GROUP PROJECT 7

Cataract Classification

a CNN-based approach

TEAM 07

We all worked together as a group, but in particular:

Lorenzo Benatti:
- Exploratory data analysis
- CNN architecture

Davide de Crescenzo:
- CNN architecture
- Preprocessing
- Dataset splitting

Matilde Farci:
- Data augmentation
- Dataset splitting
- Presentation

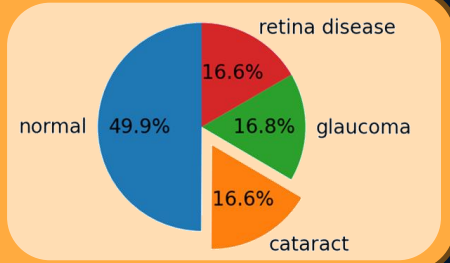
Federico Muraro:
- Preprocessing
- CNN architecture

Davide Sabbadin:
- Data augmentation
- CNN architecture

ORIGINAL PROJECT

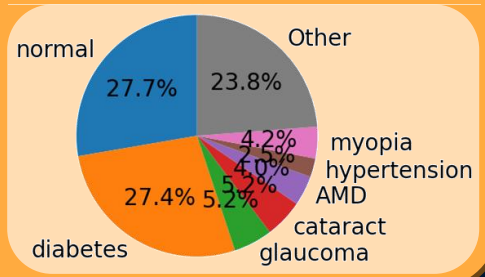
Synthetic overview of data & original task

- Unknown source
- 601 images
- 4 possible diagnoses



Two datasets

- Source: Shandong medical technologies Co.
- 7000 images from 3500 patients (both eyes)
- Text diagnosis



Custom CNN

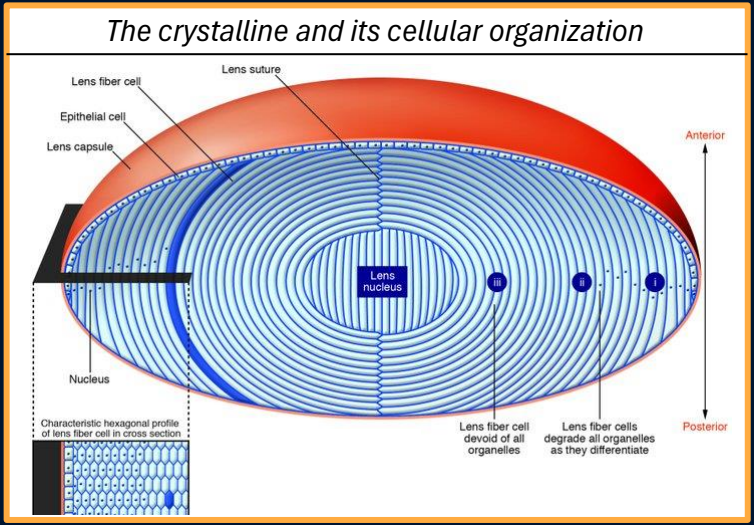
EfficientNet based model

Two models

Cataract classification from eye fundus images

ORIGINAL PROJECT

Eye anatomy & Cataract physiopathology



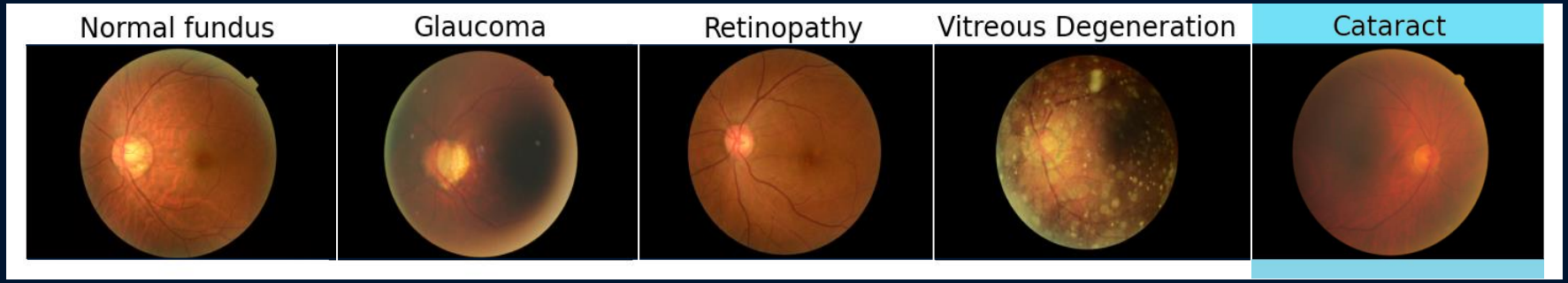
- Crystalline cells do not regenerate
- Cataract is caused by protein clumping between crystalline cells; this may happen in different regions
- Cataract may develop in different crystalline regions

- Aging population → cataract diagnoses are rising
- 1/3 of worldwide blindness is due to cataract
- Limited access to care in developing countries

ORIGINAL PROJECT

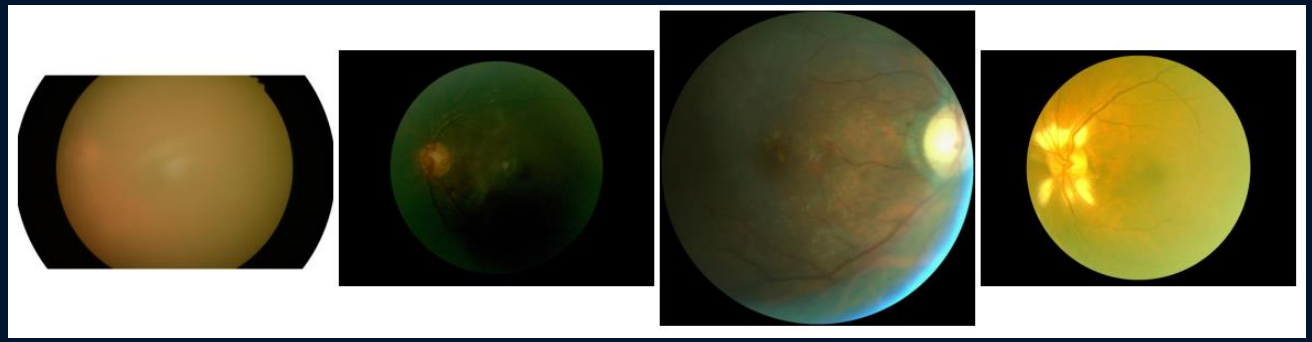
Eye fundus images

Different morphology according to diagnosis: **cataract** images are typically more homogenous



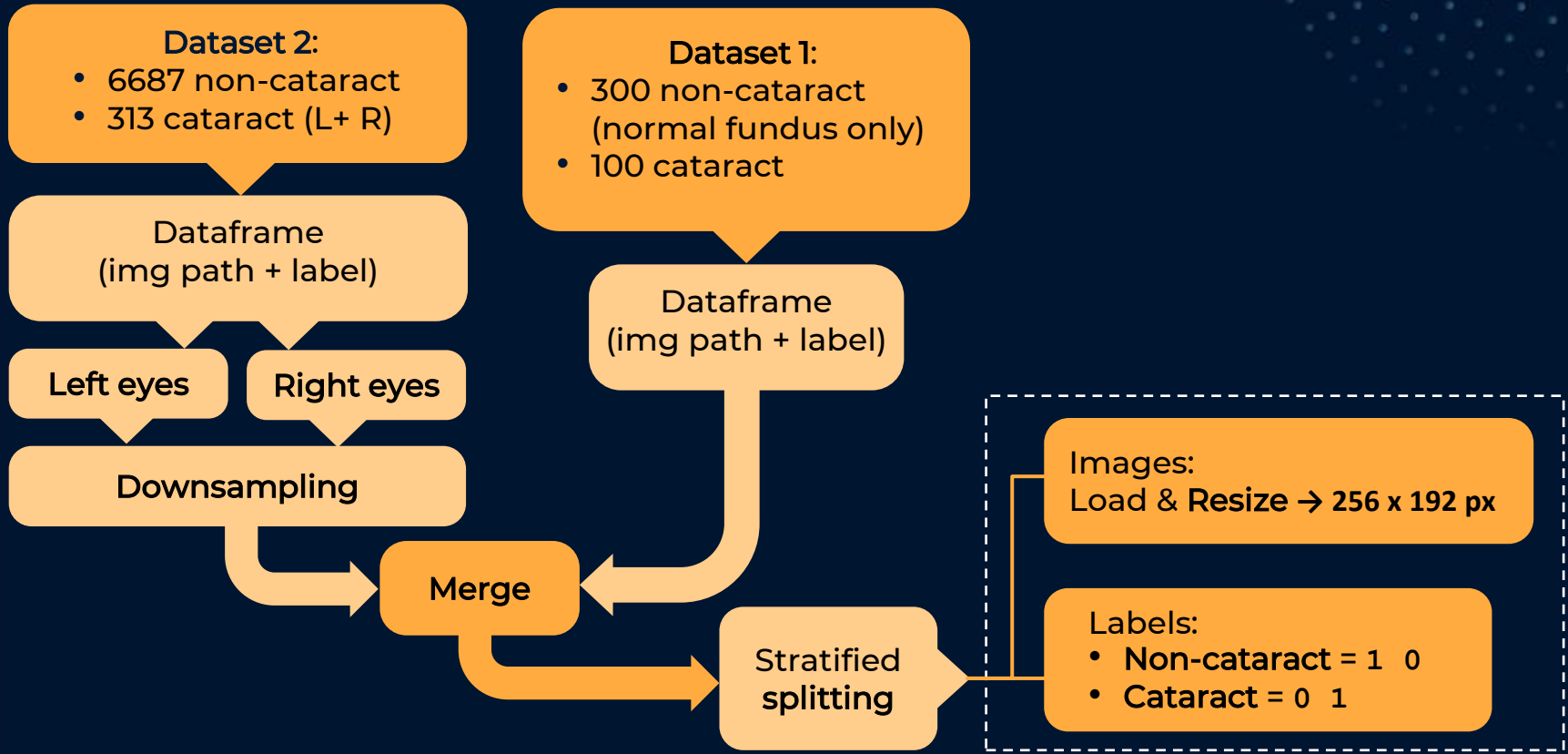
In general:

- **Heterogenous** illumination, size, sharpness...
- **Round-shaped** + black/white **border**



ORIGINAL PROJECT

Code outline
DATASETS CREATION



ORIGINAL PROJECT

Code outline
1° MODEL

Labels

Array:
256 x 192 px
images

CUSTOM CNN

Optimizer: **ADAM**

Loss: **Categorical CrossEntropy**
Metric: **accuracy**

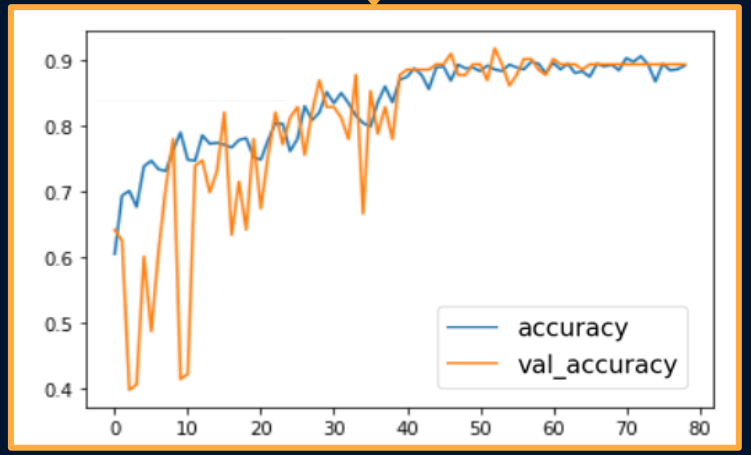
- **Conv2D**(16,3,same,mish)
- **Conv2D**(16,3,same,mish)
- **BatchNormalization**
- **MaxPool2D**(3)
- **Dropout**(0.3)

x 2

- **Flatten**
- **Dense**(2,Softmax)

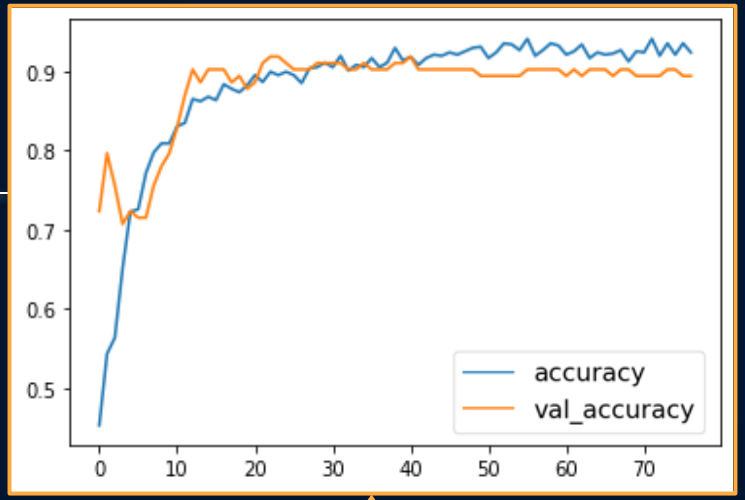
EFFNET-BASED MODEL

keras.fit
- Real time *data augmentation* (**ImageDataGenerator**, shift + flip)
- *Early stopping*
- *Reduce LR on plateau*
- Epochs = 100
- Batch size = 32
- Use of validation set



ORIGINAL PROJECT

Code outline
2° MODEL



CUSTOM CNN

Labels

Array:
256 x 192 px
images

EFFNET-BASED MODEL

Optimizer: **`ADAM`** (LR=0.000003)

Loss: **`Categorical CrossEntropy`**

Metric: **`accuracy`**

- **`EfficientNetB0`**
- **`GlobalAveragePooling2D`**
- **`Dense`**`(2, Softmax)`

`keras.fit`

- Real time *data augmentation* (**`ImageDataGenerator`**, shift + flip)
- *Early stopping*
- *Reduce LR on plateau*
- Epochs = 100
- Batch size = 32
- Use of validation set

ORIGINAL PROJECT

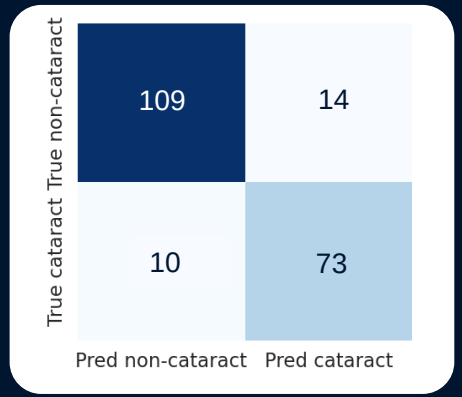
Limitations

- Images from the datasets were **pooled but** classified with different criteria
  → Classification is ***binary*** but ***ambiguous***:
  cataract *vs* non-cataract or cataract *vs* normal?
  - Non-cataract = normal
  - Non-cataract = normal + other diagnoses
- **Dataset 1:** eyes are considered to have **only** one disease **(uncommon)**
- No preprocessing was performed other than resizing
- Models were lengthy to train(>7 seconds/epoch)
- Class imbalance → heavy downsampling → only a subset is used
- Use of deprecated functions (data augmentation with `ImageDataGenerator`)
- Use of images from the same person in both the training and test set
  → Information Leakage
- CNN problems
  - Models not optimized for binary classification
  - No freezing of EfficientNet's weights and wrong input size

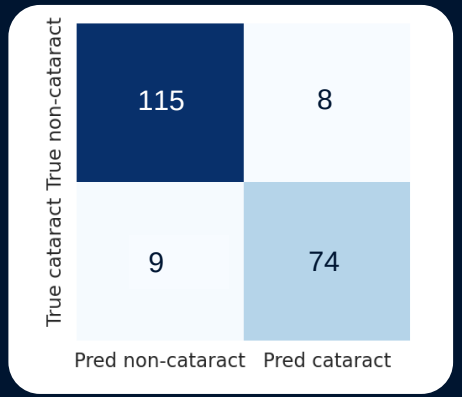
ORIGINAL PROJECT

Key strengths

- 2 datasets
- Good performance overall
- Readable and accessible to non-experts
- 2 models:
  - Custom model that achieves comparable accuracy while having a significantly fewer parameters



  - Transfer learning-based model as baseline



GROUP PROJECT

Objective

- A more general model: since we decided to perform binary classification, we choose to classify cataract *vs.* non-cataract images, rather than **cataract** *vs.* **normal**, considering **ML techniques** perform better in the latter (logistic regression → 95% accuracy)
- Making the most of our datasets: less downsampling & data augmentation
- Speed up the training process: removal of deprecated routines
- Preprocessing to **standardize** images
- Correction of inconsistencies in CNN architecture

How?

- **Scientific literature** (e.g on fundus images classification, binary classification through CNNs...)
- Python **documentation** (Keras especially)
- Laboratory projects & **lectures**

GROUP PROJECT

METHODS
Overview

- Data **leakage correction**
- Exploratory **Data Analysis**
- **Pre-processing**
    - Resizing → Grayscale → CLAHE → Median
- **Models** training and tuning
- Models **evaluation**
    - Additional metrics
    - Threshold adjustment

GROUP PROJECT

METHODS
Literature Review

- Custom CNN architecture focused on cataract classification & grading

  Junayed et al., *CataractNet: An Automated Cataract Detection System Using Deep Learning for Fundus Images.*
  DOI 10.1109/ACCESS.2021.3112938
  Elloumi, *Cataract grading method based on deep convolutional neural networks and stacking ensemble learning*.
  DOI 10.1002/ima.22722

- Use of grayscale images

  Mitra et al., *Enhancement and restoration of non-uniform illuminated Fundus Image of Retina obtained through thin layer of cataract.*
  DOI 10.1016/j.cmpb.2018.01.001

- Use of green channel-only images

  Mitra et al., *Enhancement and restoration of non-uniform illuminated Fundus Image of Retina obtained through thin layer of cataract.*
  DOI 10.1016/j.cmpb.2018.01.001

- CLAHE & filtering

  Al-Fahdawi et al., *Fundus-DeepNet: Multi-label deep learning classification system for enhanced detection of multiple ocular diseases through data fusion of fundus images*
  DOI 10.1016/j.inffus.2023.102059

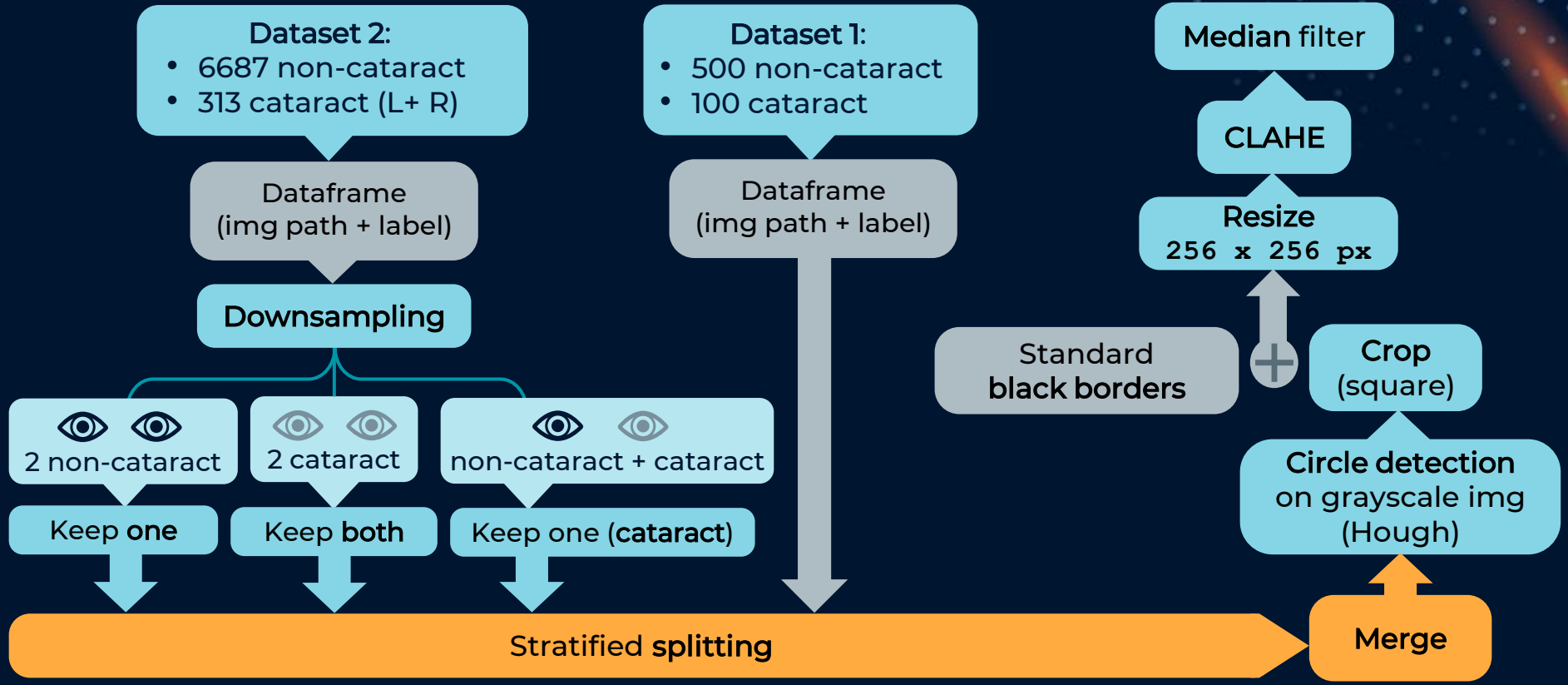
- EFFICIENTNET

  https://keras.io/api/applications/efficientnet/

GROUP PROJECT

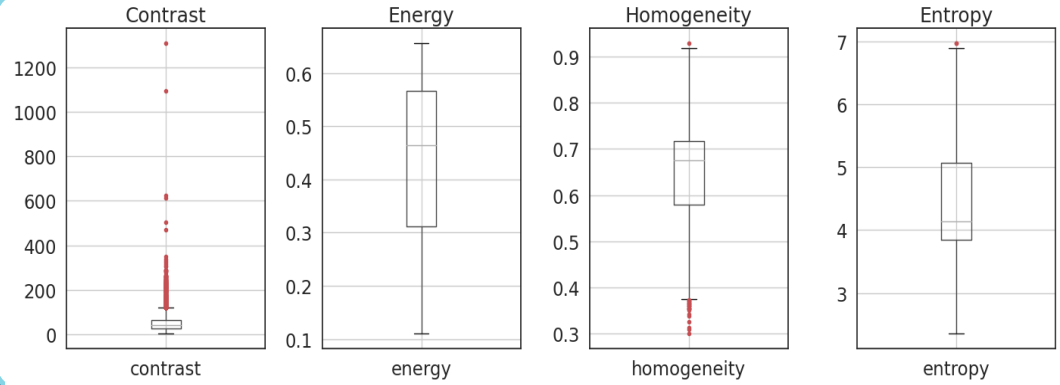
METHODS

Preprocessing

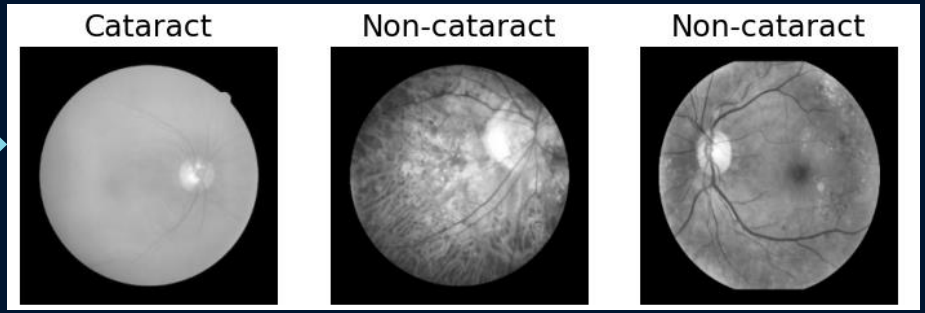


GROUP PROJECT

METHODS
Preprocessing



Resizing → Grayscale → CLAHE → Median



GROUP PROJECT

METHODS
CNN architecture

Labels

Array:
256 x 256 px
images

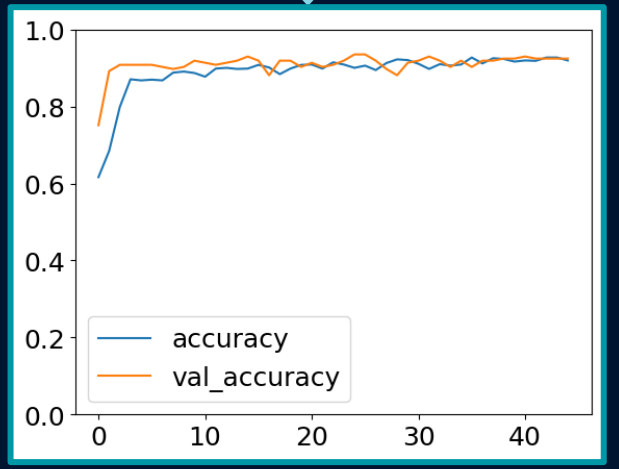
CUSTOM CNN

Optimizer : **ADAM**, *Metric* : **accuracy**
Loss : **Binary CrossEntropy**

- **RandomFlip + RandomZoom + RandomContrast**
- **Conv2d**(16,7,same,relu)
- **MaxPool2D**(3)
- **Conv2d**(16,5,same,relu)
- **MaxPool2D**(3)
- **Conv2d**(32,4,same,relu)
- **MaxPool2D**(3)
- **Conv2d**(32,3,same,relu)
- **MaxPool2D**(3)
- **Dropout**(0.3)→ **Flatten**
- **Dense**(128,relu)
- **Dropout**(0.5)
- **Dense**(1, sigmoid)

keras.fit

- *Class weights* to mitigate class imbalance
- *Early stopping*
- *Reduce LR on plateau*
- Epochs = 100
- Batch size = 32
- Use of validation set



GROUP PROJECT

METHODS
CNN architecture

Labels

Array:
224 x 224 px
images

CUSTOM CNN

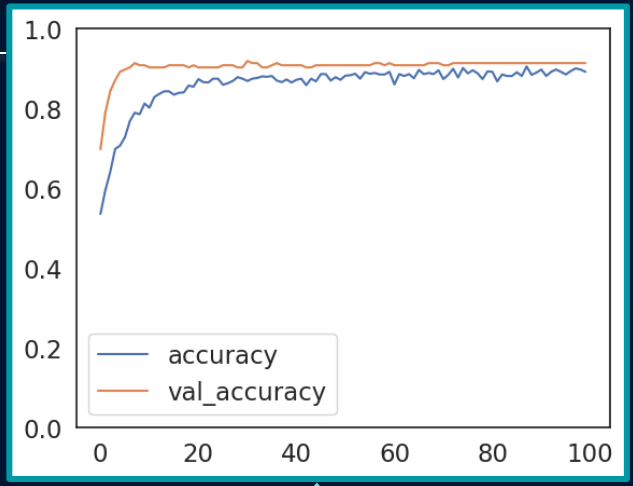
EFFNET-BASED MODEL

Optimizer : **ADAM** (LR=0.000003)

Loss : **Binary CrossEntropy**

Metric : **accuracy**

- **RandomContrast**
- **RandomFlip**
- **RandomZoom**
- **EfficientNetB0** (**freezed**)
- **GlobalAveragePooling2D**
- **Dense** (256, relu)
- **Dropout** (0.5)
- **Dense** (1, sigmoid)



keras.fit

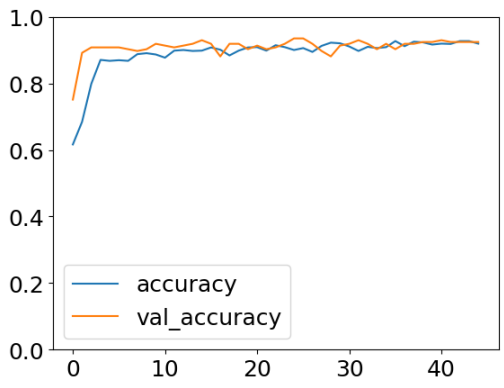
- *Class weights*
- *Early stopping*
- *Reduce LR on plateau*
- Epochs = 100
- Batch size = 32
- Use of validation set

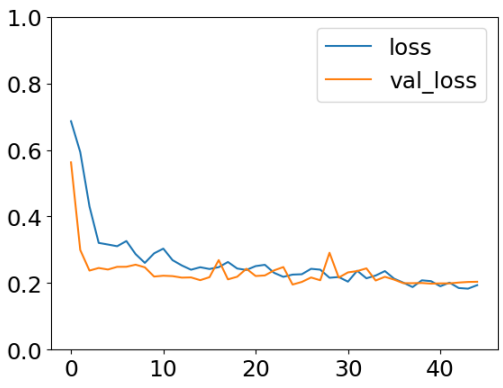
GROUP PROJECT

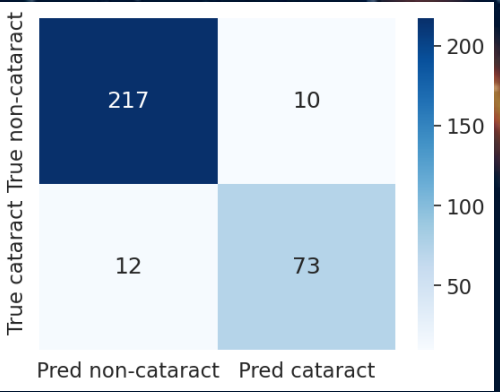
Results

1° MODEL

Accuracy: 0.93
Precision: 0.88
F1 Score: 0.87
Balanced acc.: 0.91
ROC AUC: 0.91
Sensitivity: 0.86
Specificity: 0.96

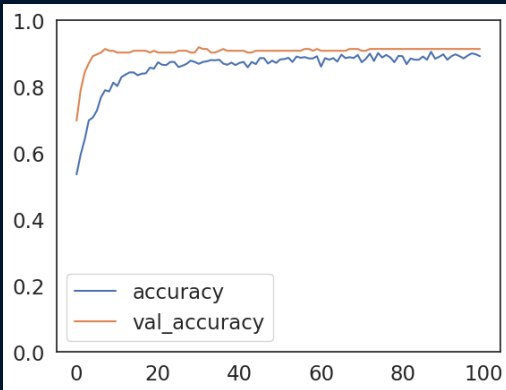


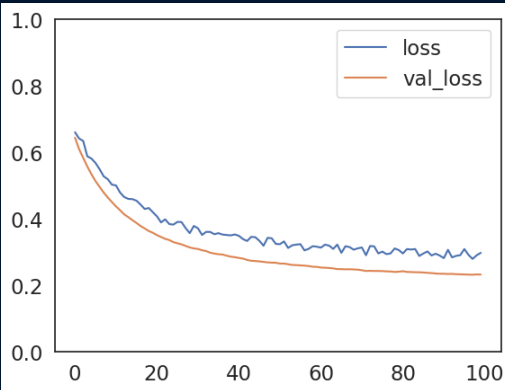


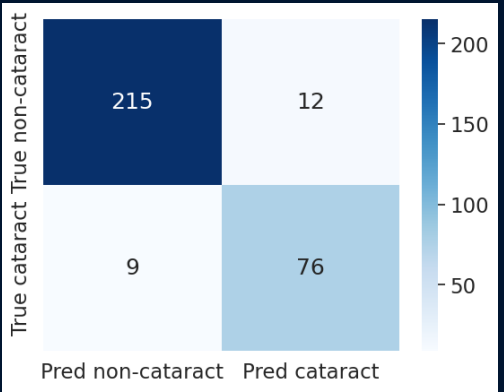


2° MODEL

Accuracy: 0.93
Precision: 0.86
F1 Score: 0.88
Balanced acc: 0.92
ROC AUC: 0.92
Sensitivity: 0.89
Specificity: 0.95



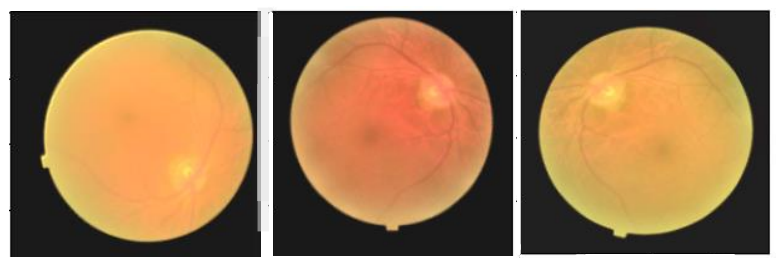




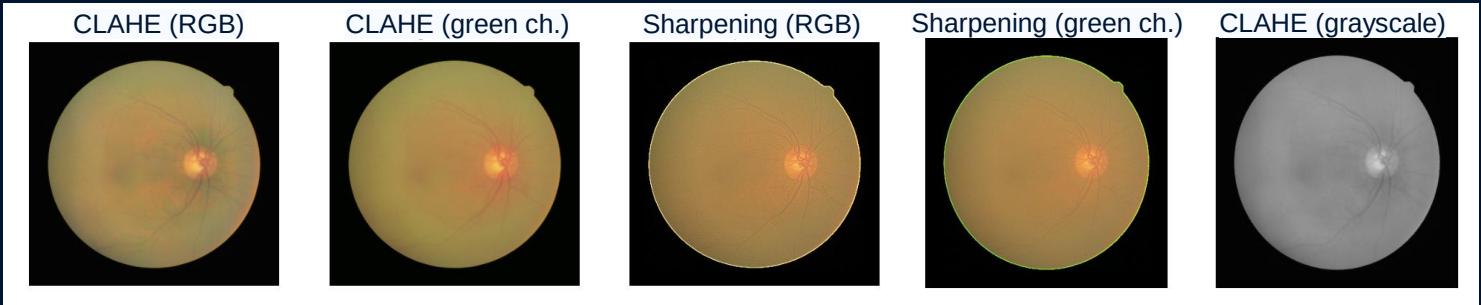
GROUP PROJECT

Discarded approaches

- Data augmentation (on RGB and grayscale images) with `Albumentations`



- Several pre-processing routines



- Several other CNN architectures

GROUP PROJECT

Possible improvements

- Multiclassification (cataract, glaucoma, retinopathy...)

- Further data augmentation on cataract class only → more cataract images
- A more focused data selection