Artificial Intelligence in Medicine: next steps

Explainability in COVID-19 pneumonia:

Finding consistent features in a multicenter radiomic study and explaining the prediction of a multi-input deep model

Camilla Scapicchio, PhD
INFN Sezione di Pisa

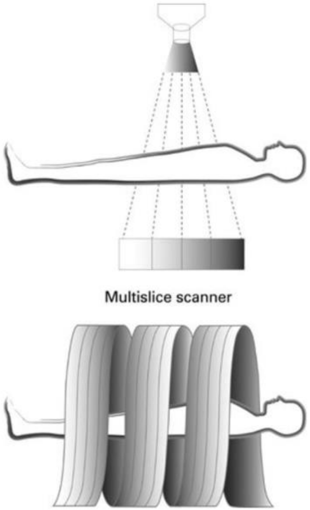
Outline

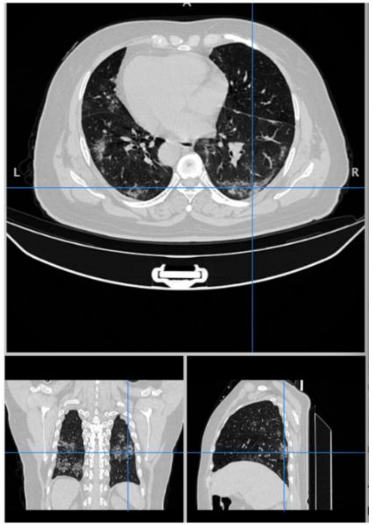
- Covid19-WG
- Explainability in Covid-19 pneumonia
- ML pipeline for severity prediction
  - Dataset and Aim
  - Machine Learning pipeline
  - Finding consistent features
- Multi-input CNN for severity prediction
  - Dataset and Aim
  - Pipeline
  - Explainability
- Conclusions

Covid19-WG [PI,MI,PV,GE,FI,PA,CA]

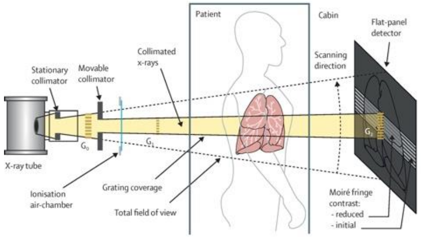
Both qualitative and quantitative characteristics of chest CTs can be used to define the severity of COVID-19 pneumonia

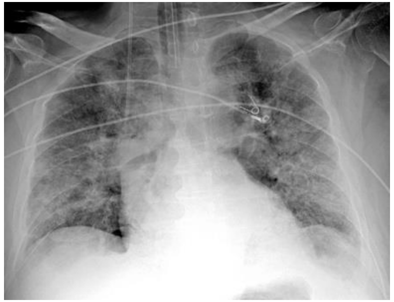
CT images (3D)



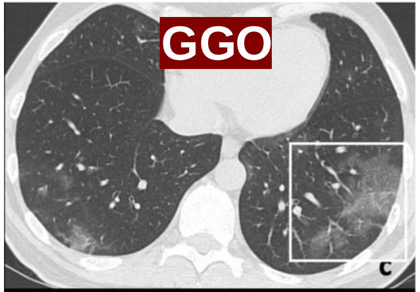


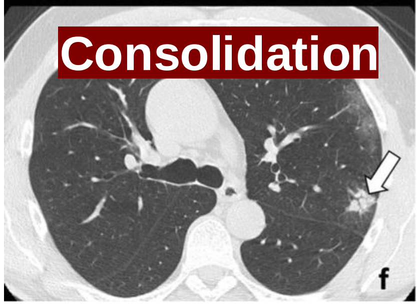
CXR images (2D)





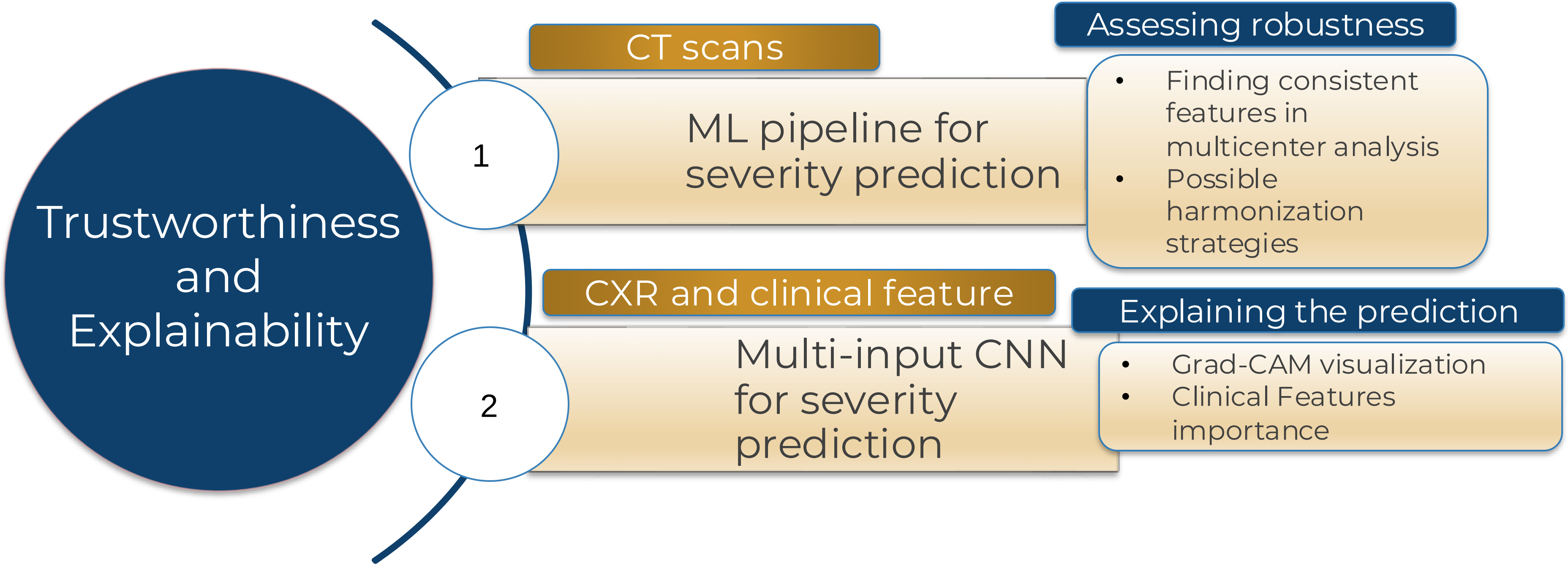
Signs of critical disease:





Explainability in Covid19 pneumonia

1) It is difficult to provide a meaningful explanation for a deep complex model.
2) Issues related to reproducibility and robustness impede the full trustworthiness.



Trustworthiness and Explainability

1 CT scans

ML pipeline for severity prediction

Assessing robustness

- Finding consistent features in multicenter analysis
- Possible harmonization strategies

2 CXR and clinical feature

Multi-input CNN for severity prediction

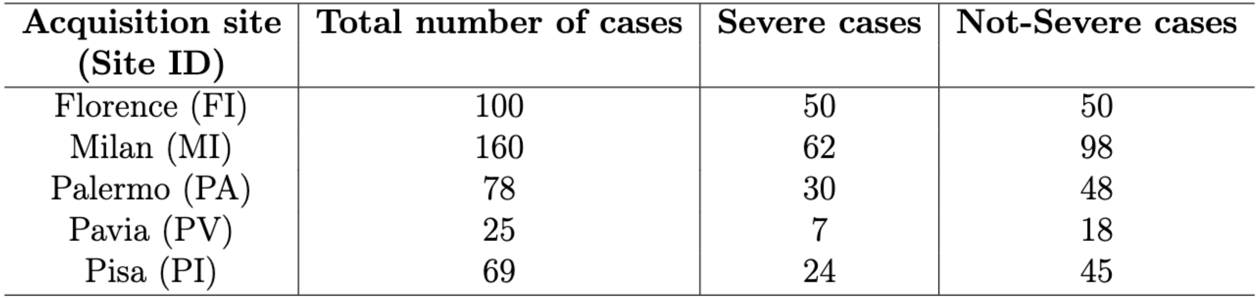
Explaining the prediction

- Grad-CAM visualization
- Clinical Features importance

ML pipeline for severity prediction

Dataset

CT scans with the related clinical outcome on severity (died or intubated) about one month after CT acquisition from 5 clinical centers.



Acquisition site (Site ID)	Total number of cases	Severe cases	Not-Severe cases
Florence (FI)	100	50	50
Milan (MI)	160	62	98
Palermo (PA)	78	30	48
Pavia (PV)	25	7	18
Pisa (PI)	69	24	45

Different vendor machines, acquisition parameters, and reconstruction filters have been used.

Aim

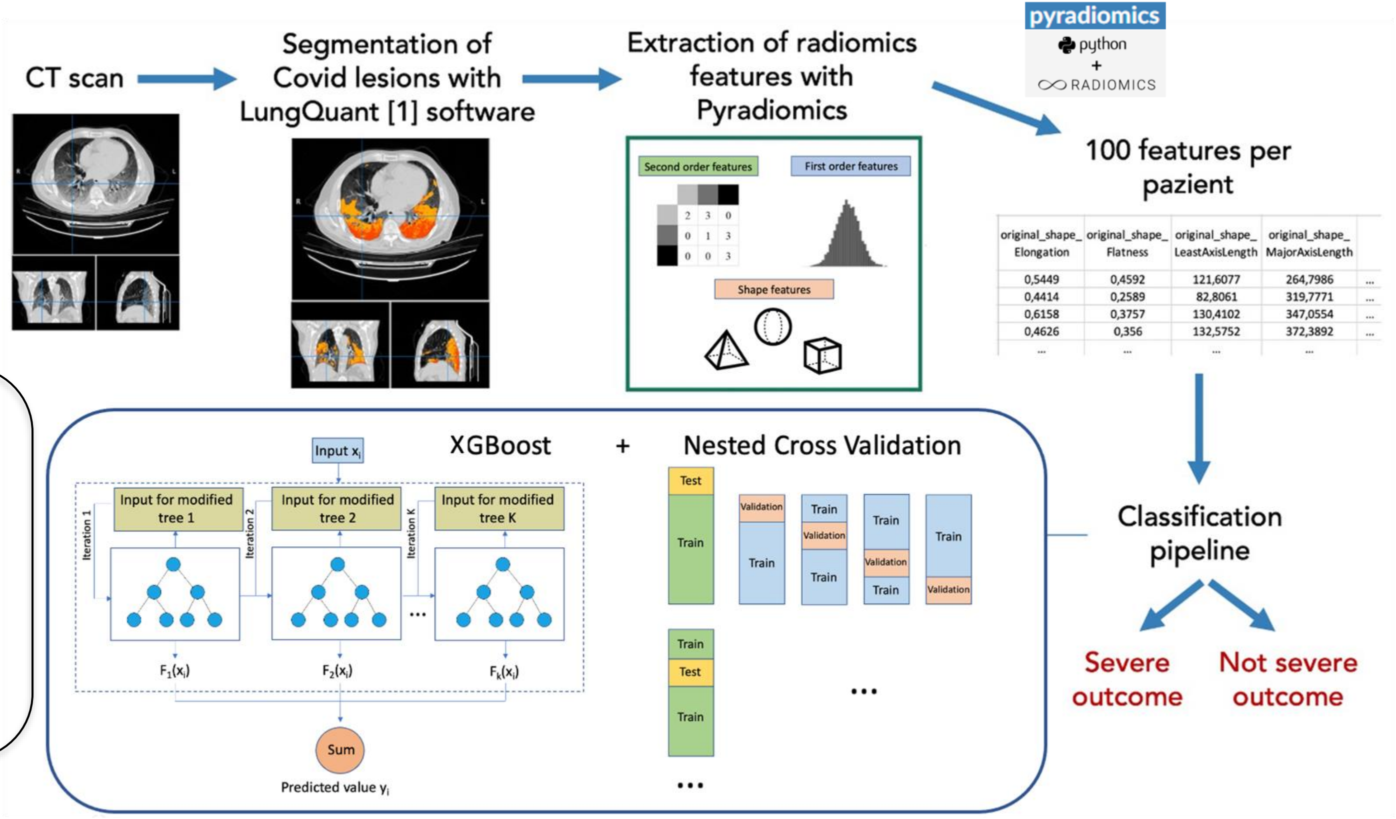
Prediction of the clinical prognosis (severe / not severe) for COVID-19 patients by means of a traditional ML-based classifier trained on **radiomics features** extracted from CT scans.

ML pipeline for severity prediction

Pipeline

10 trials
15% of the dataset in Test
4-fold CV of remaining 85%

gamma: [0.5, 1, 1.5, 2, 5]
max_depth: [3, 4, 5]
subsample: [0.6, 0.8, 1.0]
colsample_bytree: [0.6, 0.8, 1.0]
min_child_weight: [1, 5, 10]
learning_rate=0.02
n_estimators=600
objective='binary:logistic'
nthread=1

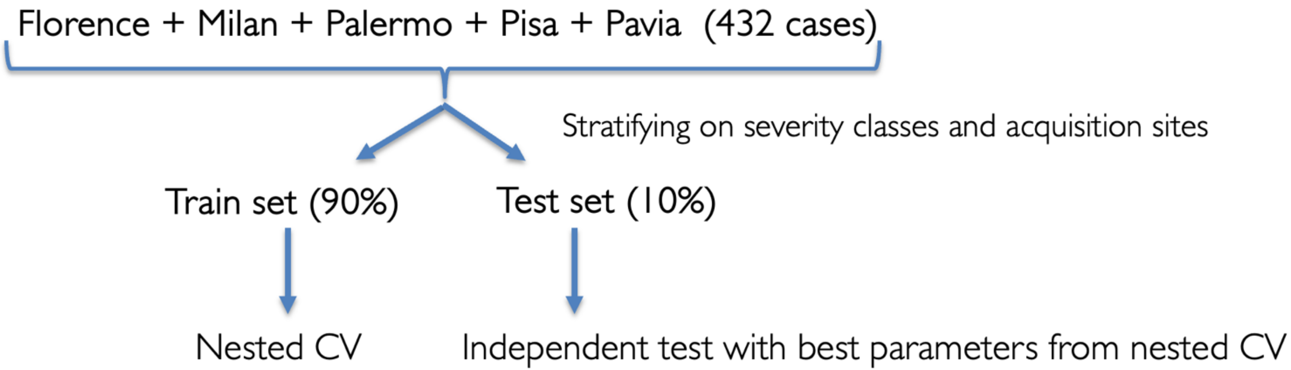


original_shape_ Elongation	original_shape_ Flatness	original_shape_ LeastAxisLength	original_shape_ MajorAxisLength	
0,5449	0,4592	121,6077	264,7986	...
0,4414	0,2589	82,8061	319,7771	...
0,6158	0,3757	130,4102	347,0554	...
0,4626	0,356	132,5752	372,3892	...
...	...	...	...	

[1] Lizzi, F., et al. https://doi.org/10.1140/epjp/s13360-023-03896-4

ML pipeline for severity prediction

First results



Florence + Milan + Palermo + Pisa + Pavia (432 cases)

Stratifying on severity classes and acquisition sites

Train set (90%)

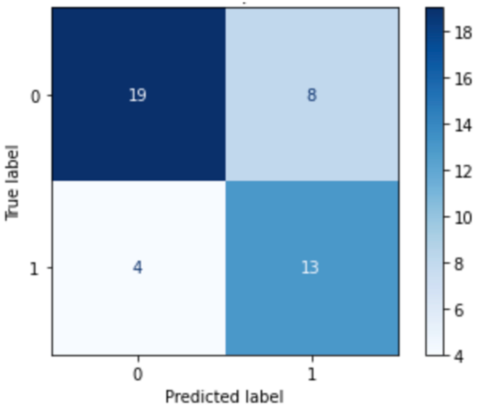
Test set (10%)

Nested CV

Independent test with best parameters from nested CV

roc_auc
0.77 ± 0.06

roc_auc	accuracy	precision	recall
0.80	0.73	0.62	0.76



Finding consistent features

Evaluation of the acquisition site effect

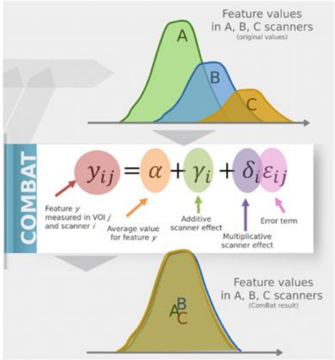
ARTIFICIAL NEURAL NETWORK

roc_auc
0.92 ± 0.04

LOGISTIC REGRESSION

roc_auc
0.87 ± 0.03

ComBat harmonization


Feature values
in A, B, C scanners
(original values)
A
B
C
COMBAT
$y_{ij} = \alpha + \gamma_i + \delta_i \varepsilon_{ij}$
Feature y measured in VOI j and scanner i
Average value for feature y
Additive scanner effect
Multiplicative scanner effect
Error term
Feature values
in A, B, C scanners
(ComBat result)
A
B
C

roc_auc
0.64 ± 0.08

roc_auc
0.52 ± 0.06

Severity prediction

Florence + Milan + Palermo + Pisa + Pavia (432 cases)

roc_auc	accuracy	precision	recall
0.85	0.80	0.70	0.82

Analysis of consistent features

most relevant features selected on different datasets and with different selection methods

- mRMR
- Feature Importance

The same Non-Uniformity features are significant before and after data harmonization!

gldm DependenceNonUniformity
gldm GrayLevelNonUniformity
glrlm GrayLevelNonUniformity
glrlm RunLenghtNonUniformity
glszm GrayLevelNonUniformity

Multi-input CNN for severity prediction

Dataset

Severity prediction dataset:

Training set:

1103 subjects -> Chest X-Ray + clinical parameters such as age, sex, presence of cough, difficulty in breathing… with missing data.

Test set:

486 subjects with both CXR images and clinical data taken from one single hospital not included in the training set.

Lung Segmentation dataset:

2 datasets collected for tuberculosis:

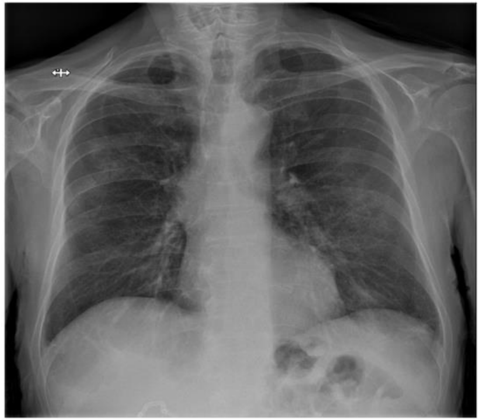
Shenzhen (340 normal and 275 abnormal images)

Montgomery (80 normal and 58 abnormal images)

Aim A fully automated algorithm for the prediction of COVID-19 patient severity outcomes based on Chest X-Ray images and clinical data.

AI4COVID Hackathon (https://ai4covid-hackathon.it/)

CRX image



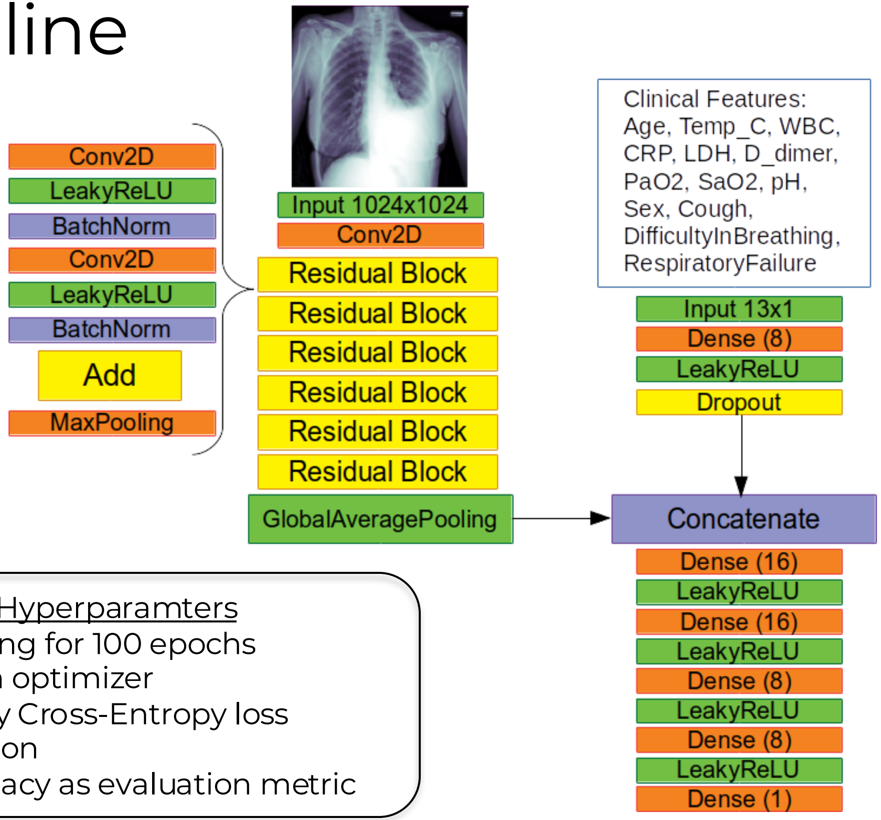
+

Clinical features

- Age
- Sex
- Body Temperature (°C)
- Cough
- Dyspnea
- WBC
- CRP
- Fibrinogen
- LDH
- D-dimer
- O2
- PaO2
- SaO2
- pH
- Cardiovascular Disease
- Respiratory Failure

Multi-input CNN for severity prediction

Pipeline



Conv2D
LeakyReLU
BatchNorm
Conv2D
LeakyReLU
BatchNorm
Add
MaxPooling

Input 1024x1024
Conv2D
Residual Block
Residual Block
Residual Block
Residual Block
Residual Block
Residual Block
GlobalAveragePooling

Clinical Features:
Age, Temp_C, WBC,
CRP, LDH, D_dimer,
PaO2, SaO2, pH,
Sex, Cough,
DifficultyInBreathing,
RespiratoryFailure

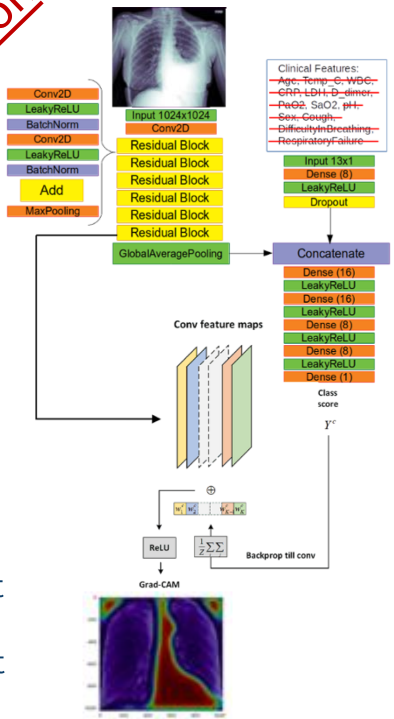
Input 13x1
Dense (8)
LeakyReLU
Dropout

Concatenate
Dense (16)
LeakyReLU
Dense (16)
LeakyReLU
Dense (8)
LeakyReLU
Dense (8)
LeakyReLU
Dense (1)

<u>Hyperparamters</u>

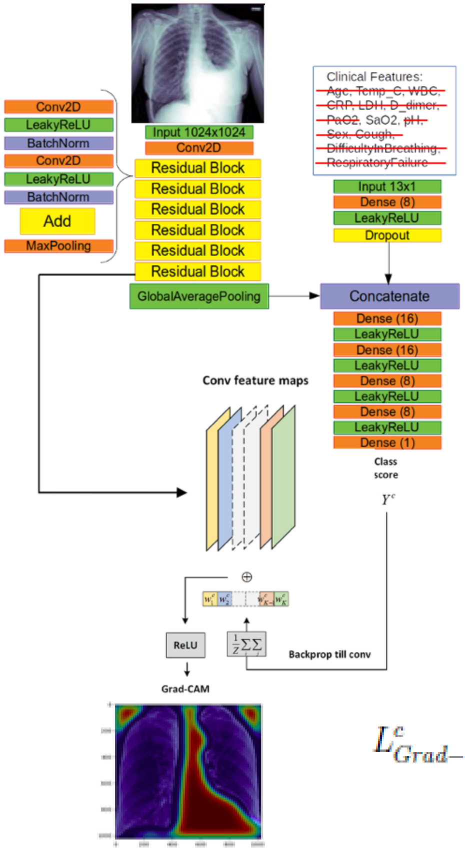
- Training for 100 epochs
- Adam optimizer
- Binary Cross-Entropy loss function
- Accuracy as evaluation metric

Explaining the prediction



The AIM-WG team achieved the **4th place** with a **74%** accuracy in predicting patient prognosis on the independent test set

Explainability



Clinical features branch

Features that can better predict the outcome:

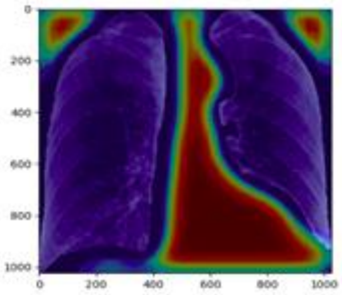
- *PaO2*

  (Partial pressure of oxygen in arterial blood)
- *SaO2*

  (arterial oxygen saturation)

By using only these features the performance is the same as obtained by using all the features.

Images branch



The explanation visualization enabled the detection of the bias and the optimization of the entire pipeline suggesting the addition of a segmentation step.

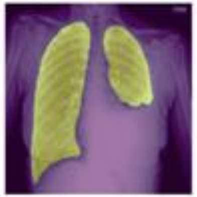
$$L^c_{Grad-CAM} = ReLU\left(\sum_k \alpha_k^c A^k\right) \quad \alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A^k_{ij}}$$

Pipeline optimization

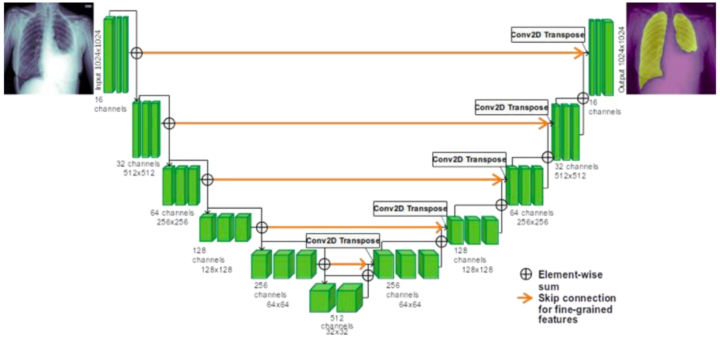
Correct gray level

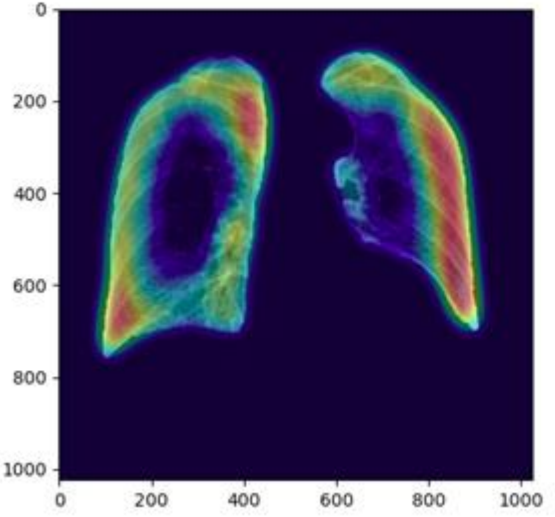
→ 2D U-net →

CXR lung mask



Clinical Features:
Age, Temp_C, WBC, CRP, LDH, D_dimer, PaO2, SaO2, pH, Sex, Cough, DifficultyInBreathing, RespiratoryFailure





Severity Net Model
(Multi-input CNN)

↓

Severe / Mild

This helped the network to concentrate on the right area of the image.

Lung Segmentation

Test set: 10% of Shenzen and Montgomery dataset

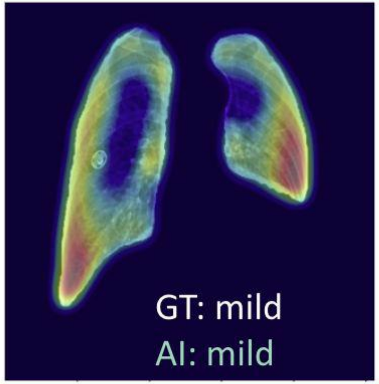
Dice Similarity Coefficient (DSC): 0.96 ± 0.03

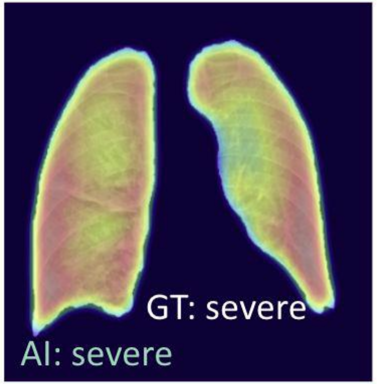
Outcome prediction

Accuracy	**76%**
Sensitivity	**77%**
Specificity	**76%**
AUC	**84%**

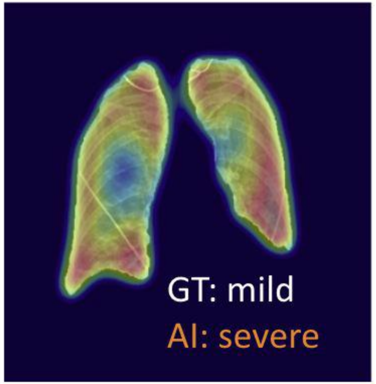
Explainability

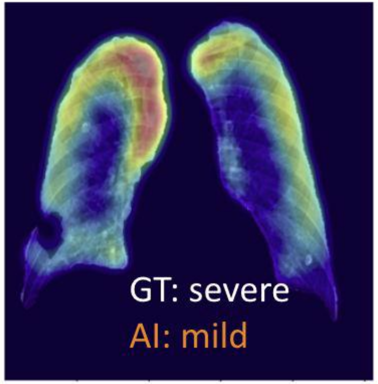
Correctly classified by the multi-input CNN





Misclassified by the multi-input CNN





Conclusions

An ML pipeline trained on a multi-center dataset of radiomics features can predict the severity outcome for COVID-19 patients.

- Some features mainly describing spatial inhomogeneity of the lesion appear to be robust with respect to the site and method of selection, and could therefore be specific to the pathology.

Scapicchio, C., et al. »COVID-19 severity prediction based on radiomic features extracted from lung ct scans using the Lungquant segmentation software." Physica Medica: European Journal of Medical Physics 115 (2023)

The multi-input CNN is able to predict the COVID-19 severity prognosis starting from both CXR images and related clinical variables.

- The explanation is in line with the clinical routine.

Lizzi, F. et al. (2024). A Multi-input Deep Learning Model to Classify COVID-19 Pneumonia Severity from Imaging and Clinical Data. https://doi.org/10.1007/978-3-031-64636-2_18

contact: camilla.scapicchio@pi.infn.it

Thank you for your attention!

INFN
Istituto Nazionale di Fisica Nucleare

Università di Pisa

Acknowledgments

Covid-19 WG people

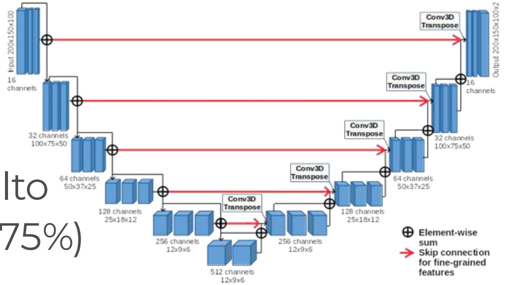
Radiologists and Medical Physicists collaborating with the nextAIM project.

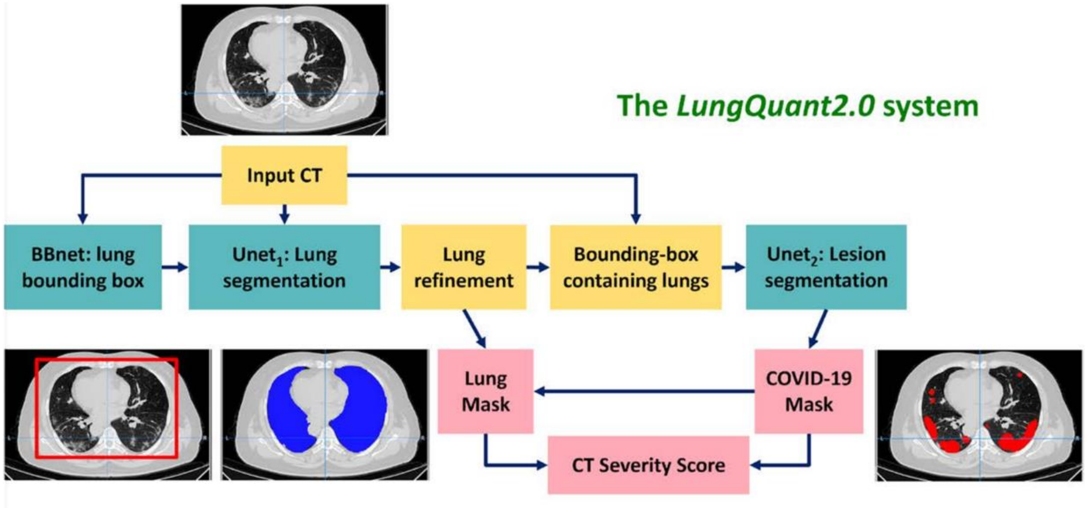
Collegues of the Data Center @ INFN Division of Pisa.

Back-up slides

Il software di segmentazione LungQuant

- U-net in cascata allenate per la segmentazione
  - Maschera del polmone
  - Maschera della lesione COVID-19
  - CT-Severity Score (CTSS), indice di gravità del polmone coinvolto dall'infezione:1 (<5%), 2 (5%-25%), 3 (25%-50%), 4 (50%-75%), 5 (>75%)





The *LungQuant2.0* system

Input CT

BBnet: lung bounding box

Unet$_1$: Lung segmentation

Lung refinement

Bounding-box containing lungs

Unet$_2$: Lesion segmentation

Lung Mask

COVID-19 Mask

CT Severity Score

$$L = Dice_{loss} + CE_{weighted}$$

$$CE_{weighted} = w(x) \sum_{x \in \Omega} log(M_{true}(x) \cdot M_{pred}(x))$$

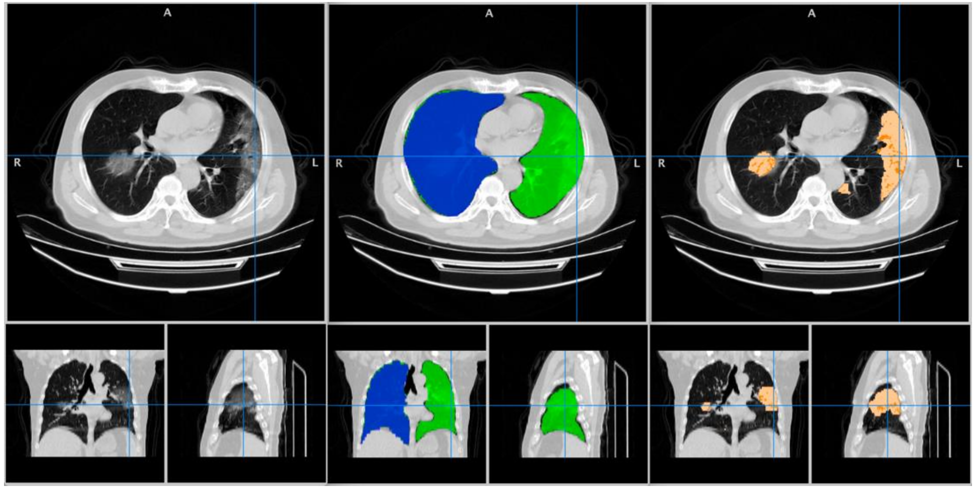
[F.Lizzi et al. Quantification of pulmonary involvement in COVID-19 pneumonia by means of a cascade of two U-nets: training and assessment on multiple datasets using different annotation criteria. International Journal of Computer Assisted Radiology and Surgery, 2021.]

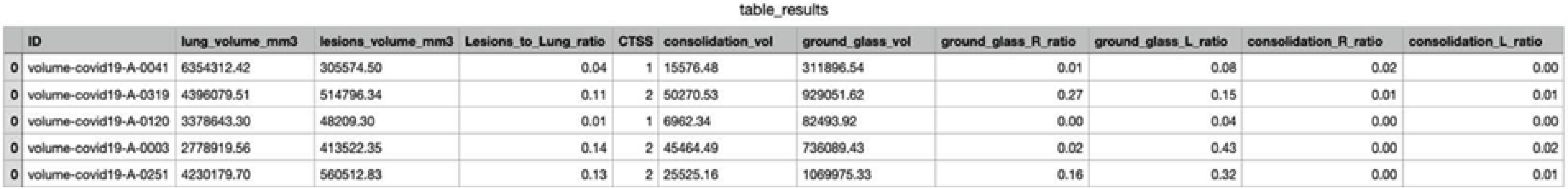
Il software di segmentazione LungQuant

- Versione 2.0
- CNN basata su regressione all’inizio della pipeline per predire il bounding box attorno ai polmoni.
- Funzione per separare polmone destro e sinistro.
- Soglia per distinguere GGO da consolidamenti.

	sDSC (5mm)	vDSC
Lung	0.97 ± 0.01	0.96 ± 0.01
COVID Lesion	0.83 ± 0.07	0.69 ± 0.08

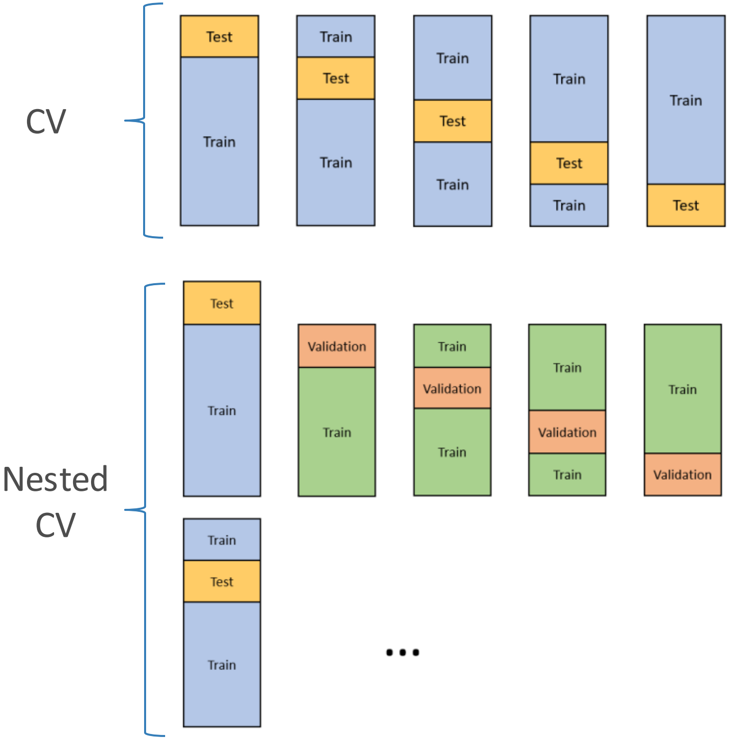
CT-SS prediction: Accuracy: 80%




table_results

	ID	lung_volume_mm3	lesions_volume_mm3	Lesions_to_Lung_ratio	CTSS	consolidation_vol	ground_glass_vol	ground_glass_R_ratio	ground_glass_L_ratio	consolidation_R_ratio	consolidation_L_ratio
0	volume-covid19-A-0041	6354312.42	305574.50	0.04	1	15576.48	311896.54	0.01	0.08	0.02	0.00
0	volume-covid19-A-0319	4396079.51	514796.34	0.11	2	50270.53	929051.62	0.27	0.15	0.01	0.01
0	volume-covid19-A-0120	3378643.30	48209.30	0.01	1	6962.34	82493.92	0.00	0.04	0.00	0.00
0	volume-covid19-A-0003	2778919.56	413522.35	0.14	2	45464.49	736089.43	0.02	0.43	0.00	0.02
0	volume-covid19-A-0251	4230179.70	560512.83	0.13	2	25525.16	1069975.33	0.16	0.32	0.00	0.01

Machine Learning pipeline: Nested CV



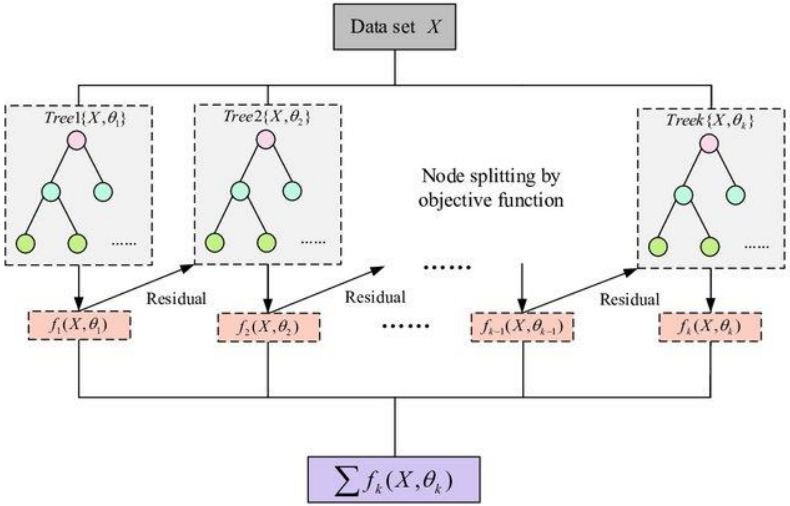
La Cross-validation può essere usata sia per l'ottimizzazione degli iperparametri sia per stimare la generalizzabilità del modello. Tuttavia, usarla per entrambi gli scopi contemporaneamente può portare a una sottostima dell'overfitting dalla procedura stessa di ottimizzazione.

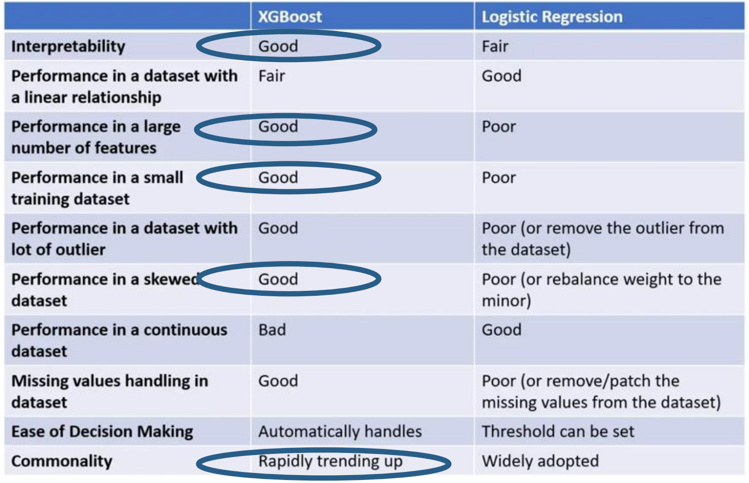
Serve un altro loop esterno di cross-validazione per valutare la generalizzabilità della performance.

- 10 trials
- 15% del dataset totale in Test
- 4-fold CV del restante 85% dei dati

Machine Learning pipeline: XGBoost classifier

EXTREME GRADIENT BOOSTING





	XGBoost	Logistic Regression
Interpretability	Good	Fair
Performance in a dataset with a linear relationship	Fair	Good
Performance in a large number of features	Good	Poor
Performance in a small training dataset	Good	Poor
Performance in a dataset with lot of outlier	Good	Poor (or remove the outlier from the dataset)
Performance in a skewed dataset	Good	Poor (or rebalance weight to the minor)
Performance in a continuous dataset	Bad	Good
Missing values handling in dataset	Good	Poor (or remove/patch the missing values from the dataset)
Ease of Decision Making	Automatically handles	Threshold can be set
Commonality	Rapidly trending up	Widely adopted

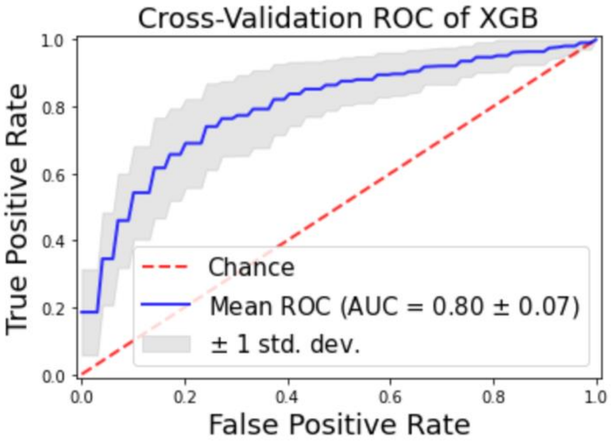
Alberi decisionali aggiunti in forma sequenziale per correggere gli errori di predizione del modello precedente.
Il peso delle variabili predette erroneamente dall'albero viene aumentato e queste variabili vengono poi date in pasto al modello successivo.
L'ensamble di tutti gli alberi dà un modello più robusto e preciso.
Gradient Boosting perché usa un algoritmo gradient descent per minimizzare la loss quando vengono aggiunti i modelli.

- gamma: [0.5, 1, 1.5, 2, 5]
- max_depth: [3, 4, 5]
- subsample: [0.6, 0.8, 1.0]
- colsample_bytree: [0.6, 0.8, 1.0]
- min_child_weight: [1, 5, 10]
- learning_rate=0.02
- n_estimators=600
- objective='binary:logistic'
- nthread=1

Combinazione dei dataset

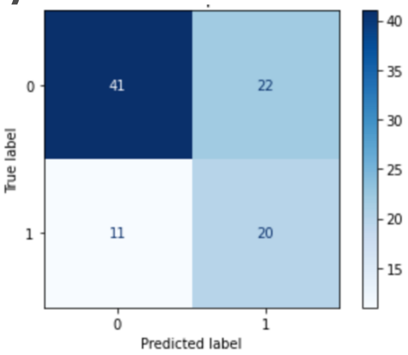
Cross-Validazione (Firenze + Milano + Palermo B80f) con best parameters ricavati dalla nested CV

roc_auc
0.80 ± 0.07



Train (Firenze + Milano + Palermo B80f) Test (Pisa + Pavia)

roc_auc
0.61



Train (Firenze + Milano + Palermo) Test (Pisa)

roc_auc
0.61

Train (Firenze + Milano + Palermo) Test (Pavia)

roc_auc
0.58

Parametri XGBoost

- gamma: parametro di regolarizzazione
- max_depth: profondità max dell'albero
- subsample: fraction of the training set that can be used to train each tree
- colsample_bytree: fraction of the features that can be used to train each tree.
- min_child_weight: minimum sum of instance weight (hessian) needed in a child.
- n_estimators: number of trees
- objective: objective function
- nthread: maximum number of threads available

Parametri per armonizzazione

	PALERMO	PISA	PAVIA	FIRENZE	MILANO
Filtro di ricostruzione	B20f B80f	LUNG	FC51 – B80f	Parenchima	Diversi filtri sharp
Slice thickness	1,0 mm	2,5 mm	[0,5 - 2,0] mm	[1,0 – 2,0] mm	[1,0 – 3,0] mm

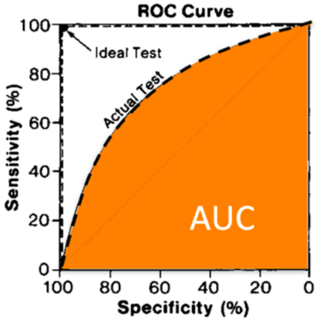
kVp, Pixel Spacing, Manufacturer, FOV (acq – rec), sex, age

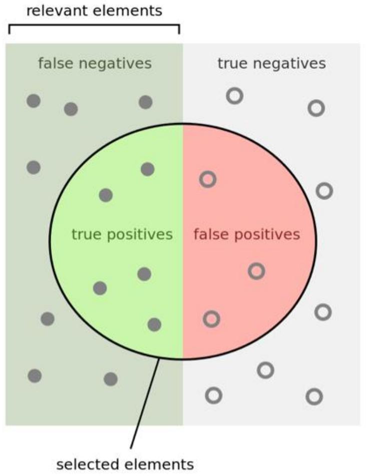
Performance metrics

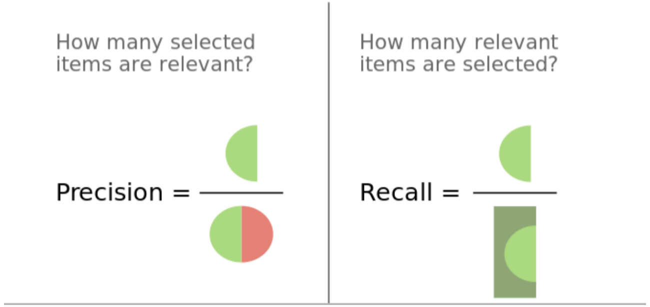
Performance evaluation

CONFUSION MATRIX		Predicted class	
		YES	NO
Actual class	YES	True Positive (TP)	False Negative (FN)
	NO	False Positive (FP)	True Negative (TN)

Area Under the ROC Curve (AUC)






How many selected items are relevant?
How many relevant items are selected?
Precision =
Recall =

Sensitivity = True Positive Rate (TPR)

Specificity =1 - False Positive Rate (FPR)

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

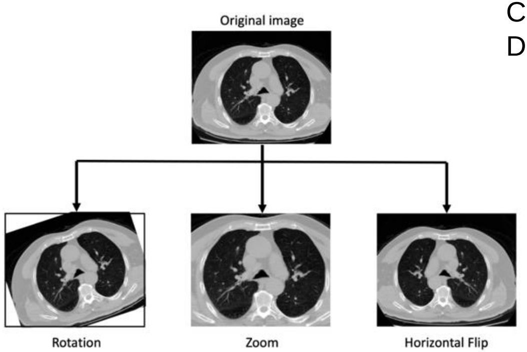
Pre-processing of CXR images

Gray level encoding

Image standardization by subtracting the mean and dividing by the standard deviation of each image.
Resize to 50x50 to allow for a light and small CNN

Lung segmentation

A contrast stretching between their 5th and 95th percentile and normalized.
Cropped and resized to 512x512.
Data augmentation: rotations, zoom, and horizontal flip



Transformation	Parameters
Permitted Rotation Angles	-20°,-15°, -10°, 10°, 15°, 20°
Zoom Percentage	5%, 4%, 3%, 2%

Severity prediction - images

Same as for segmentation
Resize to 1024x1024
Each image was Z-scored

Severity prediction - clinical features

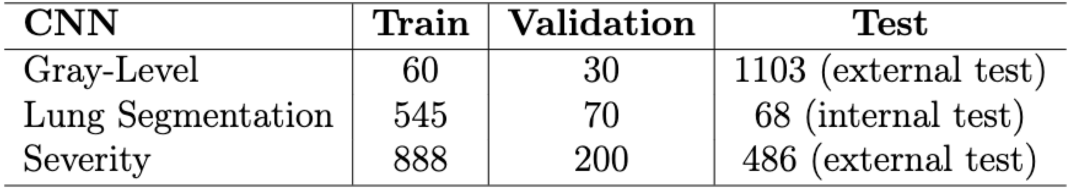
- Categorical features: missing entries (few) replaced with the most frequent class
- When missing in many patients and we filled it by training a K-Nearest Neighbors (KNN) algorithm to assign the value.
- Continuous variables: missing data produced with univariate mean imputing.

CXR images split

The Gray Net Model was trained on 120 pre-processed images selected by visual assessment among the 1103 ones of the AI4COVID dataset, 60 for each class. 30 images were used for the validation set. Then the prediction was computed on the whole AI4COVID training dataset.

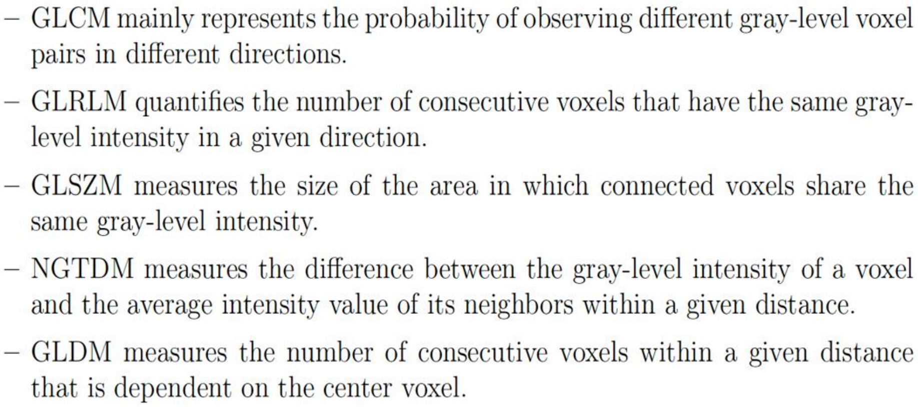
The U-net devoted to lung segmentation was trained on the Shenzhen and Montgomery datasets, divided into a training set (80%), a validation set (10%), and a test set (10%). The performance was evaluated with the mean DSC on the internal test set.

For the Severity net, 888 images of the preprocessed dataset were used for the training of the model, 200 for the validation, and the 486 additional CXR scans made available during the challenge as an external test set.

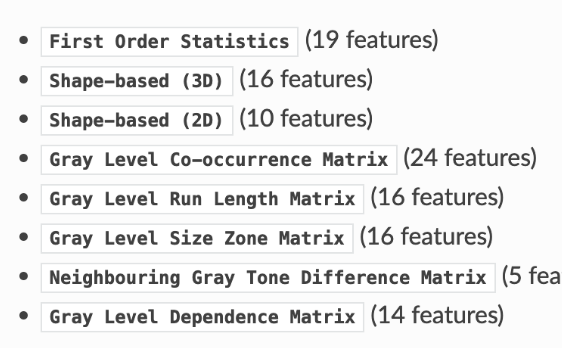


CNN	Train	Validation	Test
Gray-Level	60	30	1103 (external test)
Lung Segmentation	545	70	68 (internal test)
Severity	888	200	486 (external test)

Radiomic features

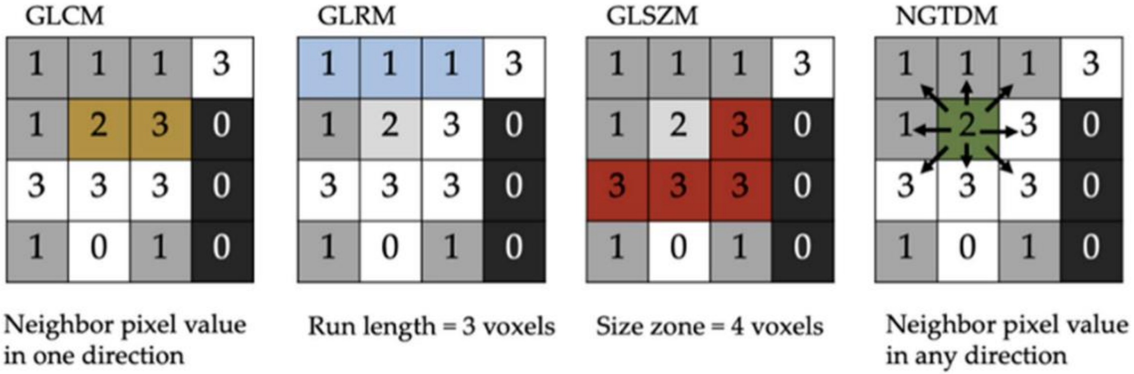


- GLCM mainly represents the probability of observing different gray-level voxel pairs in different directions.
- GLRLM quantifies the number of consecutive voxels that have the same gray-level intensity in a given direction.
- GLSZM measures the size of the area in which connected voxels share the same gray-level intensity.
- NGTDM measures the difference between the gray-level intensity of a voxel and the average intensity value of its neighbors within a given distance.
- GLDM measures the number of consecutive voxels within a given distance that is dependent on the center voxel.



- `First Order Statistics` (19 features)
- `Shape-based (3D)` (16 features)
- `Shape-based (2D)` (10 features)
- `Gray Level Co-occurrence Matrix` (24 features)
- `Gray Level Run Length Matrix` (16 features)
- `Gray Level Size Zone Matrix` (16 features)
- `Neighbouring Gray Tone Difference Matrix` (5 features)
- `Gray Level Dependence Matrix` (14 features)

[Scapicchio, Camilla, et al. "A deep look into radiomics." *La radiologia medica* 126.10 (2021): 1296-1311].



GLCM

1	1	1	3
1	2	3	0
3	3	3	0
1	0	1	0

Neighbor pixel value in one direction

GLRM

1	1	1	3
1	2	3	0
3	3	3	0
1	0	1	0

Run length = 3 voxels

GLSZM

1	1	1	3
1	2	3	0
3	3	3	0
1	0	1	0

Size zone = 4 voxels

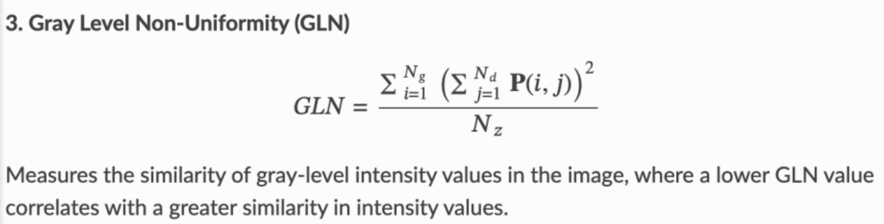
NGTDM

1	1	1	3
1	2	3	0
3	3	3	0
1	0	1	0

Neighbor pixel value in any direction

Radiomic features

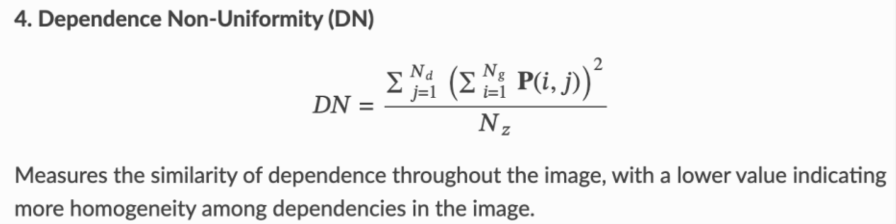
GLDM



3. Gray Level Non-Uniformity (GLN)

$$GLN = \frac{\sum_{i=1}^{N_g}\left(\sum_{j=1}^{N_d}\mathbf{P}(i,j)\right)^2}{N_z}$$

Measures the similarity of gray-level intensity values in the image, where a lower GLN value correlates with a greater similarity in intensity values.

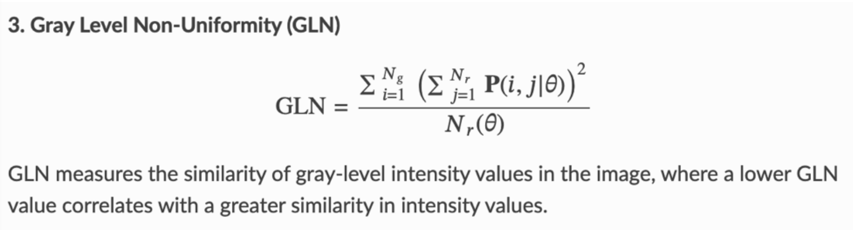


4. Dependence Non-Uniformity (DN)

$$DN = \frac{\sum_{j=1}^{N_d}\left(\sum_{i=1}^{N_g}\mathbf{P}(i,j)\right)^2}{N_z}$$

Measures the similarity of dependence throughout the image, with a lower value indicating more homogeneity among dependencies in the image.

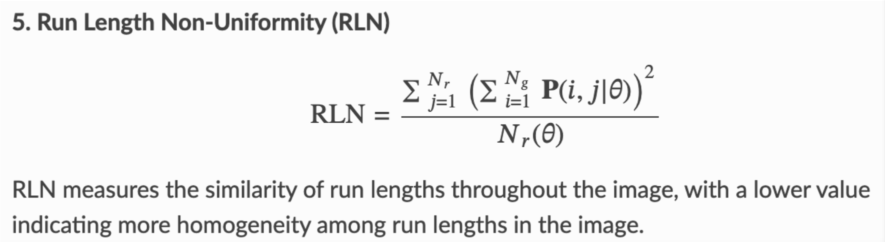
GLRLM



3. Gray Level Non-Uniformity (GLN)

$$\text{GLN} = \frac{\sum_{i=1}^{N_g}\left(\sum_{j=1}^{N_r}\mathbf{P}(i,j|\theta)\right)^2}{N_r(\theta)}$$

GLN measures the similarity of gray-level intensity values in the image, where a lower GLN value correlates with a greater similarity in intensity values.

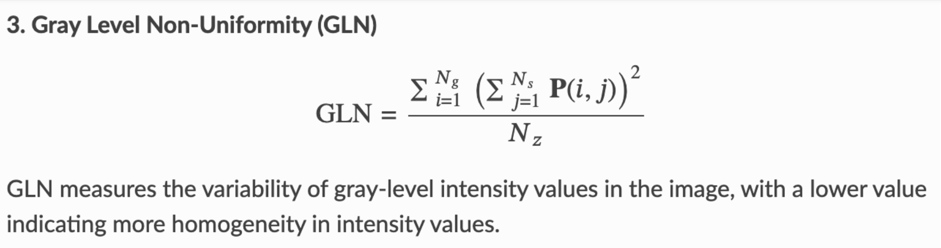


5. Run Length Non-Uniformity (RLN)

$$\text{RLN} = \frac{\sum_{j=1}^{N_r}\left(\sum_{i=1}^{N_g}\mathbf{P}(i,j|\theta)\right)^2}{N_r(\theta)}$$

RLN measures the similarity of run lengths throughout the image, with a lower value indicating more homogeneity among run lengths in the image.

GLSZM



3. Gray Level Non-Uniformity (GLN)

$$\text{GLN} = \frac{\sum_{i=1}^{N_g}\left(\sum_{j=1}^{N_s}\mathbf{P}(i,j)\right)^2}{N_z}$$

GLN measures the variability of gray-level intensity values in the image, with a lower value indicating more homogeneity in intensity values.

Feature Selection methods

- **minimum Redundancy – Maximum Relevance (mRMR):** works iteratively by assigning a score to each feature given by the ratio of the feature's relevance for the target to be predicted and the redundancy with the features selected in the previous iteration. Thus, an algorithm that goes to select the k most significant and least redundant features for the target to be predicted, and agnostically concerning the type of classifier we use. The mrmr_selection publicly available package was used and imported in Python and k =15 was set as the number of selected features.

- **Feature Importance**: this works by assigning a score to individual features based on relevance, but it does not consider redundancy with previous features and depends on the specific classifier used. Feature Importance in XGBoost was used through the feature_importances_ attribute of scikit-learn. Since the selected features can change in the various trials within the nested CV, to obtain a ranking of the features, a new score was given to each feature, which is the total score on all trials, given by the sum of the scores for individual trials. The first 15 features were then considered in this final ranking.

- **Mutual Information**: which measures the uncertainty reduction for one variable given the known value of another variable. The mutual_info_classif function of scikit-learn was used, considering the 15 best features. 15 was chosen as the number of selected features as a typically applied empirical rule
(Widrow-Hoff learning rule) suggests approximately 10 data (in our case patients) for each imaging feature used in the model and, in this case, the most populated dataset of Milan with 160 patients was considered as a reference

ComBat HARMONIZATION METHOD

This method is based on the empirical Bayes frameworks and was initially developed for large-scale genomic data analysis to remove batch effects.

In medical imaging radiomics, batches refer to scanners, imaging protocols, individual imaging parameters, etc.
Unlike imaging harmonization, the ComBat method operates directly on the computed feature values to remove any batch-induced bias.

This is a data-driven method that identifies the protocol effect assuming that the value of each feature *y*, measured in the VOI *j*, with imaging protocol *i*, can be written as

$$y_{ij} = \alpha + \gamma_i + \delta_i \epsilon_{ij}$$

α is the average value for feature y_{ij} ,
γ_i is an additive protocol effect,
δ_i is a multiplicative protocol effect affected by an error term (ε_{ij})

The compensation consists in estimating the model parameters α, γ_i, and δ_i by using a maximum likelihood approach on the basis of the set of available observations:

$$y_{ij}^{ComBat} = \frac{y_{ij} - \hat{\alpha} - \hat{\gamma}_i}{\hat{\delta}_i} + \hat{\alpha}$$