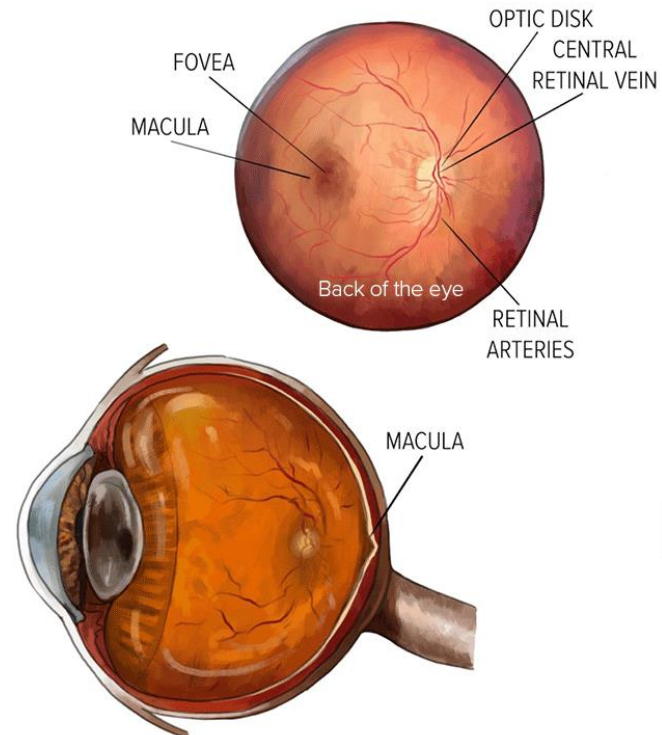


› PROJECT 1: DIABETIC RETINOPATHY CLASSIFICATION

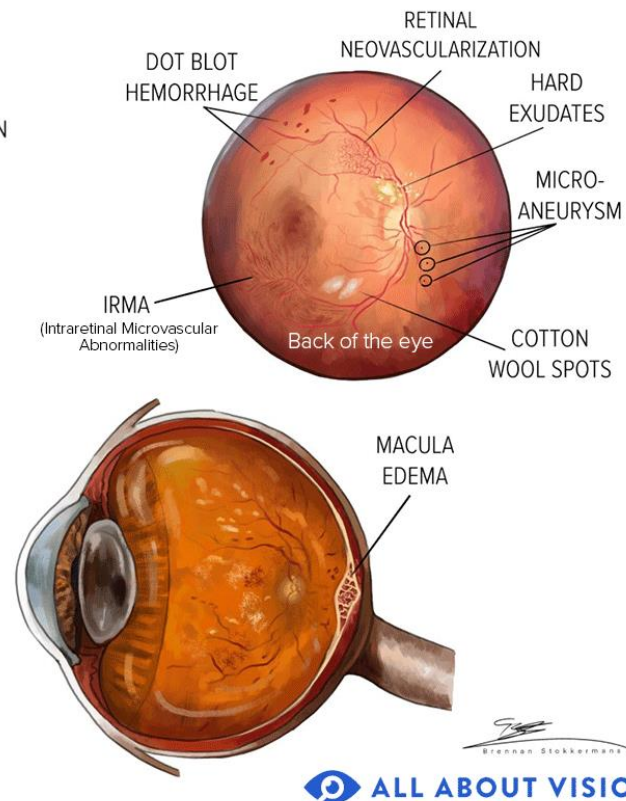
Advanced Approaches for AI-Based Image Processing

DIABETIC RETINOPATHY

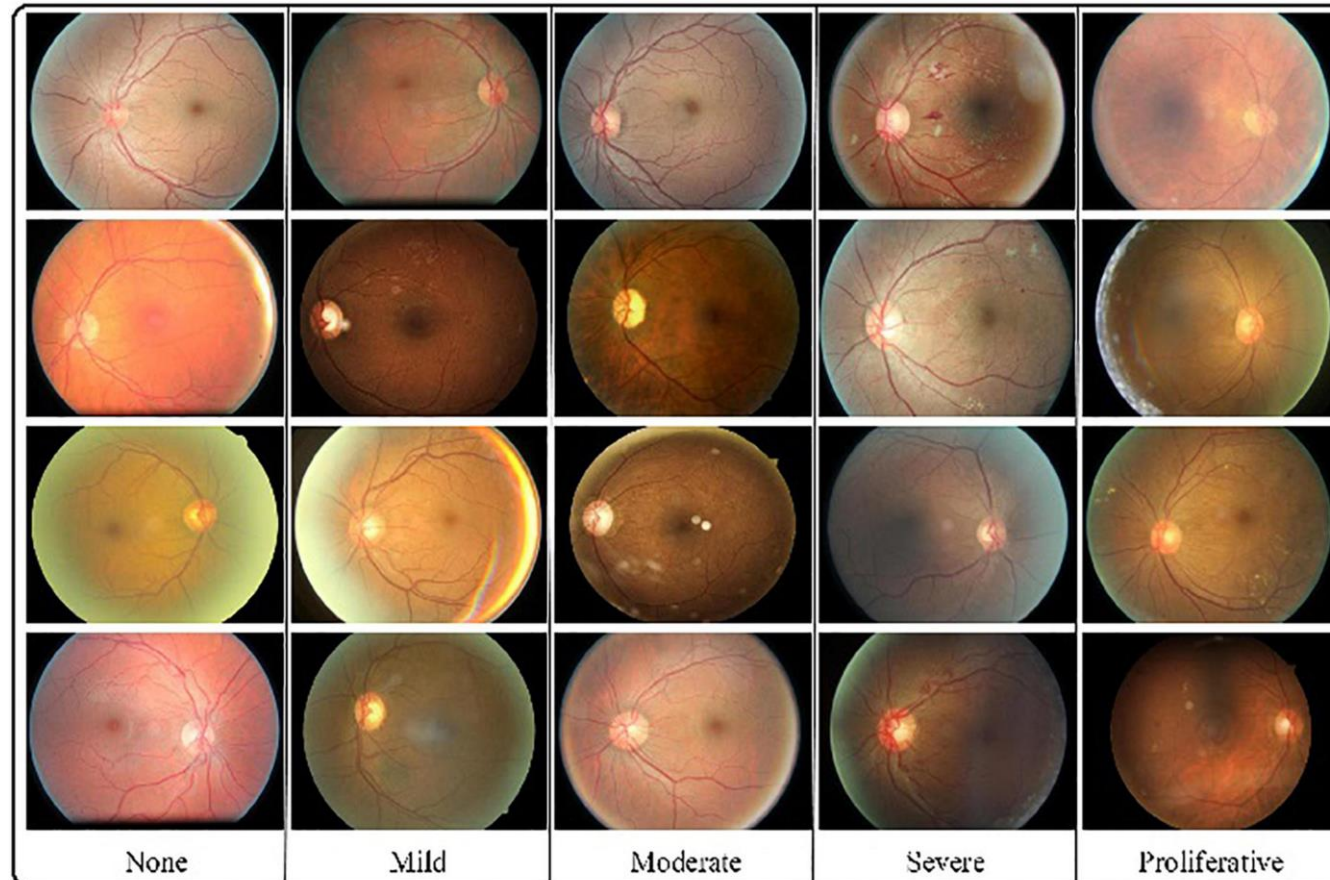
HEALTHY EYE



DIABETIC RETINOPATHY



THE DATA

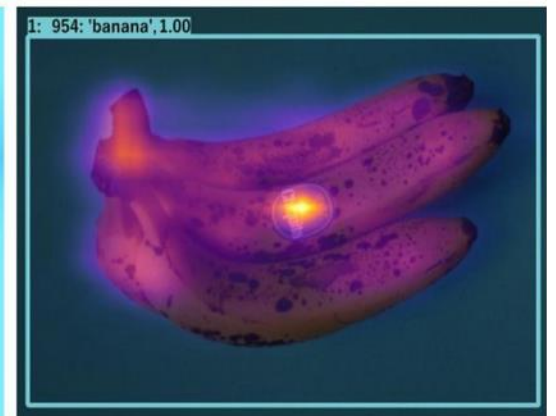


DEAL WITH LITTLE DATA

- Only a few hundred Images available for training
 - Use Data Augmentation, Transfer Learning, Big-Transfer...
-

EXPLAINING AI DECISIONS

- Build trust in your solution by making your model's decisions explainable
- E.g. Attention Maps, Self-Attention



THE TASK

- Train a classifier to distinguish between the different stages of Diabetic Retinopathy
 - You will be equipped with a data set:
<https://bwsyncandshare.kit.edu/s/wkaHiAFaMacMMct> (PSWRD: Diabetic_Retinopathy)
 - You will get access to a GitLab repository
 - **Do not add binaries or image data to the repository**
 - You can use any Python library for this task as well as Open Source code (as long as the license allows it and you cite it!)
-

THE TASK

- Implement your training and evaluation classes and methods in python source code within your repository
 - Additionally, you need to hand in one or max. two **filled-in** Jupyter notebooks to document and demonstrate your results. The notebook must contain
 - The data and model preparation (including **image augmentation**). Train and evaluate at least **two different model architectures, one CNN, one Transformer**, for the task. Motivate your model choice.
 - The training including relevant metrics as losses and accuracies (**also as plot**). Implement at least **one technique to deal with little data**. Motivate your choice. Optimize hyperparameters to get the best out of your training.
 - Evaluate your models on a **separate test set**, including accuracies, a **confusion matrix**, and the average inference time on your platform. Which model works better? Show **five examples of miss-classified images** and discuss possible reasons.
 - Use an **attention map technique** to explain your models decision using five example images. Discuss whether the model focuses on the right features
 - **Keep your trained model locally** in case I need to reproduce your results
 - It doesn't matter how big your model is. It can be interesting to try a smaller architecture and see how well it does
-

THE TASK

- Due date is **17.11.23**
 - You can receive 10 Points for this task
 - 7 points for a **complete and functional** submission
 - 3 points for **presentation, style, and creativity**
-

OUTLOOK: SECOND PROJECT (DETAILS WILL FOLLOW)

- Each group can choose a topic proposed by me
- Or come up with an own request (needs to be approved by my)
- Work on an Idea developed during the Hackathon

<https://mmmmake.com/de/event/heilbronn-future-city/>

- Goal:
 - Provide a coding tutorial on the topic for your peers
 - Present the your tutorial in the course
 - Prepare demonstrate your code live in class
-