Report Deep Learning 2024

Emanuele Pocelli
University of Oulu
emanuele.pocelli@student.oulu.fi

Prosper Playoust
University of Oulu
prosper.playoust@student.oulu.fi

Valentin Wolfer
University of Oulu
valentin.wolfer@student.oulu.fi

1. Introduction

Diabetic Retinopathy is a complication of diabetes that affects the human eye, specifically the retina [1]. In this project, we employ a deep learning model to predict the presence of diabetes and its severity starting from an image of the eye. The goal of the project is to achieve the highest kappa score. To do that, we experiment with different techniques and we see how they affect the final performance. The metric used is the kappa score, which combines accuracy, precision and recall. It gives a more comprehensive assessment on the model performance.

The techniques we will test are: image augmentation, different fine tuning approaches, incorporation of attention mechanisms, ensemble methods and application of different preprocessing techniques.

The dataset [7] is composed of images of the fundus, which is the back of the human eye. The goal is to predict, for each image, whether the patient shows signs of diabetes. If diabetes is present, the model also assesses its severity.

We achieved a kappa score of 0.8349 on the test set predictions submitted on kaggle (team name: CiaoCiao).

2. Fine-tuning

Deep neural networks are powerful and require a lot of data and resources to be trained effectively from scratch. However, sometimes gathering data is too expensive, for instance in the medical field. Because of this, it is common to resort to transfer learning, which is the concept of choosing a pre-trained model and fine-tuning it on a smaller dataset [3]. The reason why it works is that in neural networks, the first layers usually perform feature extraction, while the last layers work on fine-grained features to make predictions. So, we can replace the final layers of a model while keeping the initial ones.

2.1. Models

Firstly, we fine-tuned different pre-trained models using the DeepDRiD dataset. The idea is to investigate how a model that is pre-trained on a general dataset can adapt to a different task. The models we used are ResNet18, EfficientNet, and DenseNet. All of them have been trained on the ImageNet dataset.

We replaced the original fully connected layer with a new one composed:

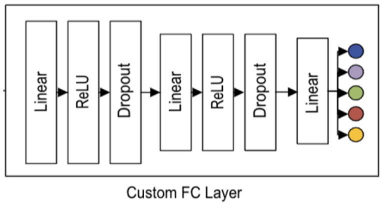

Figure 1. Fully connected layer

The dataset is small and the models are powerful. In order to achieve good performance, it is necessary to use augmentation techniques to reduce overfitting. In this first part, we evaluate the effect of augmentation on the final performance.

2.2. Effect of augmentation methods

We tested different augmentation methods (Random rotation, SLORandomPad, Flip, CutOut, Jitter). First, we used them individually, then we tried different combinations and finally we used all of them.

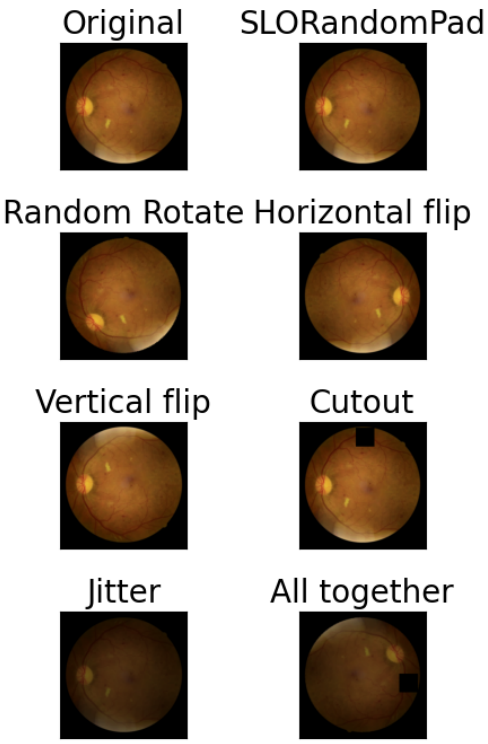

Figure 2. Augmentation methods

Using only one augmentation function barely affects the final result. Instead, using them in combination seems to reduce the overfitting: the loss curve is smoother, indicating a more stable training. For the EfficientNet and DenseNet models, the augmentation has a negative effect, as the performance is slightly better without. This can be due to the fact that the models are smaller and suffers less from overfitting. Below, an example of the effect of augmentation is provided.

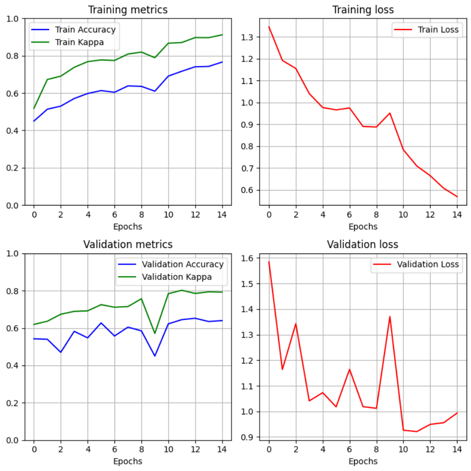

Figure 3. DenseNet training plot without augmentation

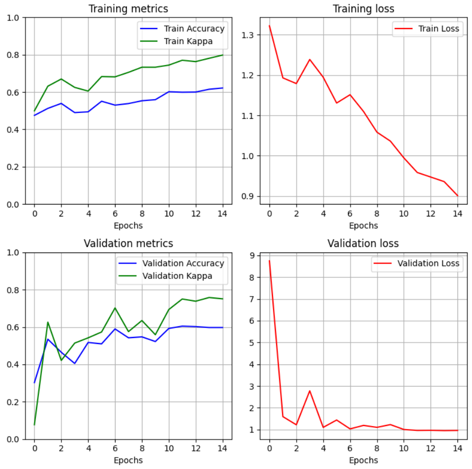

Figure 4. DenseNet training plot with all augmentation functions combined

In the following table we provide the final kappa score achieved by the employed models. ResNet18 is the biggest of the 3 (with around 11 million parameters) and incurs in more overfitting. The other models, EfficientNet and DenseNet, are smaller (respectively 5 and 8 million parameters). They still incur in overfitting, however to a lesser extent. EfficientNet proved to be the best compromise.

Model	No augmentations	Augmentations
ResNet18	0.7916	0.7909
EfficientNet	0.8256	0.8086
DenseNet	0.8024	0.7582

Table 1. Kappa score with and without augmentation

3. Alternative fine-tuning approach

In this section, we fine-tune the models using a different approach. We use a supplementary dataset, Aptos2019, which also focuses on diabetic retinopathy. First, we train the models unfreezing all the layers on this dataset. After that, keeping the backbone layers unfrozen, we further train the model on the DeepDRiD dataset.

Here are the results obtained on the Aptos 2019 and the DeepRid datasets:

Model	Aptos 2019	DeepDRiD
ResNet18	0.8060	0.7002
EfficientNet	0.8251	0.7405
DenseNet	0.8142	0.7824

Table 2. Kappa score on the different datasets

We got good results on the Aptos2019 dataset, while the results achieved on the original one are worse than before. We think that unfreezing the backbone layers of the model while training on a different dataset may have bad effects. Furthermore, the Aptos2019 dataset is bigger and, due to computational limitations, we couldn't experiment with hyperparameters much.

4. Incorporation of attention mechanism

We investigated the effect of attention mechanisms on the model's performance. The attentions used are MultiHeadAttention, Channel Attention and Spatial Attention. For this part, we used the ResNet18 model.
Channel Attention and Spatial Attention work, respectively, on the channels and on the spatial dimensions (height, width) of the image and allow the model to focus more on important features and suppress irrelevant ones. We used them in conjunction and we implemented them in the final part of the backbone. Since this attention allows the model to capture more complex patterns, we expect the model to be more powerful and thus to incur in even more overfitting. We followed the implementation of Song et al. [9].
MultiHeadAttention works on a one dimensional vector and allows to focus on different parts of the vector at the same time. It is commonly used for NLP tasks, to catch the relation between different words in a sentence. We implemented attention on the connection between the backbone and the fully connected layers of the model. The one dimensional vector is composed of the features extracted by the backbone. We do not expect any improvement using this kind of attention because the extracted features are abstract, and the attention mechanism may not add significant value in this context.

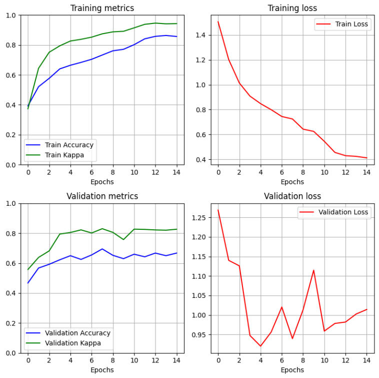

Figure 5. Channel and Spatial Attention

The training plots confirm what we anticipated. There is some overfitting even with all augmentation functions employed.

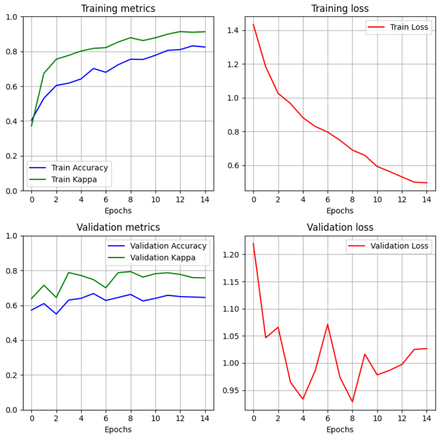

Figure 6. Multihead Attention

Also in this case, the plots confirm what we anticipated. The effect of this type of attention may even be detrimental.

5. Ensemble learning

Then, we tested the effect of ensemble methods [6]. Our models incur in overfit, so the most suitable methods are stacking and bagging. We will not try boosting, as it will probably increase the overfitting even more.
For stacking, we use the models trained in section 3. We use max voting and logistic regression.
Max voting consists in getting the results for the 3 models and then choose the most voted class. For logistic regression, the outputs from these three models are used as inputs to a new model, which is trained to optimize the parameters of the softmax function. Max voting has a very positive effect on performance, while logistic regression yields a worse result. The training of the model was problematic from the computational point of view and unfortunately, because of limited resources, we were not able to train for more epochs and to fine tune the hyperparameters.
For bagging, we used the ResNet18 model with Channel and Spatial Attention. Bagging is supposed to reduce the effect of overfitting, but we got some unexpected results as the performance we get is even worse than before. We are not sure what the reason is, if there is an error in the bagging implementation or if the choice of hyperparameters is poor. In this case, the limited resources are again a big limitation, as bagging is very intensive computationally.
Here are the results :

Ensemble method	Kappa Score
Stacking	0.6940
MaxVoting	0.8241
Bagging	0.6863

Table 3. Kappa score of the ensemble methods

6. Image preprocessing

Finally, we experimented with different preprocessing techniques for the images. These techniques can enhance the representation of the image and hopefully can make it more informative. The methods used are Ben Graham's preprocessing [10], CLAHE [5], circle cropping [4] and Gaussian blur. After trying it, we discarded the circle cropping method as the important features of the image are already located in a circle and the rest is disregarded by the model (as we will see in the next section). For this section, we used the DenseNet model. All the images in training, validation and test are transformed using these preprocessing methods.

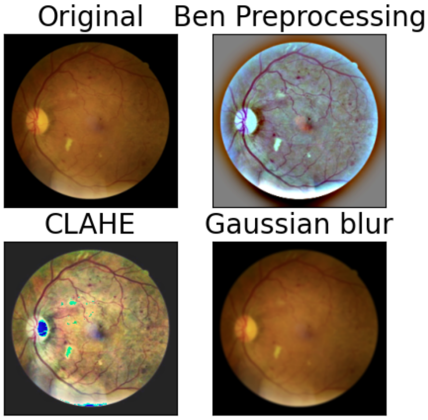

Figure 7. Preprocessing examples

The results obtained are slightly better, as the new representations are more informative. Below are the results.

Preprocessing method	Kappa Score
Ben Graham	0.8216
histogram equalizing	0.8004
Gaussian blur	0.8188

Table 4. Kappa score using preprocessing

7. Explainability

In this final section, we focused on explainability. Using the GradCam method [8], we gained insights into why the model makes a certain prediction. GradCam highlights areas of the image that contribute the most for the model decision. It can operate on specific parts of the model. We analysed some of the first and some of the last layers, but here we will comment only on the most interesting visualizations, though additional ones can be found in the notebook. The most important areas are highlighted in red and the least important ones in blue.

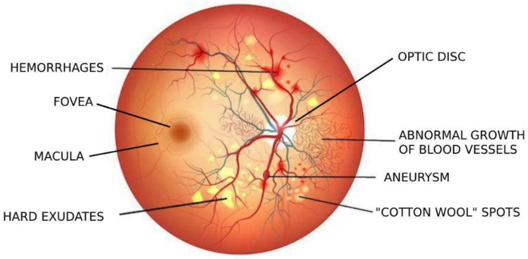

Figure 8. Fundus image

This is an example of fundus image [2]. The captions explain the different parts that can be seen in the image. Understanding these regions can help better understand the model's decision-making process.

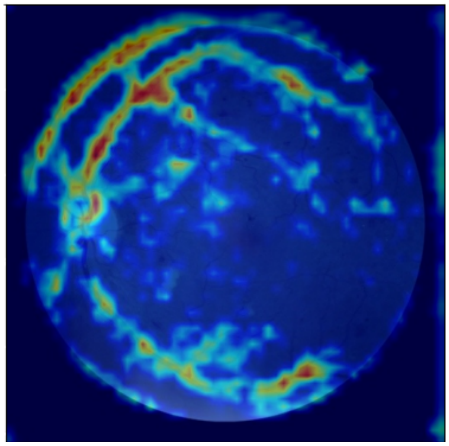

Figure 9. Dense block 1

The backbone of the model is composed of 4 blocks of layers. Here, we focus on the very first block, consisting of multiple convolutional layers. The model is still working on low level features, like the contours.

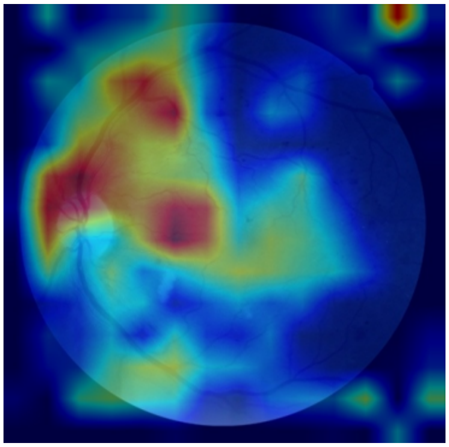

Figure 10. Transition layer 3

Here, we focus on the transition between the 3rd and the 4th block. At this point, the model is processing higher level features and is increasingly focusing on the left side of image where the macula and the fovea are located.

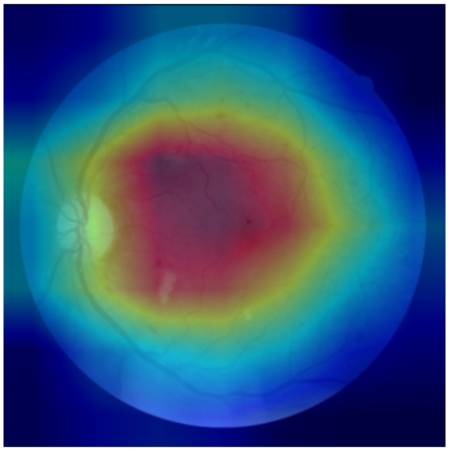

Figure 11. Dense block 4, layer 16

Here, we focus on the last layer of the last block. It is the very last layer of the backbone that passes the fine grained features to the fully connected layers. The parts that the model regards more are the macula and the fovea.

8. Work repartition

During this project, Valentin and Prosper collaborated on alternative fine-tuning, image preprocessing, and ensemble learning. Emanuele handled the remaining parts (image augmentation, attention mechanism and explainability). The interpretation of the results from these different sections was carried out collectively to ensure a comprehensive understanding of what worked well and what did not. The report was written collaboratively, with each person focusing more on their respective parts, as described above.

9. Conclusions

Through this study, we have explored various techniques to improve the performance of deep learning models in predicting diabetic retinopathy from eye images.

We found that data augmentation techniques mitigated overfitting, leading to smoother loss curves and slightly improved kappa scores. On the other hand, incorporating attention mechanisms, particularly Channel and Spatial Attention, allowed the model to focus on more relevant features, increasing the overfitting as a consequence.

Ensemble methods, specifically max voting, showed promising results in improving model performance. However, other ensemble techniques like bagging did not yield the expected benefits, possibly due to implementation issues or suboptimal hyperparameter choices.

Preprocessing techniques, such as Ben Graham's preprocessing and histogram equalization, marginally improved the kappa scores, indicating that better image representation can enhance model performance.

Despite these findings, our study faced several challenges, including limited computational resources and an imbalanced dataset. These limitations constrained our ability to fully optimize the models and explore more extensive hyperparameter tuning. It is possible to see, looking at the first epochs of training, that sometimes the models disregard the class 4 (they get very low precision and recall for that class) as there are too few samples.

Class	Number of sample
0	480
1	320
2	320
3	320
4	160

Table 5. DeepRid dataset

In future work, addressing the imbalance of the dataset, increasing its size and securing more robust computational resources could lead to further improvements in model performance. Additionally, exploring more sophisticated ensemble methods and advanced preprocessing techniques could provide additional insights and enhancements.

Overall, this study provides a comprehensive evaluation of various techniques for improving deep learning models in the prediction of diabetic retinopathy, highlighting the potential and challenges in this domain.

References

[1] K. Boyd. Diabetic retinopathy: Causes, symptoms, treatment. *American Academy of Ophthalmology*, 2024. 1

[2] T. M. E. Center. Diabetic eye disease management. Online. URL https://www.themedicaleyecenter.com/diabetic-eye-disease-management-manchester/. 4

[3] L. Craig. What is fine-tuning in machine learning and ai? Online. URL https://www.techtarget.com/searchenterpriseai/definition/fine-tuning. 1

[4] GeeksforGeeks. Cropping an image in a circular way using python. Online, . URL https://www.geeksforgeeks.org/cropping-an-image-in-a-circular-way-using-python/. 4

[5] GeeksforGeeks. Clahe histogram equalization – opencv. Online, . URL https://www.geeksforgeeks.org/clahe-histogram-eqalization-opencv/. 4

[6] GeeksforGeeks. Ensemble learning. Online, . URL https://www.geeksforgeeks.org/a-comprehensive-guide-to-ensemble-learning/. 3

[7] R. Liu, X. Wang, Q. Wu, L. Dai, X. Fang, T. Yan, J. Son, S. Tang, J. Li, Z. Gao, et al. Deepdrid: Diabetic retinopathy—grading and image quality estimation challenge. *Patterns*, 3(6), 2022. 1

[8] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: visual explanations from deep networks via gradient-based localization. *International journal of computer vision*, 128: 336–359, 2020. 4

[9] C. H. Song, H. J. Han, and Y. Avrithis. All the attention you need: Global-local, spatial-channel attention for image retrieval. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2754–2763, 2022. 3

[10] A. Syed. Enhancing image quality for machine learning: Ben graham's preprocessing. Online. URL https://medium.com/@astronomer.abdurrehman/enhancing-image-quality-for-machine-learning-ben-grahams-preprocessing-e795ad982abe. 4