Danmarks Tekniske Universitet

Introduction to intelligent systems

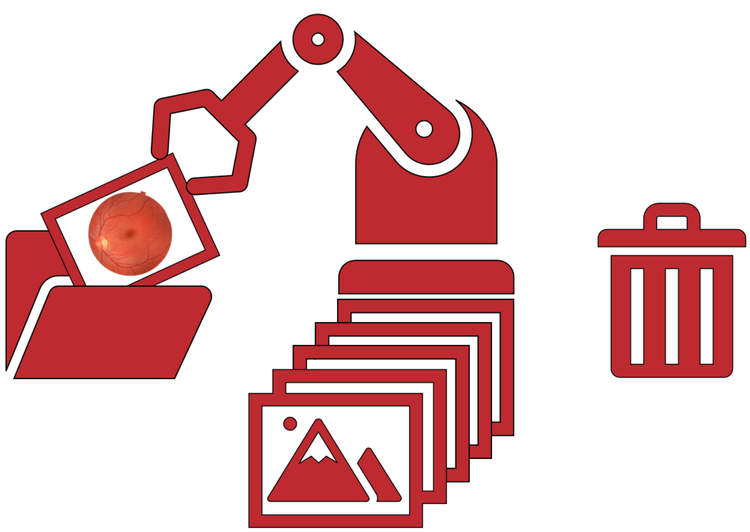
Exam project

Fundus image classification

Katja Eriksen
s214632

Peter Bonvang
s214658

Thomas S. Hegelund
s211899

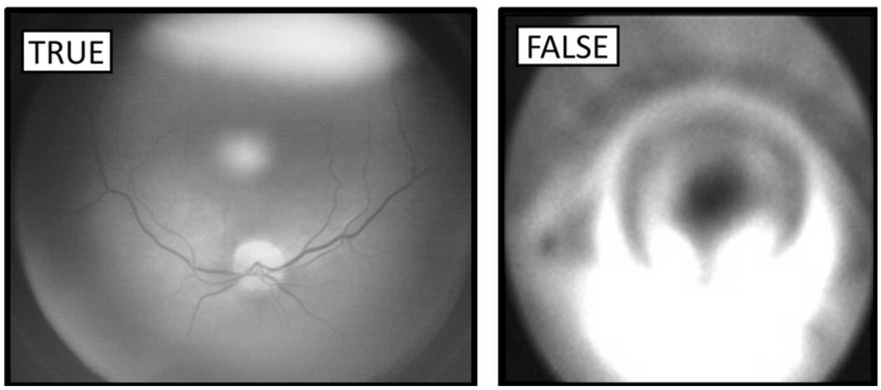


21. januar 2022

Abstract (Joined)

This study was done in collaboration with StatuManu ICP. Fundus imagery and machine learning have in recent years helped diagnose diseases like high intracranial pressure, Glaucoma, and *intraocular* (within the eye) cancer. These methods require clear pictures of the fundus, which are rather cumbersome to obtain manually and StatuManu ICP's current solution, which is based on a pre-trained ResNet model only has an accuracy of 91%. The objective of this study is to construct a custom CNN (convolutional neural network) using enhancement filters that can outperform the company's existing model. Using test accuracy as the benchmark, we compared our custom CNN with six pre-trained models and the existing model. Our custom model did not achieve the highest accuracy, however combining, filters with a pre-trained model (DenseNet201) proved to be more accurate than the existing model, achieving an accuracy of 95.89%. The results show that implementing filters improved the fundus image-classifiers, and using pre-trained models outperforms custom models when the training time is limited.

Introduction (s211899)



Figur 1: Two images from the dataset with labels.
Source: [3] figure 2.A

In recent years new methods of detecting different diseases using pictures of the fundus have been discovered (e.g. diabetes [10] and high intracranial pressure [3]).

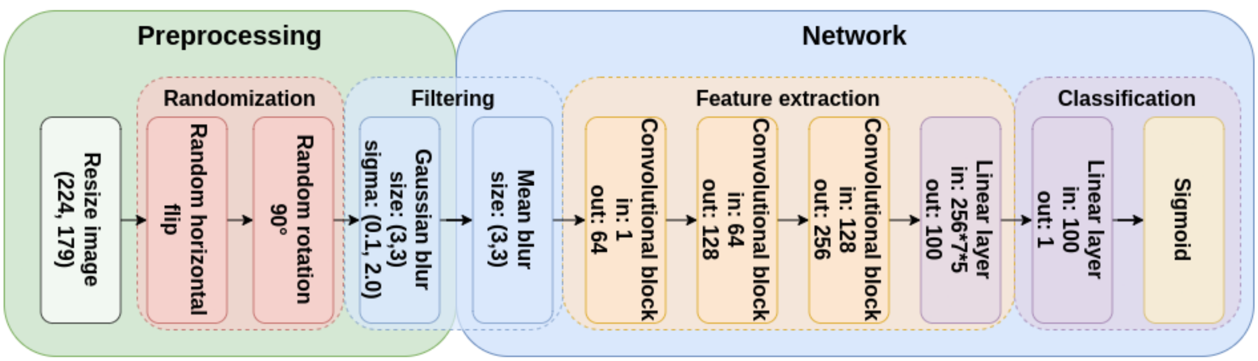
These methods rely on machine learning to identify patterns that otherwise would take an expert to discover. The fundus pictures for these methods are obtained by taking a video of the patient's fundus and then selecting the few frames which are in focus and of the fundus. An example of these types of images with labels is given in fig. 1. The selection of frames can be done by a trained professional, but it is rather cumbersome (and expensive) on a large scale. Companies, like the one this study has been done in collaboration with: (StatuManu ICP), have therefore used machine learning to solve this problem. StatuManu ICP has built a classifier based on a pre-trained ResNet-256 model with a few linear layers, which has yielded an accuracy of 91.00%. ResNet has, however, mainly been built to classify larger objects such as cars, houses, and trees. We, therefore, hypothesize that we can make a custom deep learning model, which will be able to achieve higher accuracy with less post-training than a transfer learning model, using enhancement filters.
Which has been proven on a smaller dataset where the accuracy of 96.44% was achieved by using a deep neural network with three convolutional layers and applying a combination of different filters [7].

Methods (s214658)

As a benchmark to compare pre-trained models against our custom model, we added a classification layer to 5 pre-trained models from PyTorch vision [1]: DenseNet121, DenseNet201, GoogleNet, ResNet101 & ResNet152, and let it finetune the classifier for 50 epochs with a batch-size of 32 on a training-set of 45,000 images, containing 32,876 fundus images and 12,124 non-fundus images. To counterweight the imbalanced datasets the used loss function: BCEWithLogitsLoss was set to multiply the positive class' (in this case, fundus) loss with the ratio between the two classes: $\frac{\#\text{ neg. samples}}{\#\text{ pos. samples}} \approx 0.3688$.
To validate the models, we set aside a validation set of 48,968 images (35,927 fundus / 13,041 non-fundus), which the models were tested on after finetuning.
Based on the trial runs, we expected all the models to be above 80% accuracy and therefore set the proportion to 0.8, giving us a confidence interval of 99% with an error margin below 0.5 %.

The process of determining our contending model was perhaps not the most scientific. It started by gathering information about existing image classification models, like the pre-trained models, and getting an understanding of convolutional neural networks (CNN) [9], residual blocks [6] and dense CNNs [8].
With what we felt was a basic understanding of model architecture and design, we started experimenting. Each of us build a model and trained it on a smaller dataset consisting of 2,257 fundus images and 633 non-fundus images. We then selected the model with the highest accuracy on the validation set, as a base for our custom model.
The best model, with an accuracy of 85.51%, was based on a study [7], which explored the effect of adding different filters to a fundus image-classifier, showing that multiple of the filters had increased the accuracy of the CNN used in the study. Due to a happy accident, we discovered that having Gaussian blur as a pre-processing filter while training and a mean blurring filter embedded as the first layer of the network gave a higher accuracy than each of the filters independently.

Model



Figur 2: The custom model including the pre-processing steps in the training cycle.

To optimize the custom model, we ran 250 hyperparameter trials using Optuna on the following hyperparameters and test values [model]:

- Learning rate: Between 1e-3 and 1e-7
- Optimizer function: Adam, RMSprop, SGD, Adagrad, AdamW, Adamax
- Number of output features in each of the convolutional blocks
  - 1st layer: 8, 16, 32, 64
  - 2nd layer: 16, 32, 64, 128, 256
  - 3rd layer: 32, 64, 128, 256, 512
- Number of output nodes in the first linear layer: Between 80 and 200

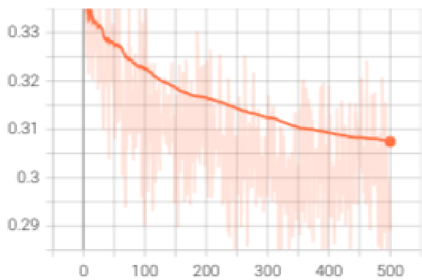
Each trial was run for 50 epochs, or until pruned, on a dataset of 5k images.

Based on the hyperparameter trials, we determined the following values for the hyperparameters:

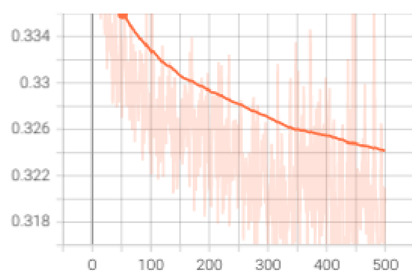
- Learning rate: Should be between 3e-6 and 1.5e-6, and decided to go with 2.2e-6.
- Optimizer function: Should be either Adam or RMSprop, and selected Adam based on recommendations from experts at StatuManu ICP and DTU.
- The convolutional blocks should have the output formation: [16,64] $\rightarrow$ 128 $\rightarrow$ 256, and selected 64 output features for the first layer, 128 for the next and 256 for the last
- The amount of output features in the first linear layer should be between 90 and 110, and selected 100

The model is shown fig. 2.

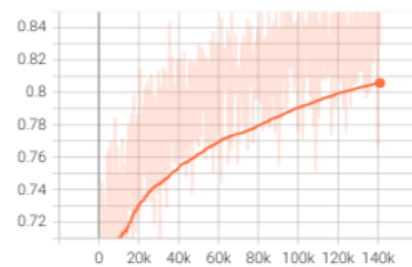
Training



(a) Training loss (per epoch) first 500 epochs

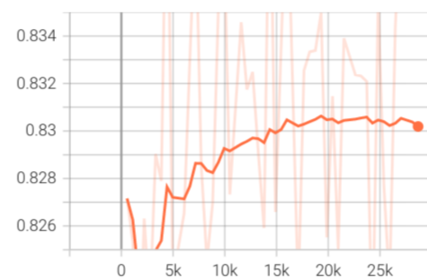


(b) Test loss (per epoch) first 500 epochs



(c) Training accuracy (per step) first 500 epochs

(d) Training loss (per step) last 5 epochs on full dataset



(e) Training accuracy (per step) last 5 epochs on full dataset

Figur 3: Graphs for performance of custom CNN with 99.9% smoothing to show tendency

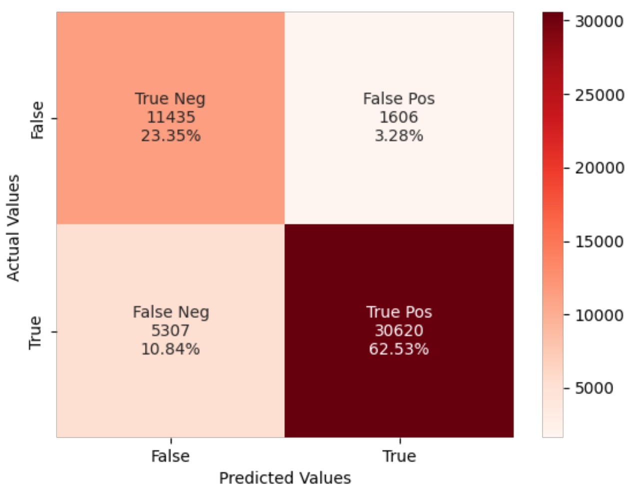
With the model determined and optimized, we set the model to train. First 500 epochs on a 10k dataset getting it to an accuracy of $\approx$ 80%. Then 120 epochs on a 100k dataset which resulted in an accuracy of $\approx$ 82%. To finetune the model it trained 5 epochs on the full training dataset of 196,103 images, achieving the final accuracy of 85.88 %.

Highly accurate model

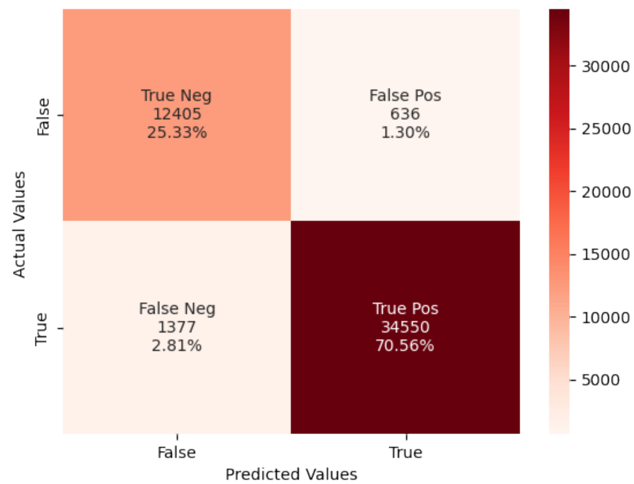
As our objective was to build a model, which could outperform StatuManu ICP's existing model, we decided to test a model consisting of the best baseline model (DenseNet201) with the enhancement (Gaussian blur and mean blur) filters. This model was trained for 25 epochs on the full training dataset and then evaluated on the validation set.

Results (Joined)

The confusion matrices of our costume and best model are given in figure 4.



(a) Costume model



(b) Best model

Figur 4: Confusion matrices from validating our costume and best model. True Neg and True Pos is when the model guesses correct. False Neg and False Pos is when the model guesses wrong.

The accuracy and confidence intervals of each of the models on the validation set of 48,968 images, are given in table 1.

Model	Accuracy	Confidence interval
Dense201 with filters	95.89%	95.66%–96.12%
StatuManu ICP*	91.00%	90.87%–91.13%
Dense201	90.77%	90.43%–91.11%
Dense121	90.52%	90.18%–90.86%
ResNet152	86.84%	86.45%–87.24%
ResNet101	86.54%	86.14%–86.93%
Custom Model	85.88%	85.48%–86.29%
GoogleNet	83.02%	82.58%–83.45%

Tabel 1: Accuracy and 99% confidence intervals.
* Tested on a different dataset

Discussion (s214632 and s211899)

Figure 4 shows that our custom model especially performs worse regarding false negatives. This may, however, be somewhat misleading as over 70% of the pictures are fundus pictures. Table 1 shows that the custom model performs worse than StatuManu ICP's, and this result is significant as evident by the non-overlapping confidence intervals. The only model our custom model ended up outperforming was GoogleNet, and this result was also significant (non-overlapping confidence intervals). The best model was the combination of the Densenet201 and the two filters, with an accuracy of 95.89%. This model also ended up having a higher accuracy than StatuManu ICP's model, which once again is significant due to the non-overlapping confidence intervals.

In general we, therefore, saw an opposite trend to what we expected with the pre-trained models doing better quicker but improving less with more training than our custom model. As evident by figure 3d and 3e our model would not improve from further training, which suggests that it might not be complex enough.

Concerning hyperparameter optimization, we were limited by time and computer power which meant, we were only able to optimize six different hyperparameters (optimizer, learning rate, and features in each layer). Our model could therefore have been improved by optimizing the number of layers and the types of layers used. Yet another limit in our optimization was the depth and the dataset we ran it on. Here we were once again quite limited and we therefore only ran it on a dataset with 5,000 images for 50 epochs.

Even though we did not prove our hypothesis, we did end up improving the existing model by combining the filters of our model with the pre-trained densenet201 model. This combination of transfer learning with custom filters seems to yield significantly better results than either of the two methods would be able to on their own. Further research could be done on this topic.
Using filters to highlight and enhance features and patterns deep neural networks otherwise struggle with recognizing, could also be a topic for further research.

As we did end up making a model which was significantly more accurate than the model currently in use, we hope our research can be used as a building block for the models mentioned in the introduction. This ranges from Glaucoma detected by measuring increased pressure in the eye, diabetic-related disease, cancer in the eye etc. [4]. Diabetic and cancer are serious diseases that affect a large portion of the population. In Denmark alone 280,130 people suffered from diabetes in 2017 [5] and 351,747 people suffered from cancer in 2019 [2]. The main benefit of fundus photographs versus traditional methods for disease detection is the gentle and inexpensive nature of the procedure. Additionally, some health impairing eye diseases can be detected earlier, through fundus photography compared to fluorescein angiography (an eye examination technique). Our neural network contributes to disease detection through fundus photography by being more accurate than previous models as well as less expensive, and faster than the human alternative. Our model takes away one of the most time-consuming tasks by sorting the frames of a video of the fundus. This makes the diagnosticating of diseases and treatment faster [4].

Conclusion (s214632)

We can conclude that despite our custom model getting an accuracy of 85.88% our model is still outperformed by the StatuManu ICP's model (91,00%). However, the accuracy can be improved using a combination of transfer learning and enhancement filters to obtain an accuracy of 95.89%. Our accuracy and loss curves indicate that our custom model would not improve from more training, but maybe more hyperparameter optimization would improve it. The concept of automating the detection of fundus images in videos will help diagnose patients earlier and therefor improve treatment and give a better change of recovery, while saving valuable resources.

References

[1] Pytorch vision models.

[2] Ida Nymand Ammundsen. Antal der lever med kraeft, 2021.

[3] Mikkel Schou Andersen, Christian Bonde Pedersen, and Frantz Rom Poulsen. A new novel method for assessing intracranial pressure using non-invasive fundus images: a pilot study. *Scientific Reports*, 10(1):13062, August 2020.

[4] Dr. Anita Dhanorkar. What is the purpose of fundus photography?, 2020.

[5] Videnscenter for diabetes. Diabetes i tal, 2020.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[7] Asha Gnana Priya Henry and Anitha Jude. Convolutional neural-network-based classification of retinal images with different combinations of filtering techniques. *Open Computer Science*, 11(1):480–490, 2021.

[8] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

[9] Laurence Moroney. Introducing convolutional neural networks (ml zero to hero - part 3), 2019.

[10] Jaskari J. Kivinen J. Sahlsten, J. Deep learning fundus image analysis for diabetic retinopathy and macular edema grading. *nature*, 2019.

Attachments

1. Code used in the study: GitFront