T-DEV-810 Group 12

INTRODUCTION TO MACHINE LEARNING

Valentin NOEL, Jonathan KHALIFA
Fabrice Sumsa, Bastien ANGLES and Kévan SADEGHI

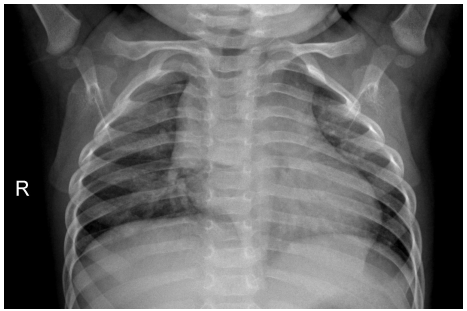
Content

1. Problem to solve
2. The different detection algorithms
3. General principles
4. Data preprocessing
5. Convolutional Neural Network
6. Boosted trees
7. Analysis of Results
8. Conclusion

PROBLEM TO SOLVE

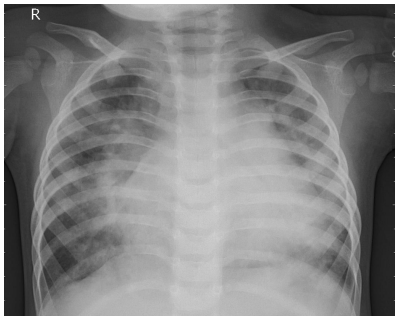
Using a machine learning algorithm, determine from a lung x-ray:

- a **healthy** patient
- a patient with **viral pneumonia**
- a patient with **bacterial pneumonia**



`healthy`

`viral pneumonia`



`bacterial pneumonia`

THE DIFFERENT DETECTION ALGORITHMS

1 - Neural networks

- K-Nearest Neighbor algorithm (KNN)
- Artificial neural network (ANN)
- Convolutional Neural Networks (CNN)

2 - Decision trees

- The Random Forests
- The Boosted Trees

We chose to use the **CNN** and **Boosted Trees** algorithms.

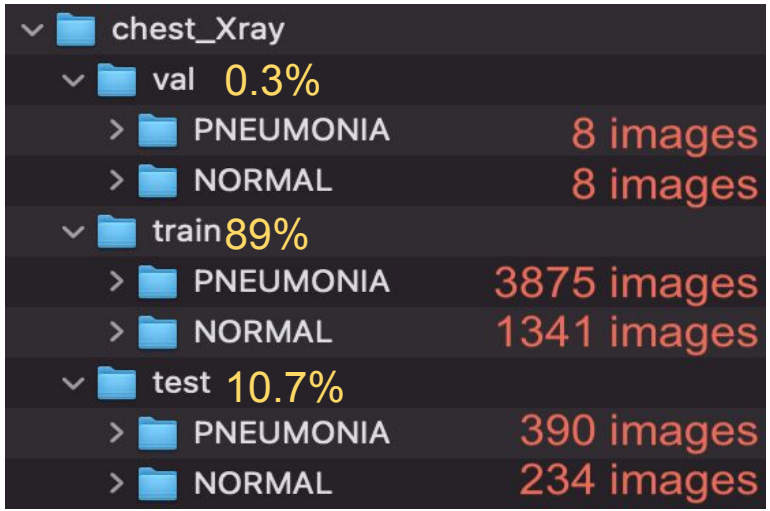
GENERAL PRINCIPLES

Important terms to remember in Machine Learning

- Overfitting
- Underfitting
- Bias
- Variance
- Learning rate
- Gradient descent
- Activation function
- A neuron

DATA PREPROCESSING

Raw Dataset : **5856** images.

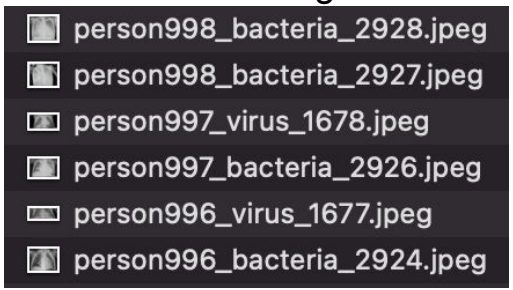


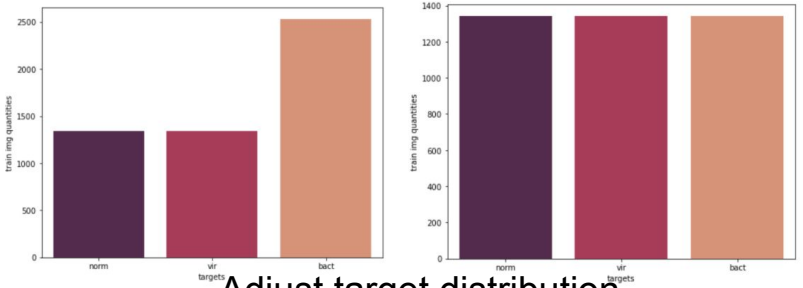
Tasks :

- Check images sizes
- Label the images
- Define the targets
- Check the color channels
- Enhance contrasts
- Adjust class distribution
- Adjust test / train proportions
- Create more images
- Normalize images

DATA PREPROCESSING

Define our targets

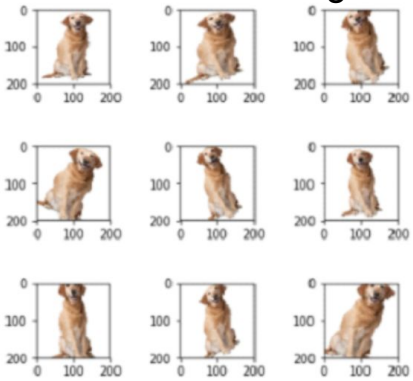




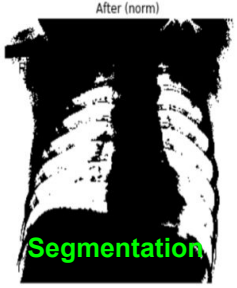
Adjust target distribution

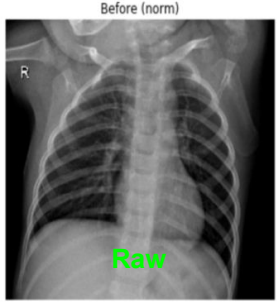
Check images sizes

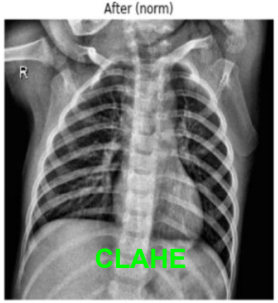
Create more images



Enhance the contrasts

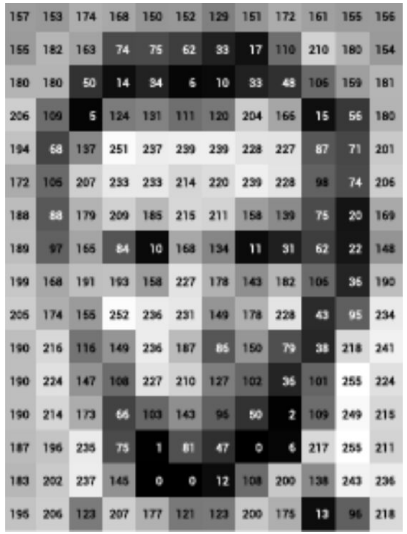




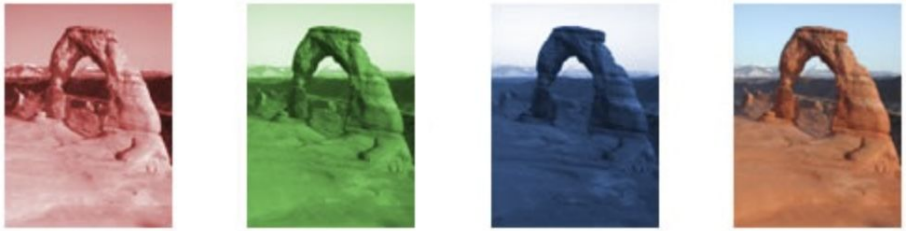


DATA PREPROCESSING

- [1, 0, 0] pour un patient sain
- [0, 1, 0] pour un patient souffrant de pneumonie virale
- [0, 0, 1] pour un patient souffrant de pneumonie bactérienne

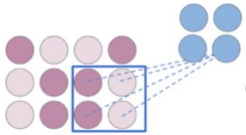


Normalization
Label images > one-hot encoding
Color channels > RGB
Adjust train / test proportions
Shuffle

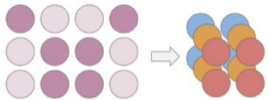


66% train
33% test

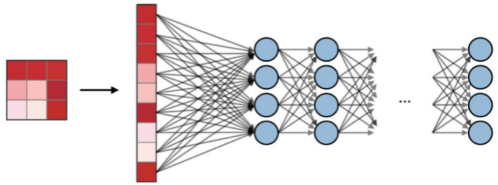
CONVOLUTIONAL NEURAL NETWORK (CNN)

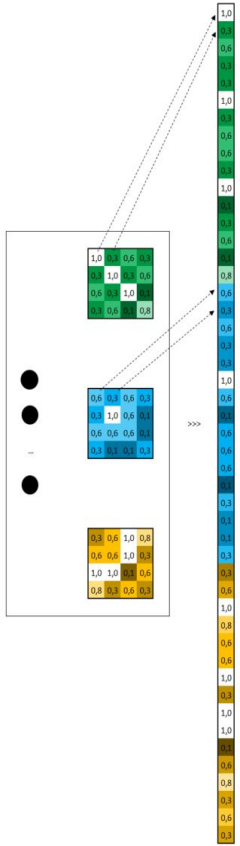


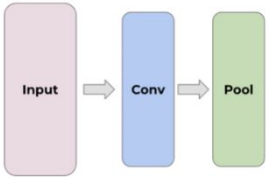
Convolution



Flatten and Dense





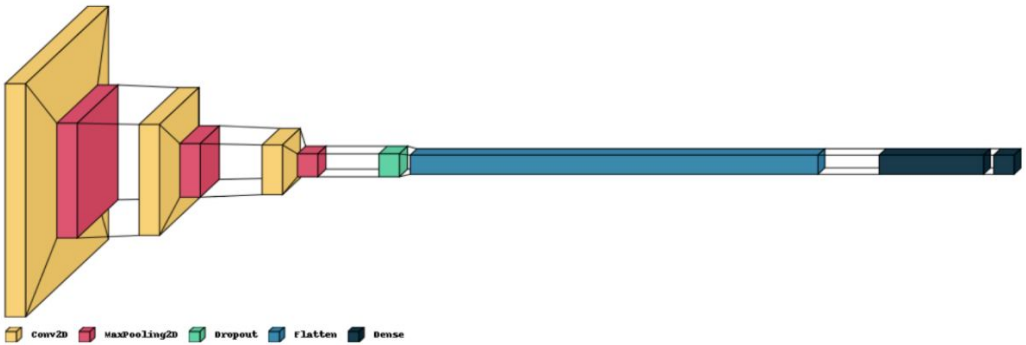


Pairing conv and pooling

- filters
- kernell
- RelU
- stride
- padding
- dropout
- pooling
- softmax
- Adam optimizer

CONVOLUTIONAL NEURAL NETWORK (CNN)

- Nous avons en entrée des images de 64x64x3
- Une première couche de convolution à 32 filtres, et un kernel de 4x4
- Une première couche de pooling avec un kernel de 2x2
- Une seconde couche de convolution à 64 filtres, et un kernel de 2x2
- Une seconde couche de pooling avec un kernel de 2x2
- Une troisième couche de convolution à 128 filtres, et un kernel de 2x2
- Une troisième couche de pooling avec un kernel de 2x2
- Un dropout de 50%
- Un flatten
- Une couche dense de 1024 neurones
- Une couche d'output de 3 neurones



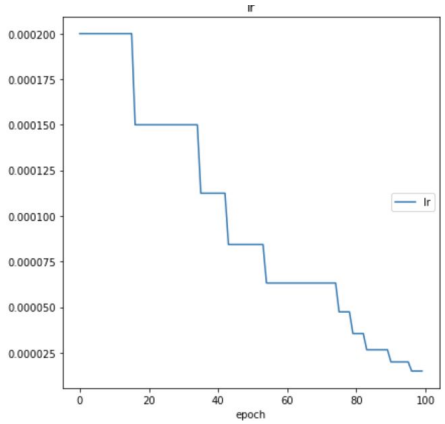
Layer (type)	Output Shape	Param #
conv2d_22 (Conv2D)	(None, 61, 61, 32)	1568
max_pooling2d_22 (MaxPooling	(None, 30, 30, 32)	0
spacing_dummy_layer_30 (Spac	(None, 30, 30, 32)	0
conv2d_23 (Conv2D)	(None, 29, 29, 64)	8256
max_pooling2d_23 (MaxPooling	(None, 14, 14, 64)	0
spacing_dummy_layer_31 (Spac	(None, 14, 14, 64)	0
lastConv (Conv2D)	(None, 13, 13, 128)	32896
1 (MaxPooling2D)	(None, 6, 6, 128)	0
spacing_dummy_layer_32 (Spac	(None, 6, 6, 128)	0
2 (Dropout)	(None, 6, 6, 128)	0
3 (Flatten)	(None, 4608)	0
spacing_dummy_layer_33 (Spac	(None, 4608)	0
4 (Dense)	(None, 1024)	4719616
7 (Dense)	(None, 3)	3075

Total params: 4,765,411
Trainable params: 4,765,411
Non-trainable params: 0

CONVOLUTIONAL NEURAL NETWORK (CNN)

Before training the model : callbacks & hyperparams

Ladder reduce
learning rate
patience : 4
factor : 0.75



Save model's
weights after
each epoch

Epochs : 100
Learning rate : 0.0002
Batch size : 8
Dynamic plotting

K-fold cross validation

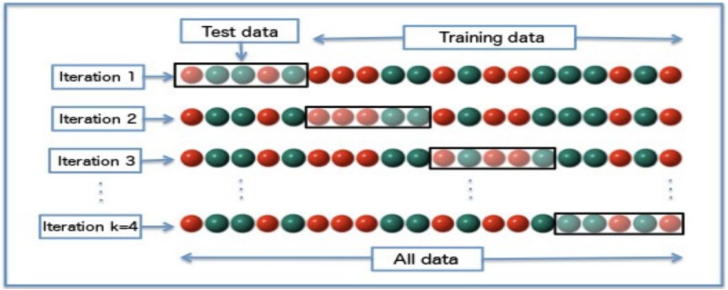
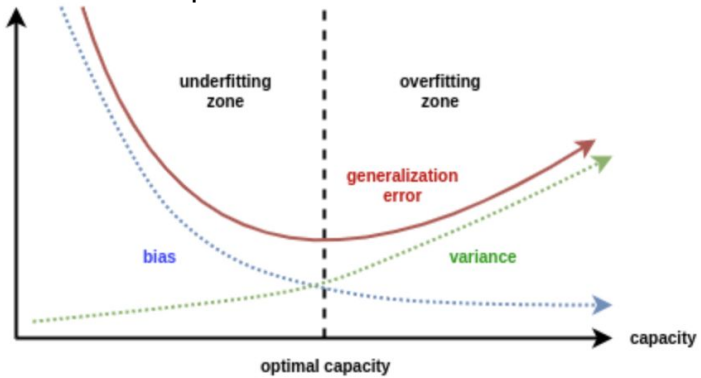
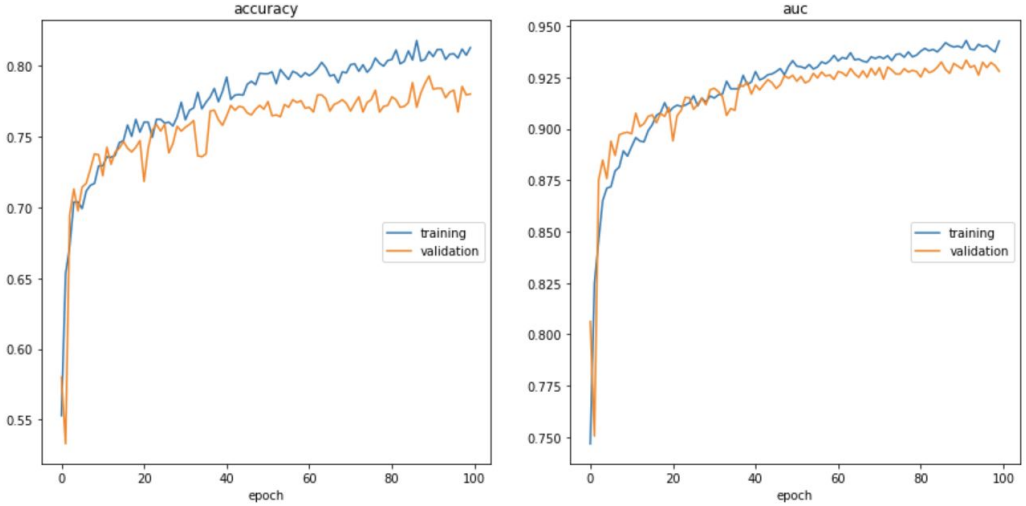


Diagram of k-fold cross-validation with k=4.

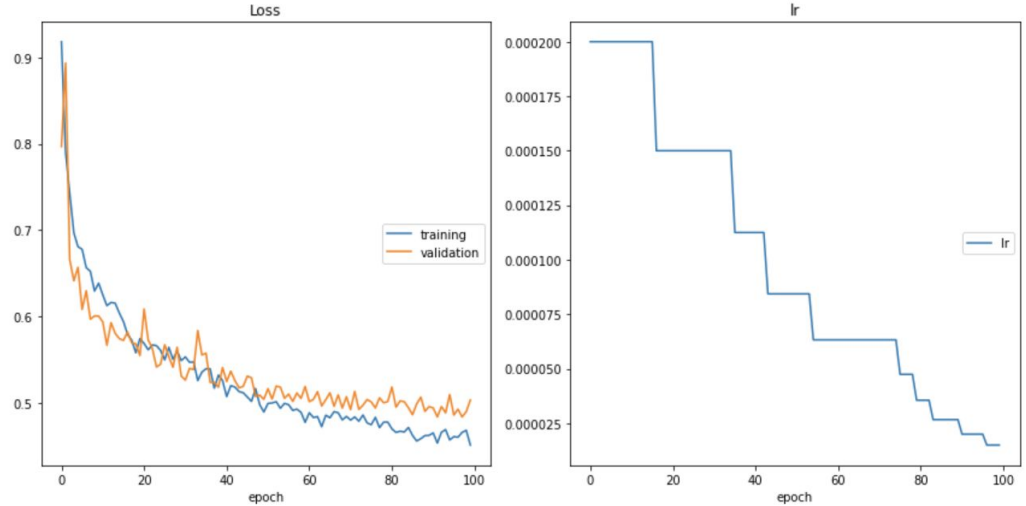
K=3
3 blocs of equal number of images
Equally distributed targets among each bloc
3 distinct trainings

Early stopping
patience : 15




accuracy
training
validation
epoch

auc
training
validation
epoch

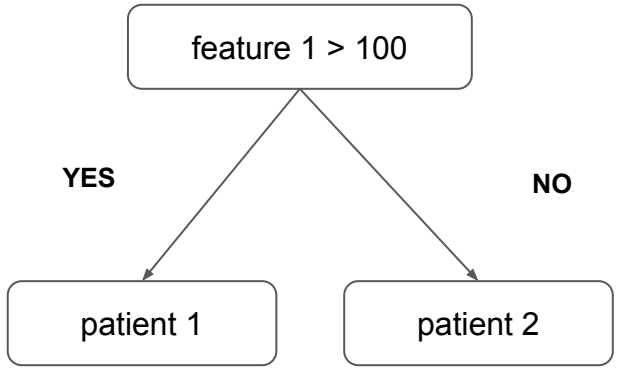

Loss
training
validation
epoch

lr
lr
epoch

BOOSTED TREES

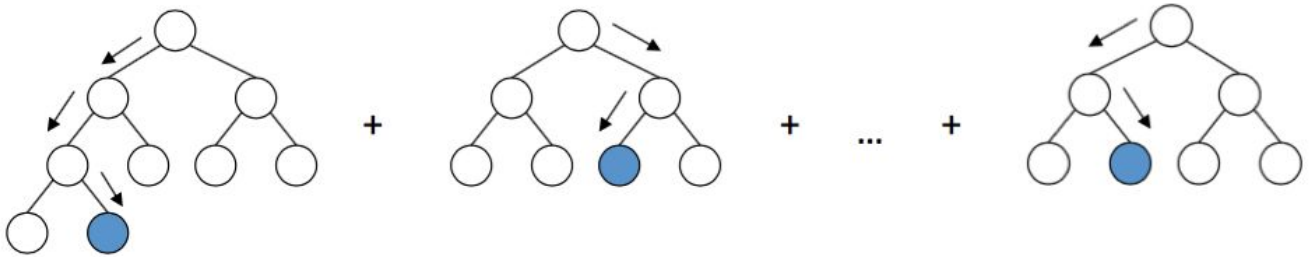
Based on the principle of decision trees. We used the **XGBoost** library.

- Each tree is **created synchronously**, following the others so that the next one learns from the previous one.
- Each internal node represents a test on an attribute.
- Each leaf node represents an item.
- The features to be analyzed are determined by XGBoost.



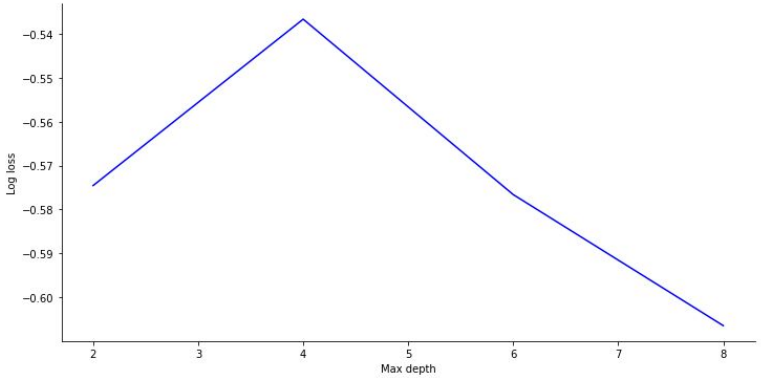
BOOSTED TREES

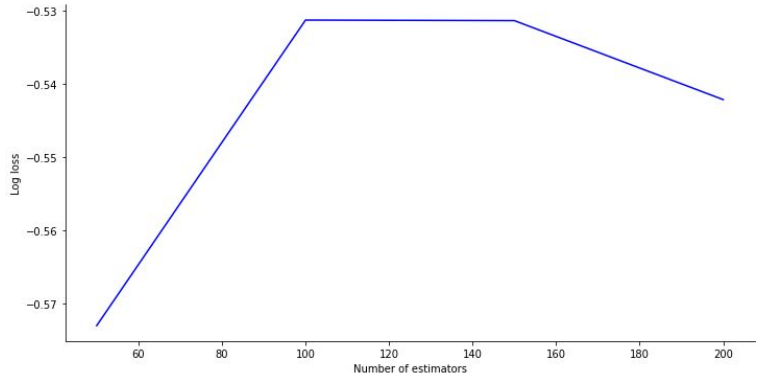
- Each tree gives a probability per patient, which will be adjusted during the training.
- To improve the probability for a patient, we **sum the associated leaf in each tree**.
- The goal is to obtain the smallest residual for each patient.
- In the algorithm, **hyperparameters** are defined: the **number of trees**, the **depth of the trees** and the **learning rate**.



BOOSTED TREES

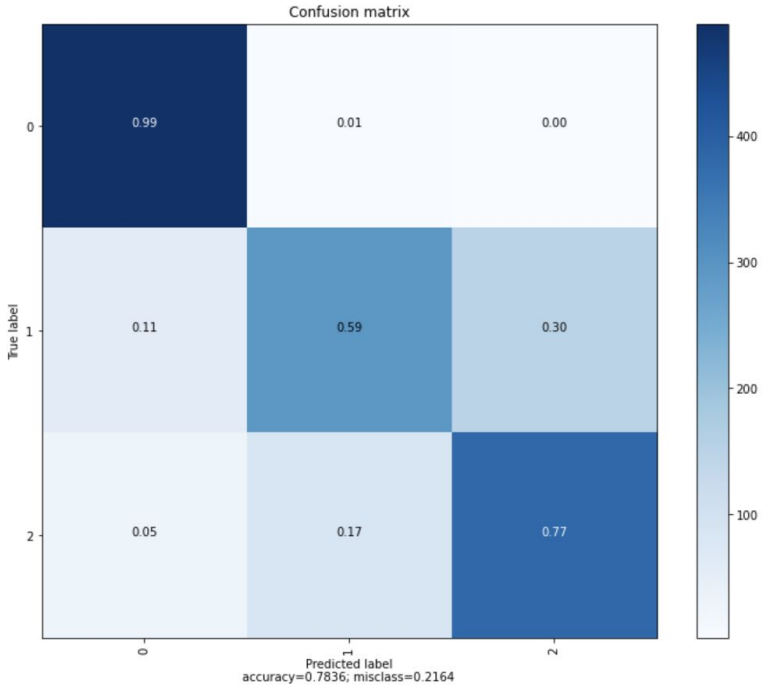
- **GridSearchCV** library is used to find the best hyperparameter settings.
- For each hyperparameters, we **test 4 values**, train the model and select the best one.
- It is preferable to have **not very deep but many trees**.
- Ideally, more values should have been tested and the **hyperparameters should have been combined together**.



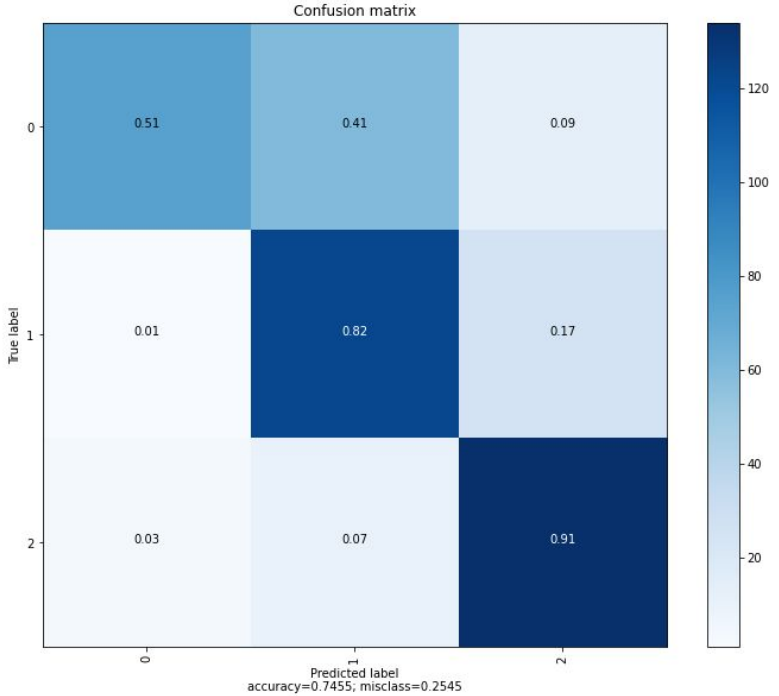


ANALYSIS OF RESULTS

CONVOLUTIONAL NEURAL NETWORK (CNN)

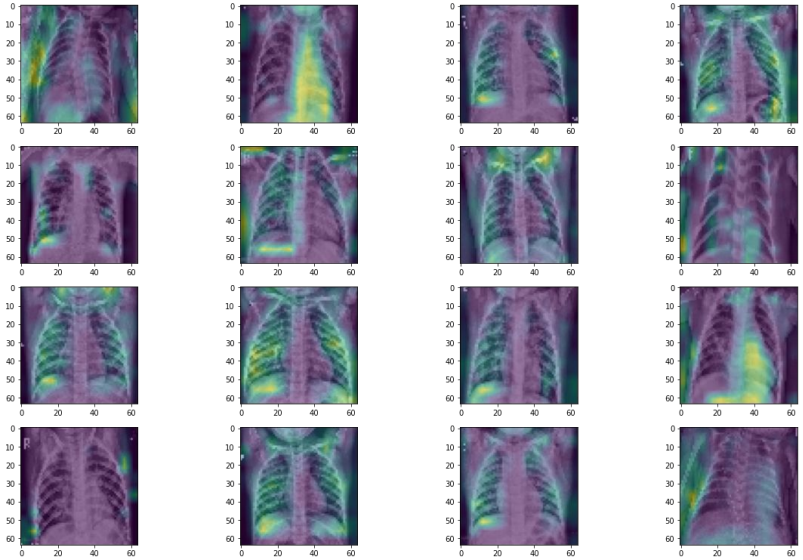


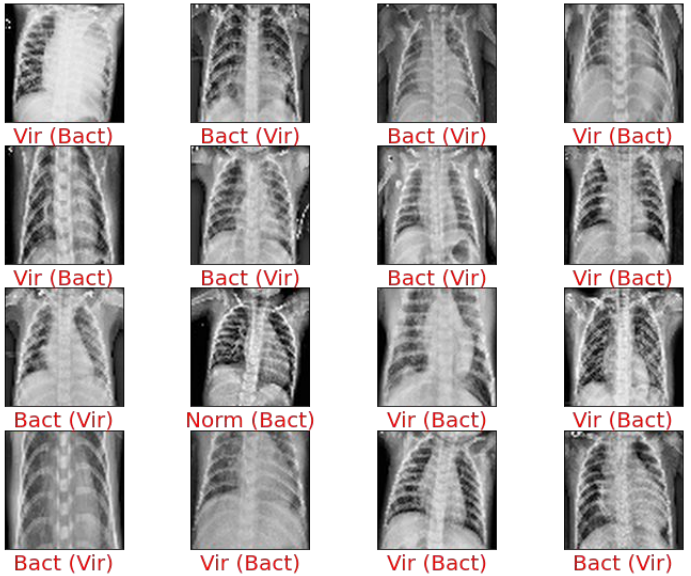
BOOSTED TREES



ANALYSIS OF RESULTS

Details of the CNN’s activity





Discussions

- CNN and Boosted Trees succeed on different points.
- Our low calculation performance may have limited our research.
- Boosted Tree might have corroborated errors along the way.
- Hard to adjust hyperparameters.
- Might use **Ensemble Learning** to perfect our algo.

CONCLUSION

- Different algorithms works totally differently
- Interesting that Decision Tree ≅ Neural Network
- Not a single answer to all the problems
- Realistic problems
- Great introduction to IA

QUESTIONS ?