniques.

1 Introduction

Since DR is the most prevalent cause of avoidable vision impairment, early and regularly monitoring of the eye is a necessary procedure and has a lot of optimization potential. Fast and reliable detection systems are needed to select individuals to be treated thoroughly. Our approaches for building such automated systems are based on deep learning methods for image processing. After a short section about the data pipeline and image preprocessing, we present several architectures ranging from simple Convolutional Neural Network (CNN) structures to more complex ensemble and transfer models. Furthermore, we present state of the art deep visualization methods to gain a better understanding of the decision process of CNN models. For all our experiments, we utilized the Indian Diabetic Retinopathy Image Dataset (IDRID) [2]. [PB]

2 Data Pipeline & Preprocessing

IDRID contains a total of 516 images and is therefore quite small. To enlarge the data, several image augmentation techniques including rotation, cropping, change of contrast/brightness and flipping are used. Since the images are of good quality, we did not try to enhance any textures using classical computer vision techniques (although this could enhance the model performance even further). All images contain a large number of black pixels, which had to be reduced before downsampling to the final resolution of 256 x 256. The original images are non-square and a naive approach for downsampling a picture without any distortions would be to add zero pixels to the edges until one obtains a square image (padding of the picture). However, this approach introduces even more black spaces that are useless, resulting in an information loss per pixel. Cropping the interesting region "*by hand*" is also not a suitable opportunity. The best crop is different for each image and information could be easily lost by having wrong cropping parameters for a certain image. This led us to an automated cropping procedure that automatically processes each image to retain the relevant content. In detail, this procedure searches for the left- and rightmost non-black pixel and crops the black sides off. Afterwards, the image is padded to square and gets downsampled as required. This simple procedure ensures that the images still contain all the relevant information, while the number of black pixels is minimized. [PB]

[*] [PB]
[†] [KP]

3 Models

For image processing CCNs are widely know to be very effective. While single neurons only have a sparse connection to the previous layer, CNNs are still capable of implicitly extracting larger spatial dependencies through the information propagation across layers. At the same time, the computational complexity of CNNs is dramatically reduced in comparison to dense networks. Furthermore, it is shown that a stack of convolutional and pooling layers become invariant to a (large) input shift. In other words, an image feature can be detected regardless of its spatial occurrence. [PB]

3.1 Team20_CNN's

Our first approach was a classical CNN architecture as illustrated in 1. The first two layers of the Team_20_CNN_01 model are self-defined CNN-BP blocks. Each block consists of a convolution layer followed by maxpooling and batch normalization. After another convolutional layer, global average pooling (GAP) is performed to flatten the data. Instead of directly reducing the network to the desired output size, another fully connected layer is added. In the end, a (sparse) categorical cross entropy loss is applied to the model output. To optimize the hyperparameters (stride, kernel size, etc.) a bayesian search was performed using *Weights & Biases*.

We came up with two more models that follow the same general structure (Team_20_CNN_02 and CNN_Blueprint). While Team_20_CNN_02 adds some additional convolutional layers, the CNN_Blueprint is designed as flexible architecture that can be quickly adapted. [PB]

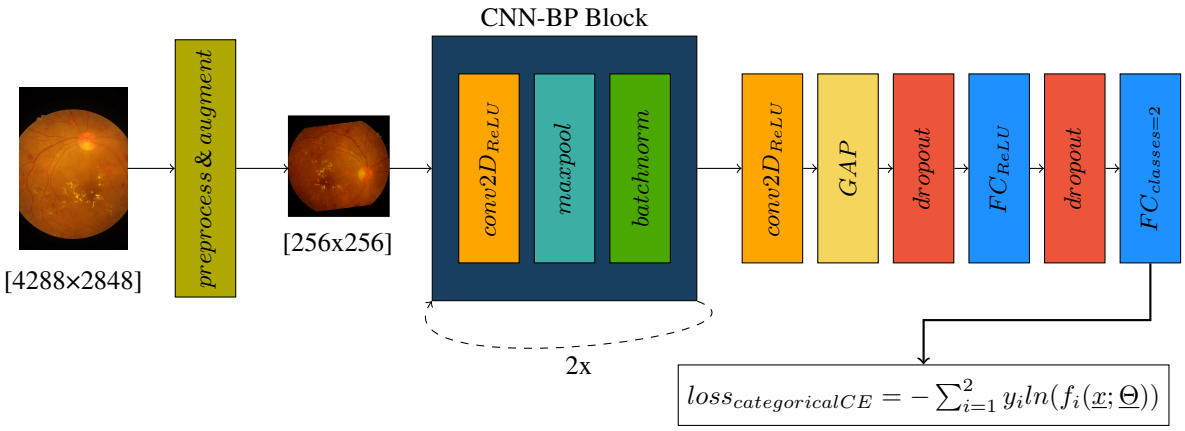

Figure 1: Pipeline of the Team20_CNN_01 model. [PB]

3.2 Resnet & VGG

In addition to the self defined architectures, the well known Resnet and VGG blocks were used to build models. [PB]

3.3 Ensemble Learning

The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability and robustness over a single estimator [1]. In our work, we employed voting and average methods for ensamble leaning. While the voting method counts the classification of each model directly, the average method first gets the average prediction of all applied models and then puts it into the metric. Both ensemble learning methods use the three best trained models, namely VGG, Resnet, and Team20_01. The result is shown in Table 2. [KP]

3.4 Transfer Learning

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. We chose two models pre-trained on ImageNet to do transfer learning, which are Xception and InceptionResnet. During training, a few of the top layers of the frozen transfer learning model are unfrozen, except

the batch normalization layer, which would destroy the trained structure of the base model if retrained. Two dense layers and Dropout layer will be added at the top of base model. The number of unfrozen layers, dropout rate and dense units number will be considered as hyperparameters, which are tuned with "*Weights & Biases*". The result of transfer learning has been shown in Table 2. [KP]

4 Deep Visualization

The various CNN models above have high accuracy in this classification problem but the content and rules they learn are difficult to be understood by humans. Commonly known as "black boxes", neural networks are systems that hide their internal logic to the user. It is very helpful to understand the prediction process of the models, which enables one to explain why a particular prediction choice was made. The deep visualization methods are used to achieve this, and some of the most common methods are Grad-CAM [3], Guided Backpropagation [4], Guided Grad-CAM [3] and Integrated Gradients [5]. Grad-CAM uses the gradients of any target concept, flowing into the final convolutional layer to produce a coarse localization map highlighting important regions in the image for predicting the concept [3]. Whereas in Guided Backpropagation, we only keep the influence of positive gradients on the class score, and suppress the ones that have negative influence, in order to obtain much cleaner looking images. The heat map combination of above two is Guided Grad-CAM. The integral gradient method expects the contribution of non-zero gradients in the non-saturated region to the decision importance by integrating the gradients along different paths. We use a noisy image as the integration baseline and the integration path is chosen as linear interpolation. The result of these methods on the best trained Team20_CNN_01 model has been shown in Figure 3. Different symptoms associated with Diabetic Retinopathy such as hemorrhages and hard exudates have been clearly detected. [KP]

5 Evaluation

In order to evaluate the proposed methods, *accuracy*, *sensitivity*, *specificity* and *f1-score* are used as performance measures. Since the test data is not resampled to the count majority class, the *balanced-accuracy* is introduced as additional metric.
All models are trained and evaluated five times in a row. Thereafter, the best run of each model is selected as final score. We regularly encountered significant performance differences between models of the same type (≈10% for some models), and therefore provide the log-files of all test runs on our GitHub[3] repository. [PB]

Model	Accuracy [%]	Balanced-Accuracy [%]	Sensitivity [%]	Specificity [%]	F1-Score [%]
Team20_CNN_01	81.6	80.3	73.8	86.9	**86.8**
Team20_CNN_02	80.6	79.4	74.3	84.4	74.3
CNN_Blueprint	78.6	77.3	71.8	82.8	71.8
Resnet	84.5	84.3	72.5	96.2	82.2
VGG_Like	85.4	85.1	74.0	96.2	83.1
InceptionResnet	81.6	80.4	72.7	88.1	77.1
Xception	81.5	81.2	70.0	92.4	78.7
Ensemble (vote)	**87.4**	86.7	**77.1**	96.4	85.1
Ensemble (avg.)	**87.4**	**87.1**	76.0	**98.1**	85.4

Figure 2: Detailed performance measurements of all tested models. [PB]

6 Conclusion

In this paper we presented numerous architectures to solve the DR detection problem, ranging from basic CNNs to more complex transfer and ensemble methods. In particular, an increase in performance was observed for ensemble methods that combine the best models via voting or prediction averaging. During all experiments, the validation accuracy was noticeably higher than the test accuracy. Since the validation images are randomly obtained from the original IDRID training data, we believe that there is a (small) *dataset shift* between the training and the test set. Both problems (fluctuating model performance and higher validation than test accuracy) are therefore most likely due to the small dataset and could be solved by collecting more samples of the same quality.

[3] https://github.tik.uni-stuttgart.de/iss/dl-lab-21w-team20

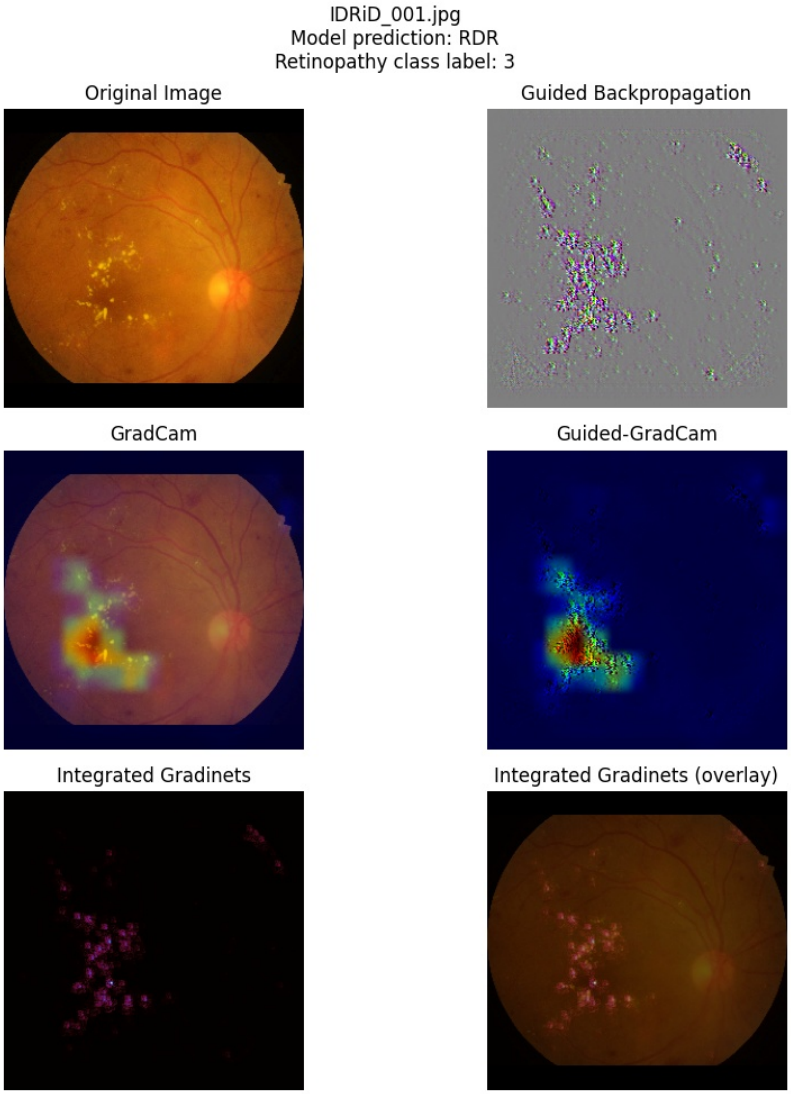

Figure 3: Deep visualization of an example image from the IDRID dataset [PB] & [KP]

References

[1] Thomas G Dietterich et al. Ensemble learning. *The handbook of brain theory and neural networks*, 2(1):110–125, 2002.

[2] Prasanna Porwal, Samiksha Pachade, Ravi Kamble, Manesh Kokare, Girish Deshmukh, Vivek Sahasrabuddhe, and Fabrice Meriaudeau. Indian diabetic retinopathy image dataset (IDRiD): A database for diabetic retinopathy screening research. *Data*, 3(3), sep 2018.

[3] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[4] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[5] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.