Online Data Supplement

Supplementary methods

Supplementary methods S1. CT imaging protocols for the COVID-19 dataset and CT-EGFR dataset

COVID-19 dataset: This dataset was collected from six hospitals: Renmin Hospital of Wuhan University (IRB: WDRY2020-K088), Henan Provincial People's Hospital (IRB: 202029), the First Affiliated Hospital of Anhui Medical University (IRB: PJ2020-02-10), Beijing Youan Hospital of Capital Medical University, Huangshi Central Hospital, and the First Hospital of China Medical University (IRB: 2020-215-2). In the six involved hospitals, CT scans were performed using one of the following scanners: Philips Brilliance Big Bore, Philips Brilliance 16, GE Light speed VCT 4, GE Light speed Pro 16, GE Optima CT540, GE Optima CT660, GE Optima CT680, GE Discovery CT750 HD, Siemens Emotion 16, and Toshiba Aquilion. More than 80% of the CT images were acquired using GE scanners, and about 16% of the CT images were acquired from Philips scanners. For each CT scanner, we chose the images with the sharpest reconstruction kernel. For GE scanners, we chose the images reconstructed with LUNG kernel; for Philips scanners, we chose images constructed with YA or L kernel; for Siemens scanner, we used the B70 kernel; for Toshiba scanner, we chose the FC51 kernel. All the patients underwent spiral CT scans from the lung apex to base at suspended maximum inspiration. The scans were performed at tube voltage 120 kV, tube current 200~500 mAs, rotation time 0.4~0.7 s, pixel matrix 512 × 512. Most CT scans were reconstructed with a slice thickness $\leqslant$ 5 mm.

CT-EGFR dataset: Patients in this dataset were collected from West China Hospital of Sichuan University. There were 4106 patients with lung cancer in this

dataset, including 1991 EGFR-wild type patients and 2115 EGFR-mutant patients. The CT scans were performed using one of the following scanners: Philips Brilliance Big Bore, and GE Discovery CT750 HD. All the patients underwent spiral CT scans from the lung apex to base at suspended maximum inspiration. The scans were performed at tube voltage 120 kV, tube current 200~500 mAs, rotation time 0.4~0.7 s, pixel matrix 512×512. Most CT scans were reconstructed with a slice thickness of 5 mm with the same increment.

Supplementary methods S2. Criteria of discharging hospital for patients with COVID-19

The recovery and discharge criteria were: normal body temperature for greater than 3 days, and significantly improved respiratory symptoms, and significantly improved exudative lesions through radiological evaluation, and two consecutive negative nucleic acid detection with at least 24 hours apart. (National Health Commission of the People's Republic of China. Diagnosis and treatment protocols of pneumonia caused by a novel coronavirus (trial version 6))

Supplementary methods S3. Mathematical description of the DL model

The computational units in the DL model are defined as layers, which include convolution, activation, pooling and batch normalization. The details are explained as following.

***Convolution*.** Convolution is used to extract features from CT image. Different convolutional filters can extract different features to characterize the lung. Assuming matrix $I = \begin{pmatrix} I_{11} & I_{12} & I_{13} \\ I_{21} & I_{22} & I_{23} \\ I_{31} & I_{32} & I_{33} \end{pmatrix}$ is the mathematical representation of the lung CT image,

and matrix $K = \begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix}$ is the convolutional filter. Then, the output of the convolution layer is $F = conv(I, K)$, where *conv* represents convolutional operation. This can be further understood as the following formula.

$$F = conv(I, K)$$

$$= \begin{pmatrix} I_{11} * k_{11} + I_{12} * k_{12} + I_{21} * k_{21} + I_{22} * k_{22} & I_{12} * k_{11} + I_{13} * k_{12} + I_{22} * k_{21} + I_{23} * k_{22} \\ I_{21} * k_{11} + I_{22} * k_{12} + I_{31} * k_{21} + I_{32} * k_{22} & I_{22} * k_{11} + I_{23} * k_{12} + I_{32} * k_{21} + I_{33} * k_{22} \end{pmatrix}$$

The output F is called feature map.

Activation. After the operation of convolution, the result (feature map) will be activated by an activation function to obtain non-linear features, here we adopt the "ReLU" function [1] $ReLU(x) = max(0, x)$. When the input x is negative, the output of the activation function will be zero, and when the input is positive, the result will be equal to the input.

Pooling. To select representative features that are strongly associated with COVID-19, non-relevant and redundant features need to be eliminated. This is achieved by pooling operation. Assuming the feature map is $F = \begin{pmatrix} 1 & 5 & 2 & 8 \\ 3 & 9 & 7 & 8 \\ 1 & 0 & 2 & 6 \\ 8 & 5 & 3 & 2 \end{pmatrix}$, whose size is 4×4, and pooling window is 2×2 with stride 2. The pooling operation will divide the matrix F into four disjoint small matrixes of size 2×2, each maximum value of the small matrix will be extracted to form the result matrix $P = \begin{pmatrix} 9 & 8 \\ 8 & 6 \end{pmatrix}$.

Batch normalization. To accelerate the training process of the DL model, we use batch normalization [2] operation to normalize the feature maps from each

convolutional layer. This strategy avoids gradient vanishing during training, and therefore accelerates the learning process of the DL model.

Supplementary methods S4. Details of the DenseNet121-FPN and the non-lung area suppression

To segment lung automatically from chest CT images, we used FPN network with DenseNet121 as backbone. The DenseNet121 was pre-trained in ImageNet dataset with more than 1 million natural images. Afterwards, the DenseNet121-FPN was fine-tuned using VESSEL12 dataset. In the VESSEL12 dataset, chest CT image and manual lung annotation of 20 subjects were provided. Specifically, we used every three adjacent CT slices to combine a three-channel image as input to the DenseNet121-FPN network. The code and pretrained weights of the DenseNet121-FPN was available in https://github.com/divamgupta/image-segmentation-keras. During testing phase, lung segmentation was performed slice-by-slice.

In the non-lung area suppression operation, firstly, we sorted the intensities of the whole CT scan, and find the top 5% intensity value (defined as *top_threshold*). Image intensities larger than the *top_threshold* were limited to the *top_threshold*. Secondly, we calculated the mean ($lung_{mean}$) and standard deviation ($lung_{std}$) of the intensities of lung tissues inside lung mask. Afterwards, we limited the intensity of tissues outside lung mask into the range [$lung_{mean}$ - $4lung_{std}$, $lung_{mean}$ + 4 $lung_{std}$]. Through this operation, intensity of non-lung tissues is suppressed to a small range. This operation is important for DL, since DL tends to focus on locations with high intensity or high intensity gradient. Bones or muscles usually have high intensity and sharp edges (gradient), and may mislead the DL model. Consequently, this non-lung area suppression operation can increase the robustness of the DL system.

Supplementary methods S5. Training process of the COVID-19 Net

Model training aims at optimizing the parameters of the DL model to build the relationship between CT image and COVID-19. The model training is an iterative process, which optimizes the model at each iteration until the model achieves the best predictive performance. At each iteration, we used cross entropy as cost function to measure the predictive performance of the DL model:

$$\mathrm{L}(w) = \frac{1}{N}\sum_{n=1}^{N}[y_n log p_n + C(1-y_n)log(1-y_n)]$$

In this formula, *w* was the parameter of the model that needed to be trained; *N* was the training sample number; y_n represented the true pneumonia type of patient (1 for COVID-19, 0 for other types of pneumonia); p_n was the predicted COVID-19 probability. Since the ratio of COVID-19 to other types of pneumonia is imbalanced in the training set, we used a class weight *C*=5 in this formula. If the model falsely predicts other types of pneumonia into COVID-19, the cost value will be 5 times larger than normal. This strategy makes the deep learning model pay more attention to predict other types of pneumonia correctly. If the cost function $\mathrm{L}(w)$ was not minimum, we used SGD algorithm to update the parameters of the DL model and minimize the loss function. The learning rate of SGD was set to 0.005, and was reduced by 0.8 times when the loss stopped decreasing for 2 epochs.

To let the COVID-19Net learn lung features from large dataset, we used CT-EGFR dataset (4106 patients) to pre-train the network. Afterwards, the training set of the COVID-19 dataset was used to fine-tune the COVID-19Net.

Supplementary methods S6. Details of the DL-discovered suspicious lung area

When the DL model is well trained, the network established thousands inference paths that work together for COVID-19 diagnosis. Given a lung-ROI, we calculated the gradient of the predicted COVID-19 probability with respect to the input image. This gradient told us how the predicted probability changes with respect to a small change in voxels in the lung-ROI. Hence, visualizing these gradients helped us to find the attention of the DL model [3, 4].

Supplementary methods S7. Details of the DL feature pattern visualization

We used convolutional filter visualization technique to acquire the feature patterns extracted by convolutional layers [3, 4]. For each convolutional filter in the DL model, we input an image initialized with random white noise to observe the filter response. If the filter response reaches a maximum, the input image reveals the feature pattern extracted by the convolutional filter; otherwise, a back-propagation algorithm was involved to change the input image until the filter response reaches a maximum. Through this convolutional filter visualization method, we can understand the feature patterns extracted by each convolutional filter in the DL model.

Supplementary table S1. Main parameters and structure of the COVID-19Net.

layer	size	parameter
Input	48×240×360	
Convolution	48×120×180@16	Filter = 16, kernel = 3×3×3, stride = 1×2×2
Convolution	48×120×180@16	Filter = 16, kernel = 3×3×3, stride = 1×1×1
DenseBlock1	48×120×180@16	Conv: filter=16, kernel=3, stride=1 Conv: filter=16, kernel=3, stride=1, concatenate Conv: filter=16, kernel=3, stride=1 Conv: filter=16, kernel=3, stride=1, concatenate Conv: filter=16, kernel=1, stride=1
Max pooling	24×60×90@16	Window=2, stride=2
DenseBlock2	24×60×90@24	Conv: filter=24, kernel=3, stride=1 Conv: filter=24, kernel=3, stride=1, concatenate Conv: filter=24, kernel=3, stride=1 Conv: filter=24, kernel=3, stride=1, concatenate Conv: filter=24, kernel=3, stride=1 Conv: filter=24, kernel=3, stride=1, concatenate Conv: filter=24, kernel=1, stride=1
Max pooling	12×30×45@24	Window=2, stride=2
DenseBlock3	12×30×45@152	Conv: filter=32, kernel=3, stride=1 Conv: filter=32, kernel=3, stride=1, concatenate Conv: filter=32, kernel=3, stride=1 Conv: filter=32, kernel=3, stride=1, concatenate Conv: filter=32, kernel=3, stride=1 Conv: filter=32, kernel=3, stride=1, concatenate Conv: filter=32, kernel=3, stride=1 Conv: filter=32, kernel=3, stride=1, concatenate
Max pooling	6×15×23@152	Window=2, stride=2
DenseBlock4	6×15×23@216	Conv: filter=32, kernel=3, stride=1 Conv: filter=32, kernel=3, stride=1, concatenate Conv: filter=32, kernel=3, stride=1 Conv: filter=32, kernel=3, stride=1, concatenate
Max pooling	6×8×12@216	Window=1×2×2, stride=1×2×2
Convolution	6×8×12@64	Conv: filter=64, kernel=1, stride=1
Global average pooling	64	
Fully connection	1	sigmoid

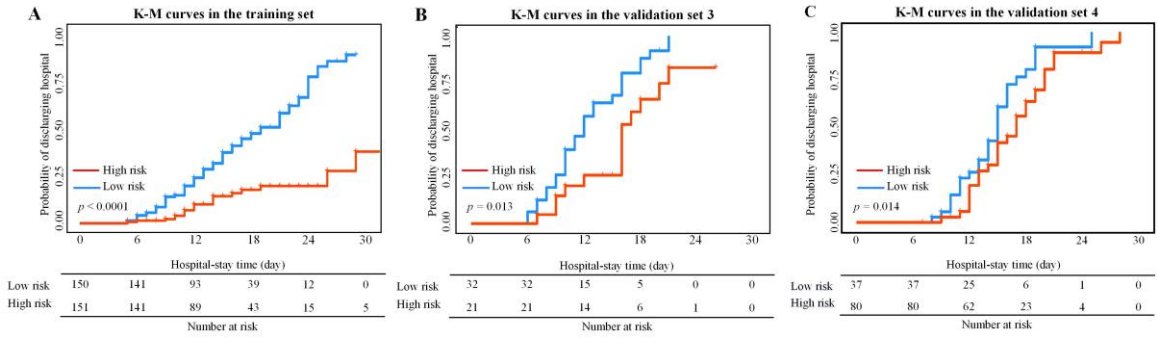
Supplementary table S2. Comparison between the DL systems without auxiliary training (random-DL system) and with auxiliary training (DL system).

Methods	datasets	AUC	Accuracy	Sensitivity	Specificity	F1-score
Random-DL system	Training	0.91 (0.90, 0.91)	79.83%	77.50%	88.59%	85.85%
	Validation 1	0.78 (0.76, 0.80)	68.58%	50.98%	83.06%	59.43%
	Validation 2	0.67 (0.64, 0.70)	60.25%	43.48%	82.61%	55.56%
DL system	Training	0.90 (0.89-0.91)	81.24%	78.93%	89.93%	86.92%
	Validation 1	0.87 (0.86-0.89)	78.32%	80.39%	76.61%	77.00%
	Validation 2	0.88 (0.86-0.90)	80.12%	79.35%	81.16%	82.02%

Random-DL system is the DL system without auxiliary training in the CT-EGFR dataset.
AUC is area under the receiver operating characteristic curve.
Data in parentheses are the 95% confidence interval.

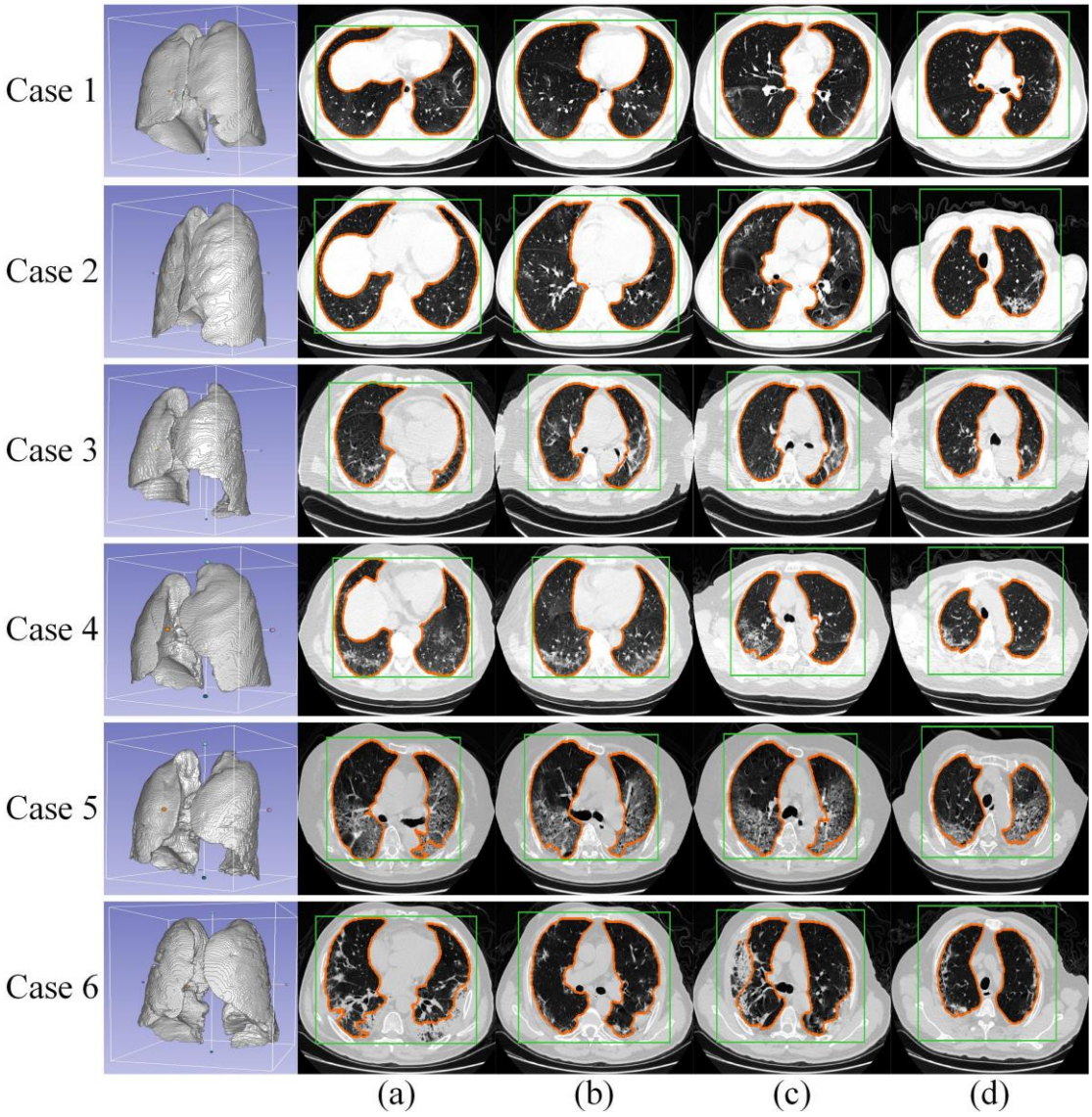
Supplementary table S3. Selected prognostic features.

Feature name	Univariate HR	p-value
age	0.96 (0.95-0.98)	<0.001
DL feature-22	0.04 (0.01-0.25)	<0.001
DL feature-44	0.04 (0.01-0.32)	0.002



Supplementary figure S1. Kaplan-Meier analysis of the prognostic performance of the DL system.

Vertical lines in this figure represents censored data.



Supplementary figure S2. Segmentation results. Each row represents a patient. The first column is the 3-dimensional segmentation results, (a)-(d) are four image slices. Orange contour is the segmentation result by the DenseNet121-FPN model. Green rectangle is the lung-ROI. From case 1 to case 4, the segmentation model generates good results. In case 5a-b and case 6a-c, the segmentation model misses some inflammatory tissues inside lung. However, the lung-ROI can always include all lung areas.

References

1. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems; 2012; 2012. p. 1097-1105.

2. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:150203167* 2015.

3. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In: 2017 IEEE International Conference on Computer Vision (ICCV); 2017; 2017. p. 618-626.

4. Kotikalapudi Rac. keras-vis. GitHub, https://github.com/raghakot/keras-vis, 2017.