

Supplementary Materials

Multi-classifier-based Identification of COVID-19 from Chest CT using Generalizable and Interpretable Radiomics Features

Appendix A: CT scanning parameters

In the study population, patients from the hospitals in China were imaged with a CT slice of 1-mm thickness on a GE Revolution 256 scanner (GE Medical Systems, Waukesha, USA), or imaged with a CT slice of 5-mm thickness on a NeuViz 128 scanner (Neusoft, Shenyang, China), or with a CT slice of 5-mm thickness on a NeuViz 64 scanner (Neusoft, Shenyang, China).

CT scans of the patients from the hospital from the United States were obtained at 1-3 mm slice thickness with or without contrast (Lightspeed VCT and Revolution, GE Healthcare, Milwaukee, WI; Aquilion, Toshiba Medical Systems, Otawara, Japan; SOMATOM, Siemens Healthineers, Erlangen, Germany). CT images were reviewed by radiologists at each center having more than ten years of individual experience.

Appendix B: Samples of the manually segmentation of pneumonia lesions

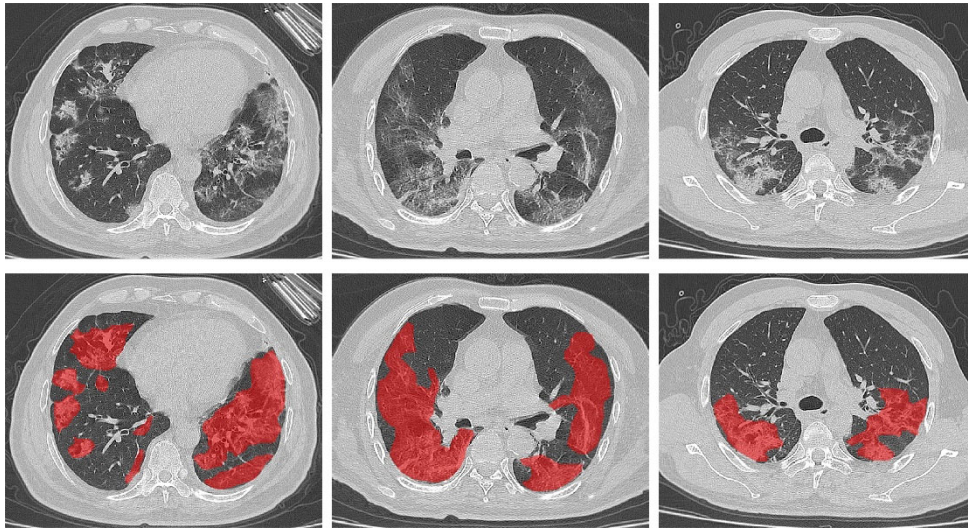


Figure S1. The mask of the region of interest of SARS-CoV-2 positive pneumonia lesions which manually segmented by radiologists. The three CT slices were from different patients.

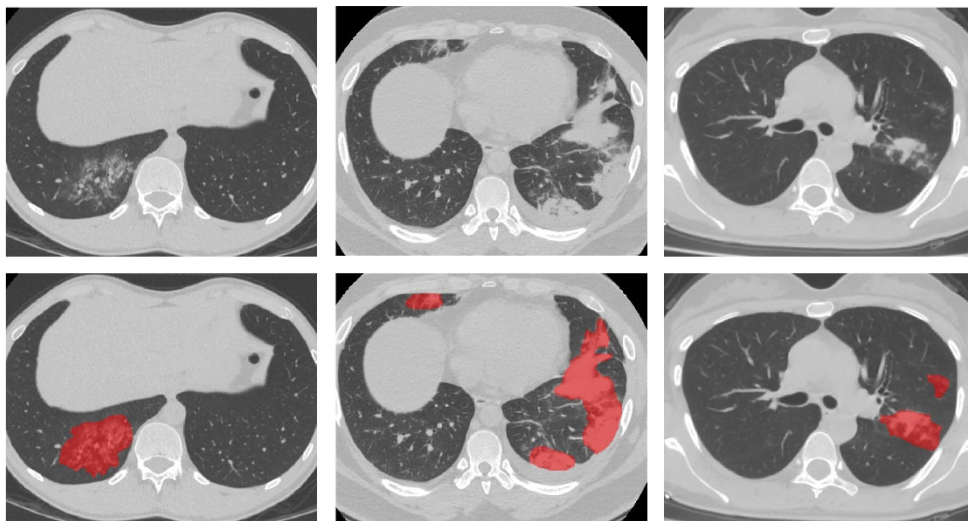


Figure S2. The mask of the region of interest of SARS-CoV-2 negative pneumonia lesions which manually segmented by radiologists. The three CT slices were from different patients.

Appendix C: Selected significant radiomics features by each classifier

Lasso Classifier = 3.653673

*+ 4.958981e-03 * diagnostics_Image-original_Mean +*
*-4.263274e-05 * diagnostics_Image-original_Minimum +*
*3.322157e-01 * diagnostics_Mask-original_VolumeNum +*
*3.989302e-03 * original_shape_MajorAxisLength +*
*1.870174e-03 * original_shape_Maximum2DDiameterColumn +*
*-1.788563e-02 * original_shape_SurfaceVolumeRatio +*
*-1.297587e-04 * original_firstorder_RootMeanSquared +*
*2.658778e-01 * original_firstorder_Skewness +*
*-1.333541e+01 * original_firstorder_Uniformity +*
*1.022123e-03 * original_glcM_Contrast +*
*1.067405 * original_gldm_DependenceNonUniformityNormalized +*
*-7.053722 * original_gldm_LowGrayLevelEmphasis +*
*2.146219e-05 * original_gldm_SmallDependenceHighGrayLevelEmphasis +*
*-7.310696 * original_gldm_SmallDependenceLowGrayLevelEmphasis +*
*-1.707699e-01 * original_glrIm_ShortRunLowGrayLevelEmphasis +*
*-6.043230e-05 * original_glszm_LargeAreaHighGrayLevelEmphasis +*
*-8.216791e-01 * original_ngtdm_Busyness*

Linear classifier ($p < 0.0001$):

[1] diagnostics_Image-original_Mean

- [2] diagnostics_Mask-original_VoxelNum
- [3] diagnostics_Mask-original_VolumeNum
- [4] original_shape_Maximum2DDiameterSlice
- [5] original_shape_MeshVolume
- [6] original_shape_Sphericity
- [7] original_shape_SurfaceArea
- [8] original_firstorder_Entropy
- [9] original_firstorder_InterquartileRange
- [10] original_firstorder_MeanAbsoluteDeviation
- [11] original_firstorder_Minimum
- [12] original_firstorder_RobustMeanAbsoluteDeviation
- [13] original_firstorder_RootMeanSquared
- [14] original_glcm_DifferenceEntropy
- [15] original_glcm_Id
- [16] original_glcm_Idm
- [17] original_glcm_Imc2
- [18] original_glcm_InverseVariance
- [19] original_glcm_JointAverage
- [20] original_glcm_SumEntropy
- [21] original_gldm_DependenceEntropy
- [22] original_gldm_GrayLevelNonUniformity
- [23] original_gllm_GrayLevelNonUniformity

- [24] original_grlm_RunEntropy
- [25] original_glszm_GrayLevelNonUniformity
- [26] original_firstorder_Uniformity

Least absolute shrinkage and selection operator (λ .min):

- [1] diagnostics_Image-original_Mean
- [2] diagnostics_Image-original_Minimum
- [3] diagnostics_Mask-original_VolumeNum
- [4] original_shape_MajorAxisLength
- [5] original_shape_Maximum2DDiameterColumn
- [6] original_shape_SurfaceVolumeRatio
- [7] original_firstorder_RootMeanSquared
- [8] original_firstorder_Skewness
- [9] original_firstorder_Uniformity
- [10] original_glcm_Contrast
- [11] original_gldm_DependenceNonUniformityNormalized
- [12] original_gldm_LowGrayLevelEmphasis
- [13] original_gldm_SmallDependenceHighGrayLevelEmphasis
- [14] original_gldm_SmallDependenceLowGrayLevelEmphasis
- [15] original_grlm_ShortRunLowGrayLevelEmphasis
- [16] original_glszm_LargeAreaHighGrayLevelEmphasis
- [17] original_ngtdm_Busyness

Random forest (Top-20 features of significance):

1. diagnostics_Image-original_Mean
2. original_glrlm_ShortRunLowGrayLevelEmphasis
3. original_glrlm_LowGrayLevelRunEmphasis
4. original_gldm_LowGrayLevelEmphasis
5. original_glszm_LowGrayLevelZoneEmphasis
6. original_shape_Maximum2DDiameterSlice
7. original_firstorder_Median
8. original_shape_Maximum3DDiameter
9. original_shape_MajorAxisLength
10. original_ngtdm_Coarseness
11. original_gldm_LargeDependenceHighGrayLevelEmphasis
12. diagnostics_Image-original_Maximum
13. original_firstorder_RootMeanSquared
14. diagnostics_Image-original_Minimum
15. original_shape_Maximum2DDiameterColumn
16. original_glszm_ZoneVariance
17. original_firstorder_10Percentile
18. original_firstorder_Uniformity
19. diagnostics_Mask-original_VolumeNum
20. original_firstorder_Minimum

Appendix D: The procedure of radiomics model training by LASSO

The R language was used to construct the LASSO radiomics classifier based on the features extracted by Pyradiomics, and the construction of other classifiers is similar to this process. First, we input the COVID-19 status of each image (1 meant positive and 0 meant negative) and the corresponding 120-dimensional radiomics feature of each image into R. Then, the "glmnet" and "cv.glmnet" functions were used to obtain the coefficients of each feature by LASSO classifier and the final LASSO model, respectively. Finally, the LASSO radiomics model was further tested.

The above-mentioned well-trained LASSO radiomics model is included as supplementary study data. Please use the command "load()" to reuse the model. The source code of this algorithm is shown below.

```
y<- TrainData$COVID_Status

TrainData_Feature_Lasso <- TrainData$Feature

names(TrainData_Feature_Lasso) <- NULL

TrainData_Feature_Lasso <-data.matrix(TrainData_Feature_Lasso)

fit<-glmnet(TrainData_Feature_Lasso,y,alpha=1,family='binomial')

cv.fit<-cv.glmnet(TrainData_Feature_Lasso,y,family="binomial")

Test <-predict(fit,type="response",newx = Test_Feature,s=cv.fit$lambda.1se)

Roc_Test<-roc(TestData$COVID_Status, Test)

auc(Roc_Test)
```