iScience, Volume *25*

Supplemental information

Automated segmentation of lungs and lung tumors in mouse micro-CT scans

Gregory Z. Ferl, Kai H. Barck, Jasmine Patil, Skander Jemaa, Evelyn J. Malamut, Anthony Lima, Jason E. Long, Jason H. Cheng, Melissa R. Junttila, and Richard A.D. Carano

Model	Hyperparameter	Type	Search Space	Optimal Hyperparameter	Accuracy		F1 score	
					training	*hold-out*	*training*	*hold-out*
k-nearest neighbors	Coding	categorical	onevsall, onevsone	onevsall	1	0.73	1	0.80
	Num Neighbors	integer	[1, 479]	3				
	Distance	categorical	See footnote*	cosine				
	Distance Weight	categorical	equal, inverse, squaredinverse	inverse				
	Exponent	real	[0.5, 3]	n/a				
	Standardize	categorical	true, false	true				
Support vector machine	Coding	categorical	onevsall, onevsone	onevsone	0.89	0.79	0.93	0.85
	Box Constraint	real	[0.001, 1000]	4.7927				
	Kernel Scale	real	[0.001, 1000]	3.4563				
	Kernel Function	categorical	gaussian, linear, polynomial	gaussian				
	Polynomial Order	integer	[2, 4]	n/a				
	Standardize	categorical	true, false	false				
Classification ensemble method	Method	categorical	Bag, AdaBoostM2, RUSBoost	AdaBoostM2	1	0.79	1	0.85
	Num Learning Cycles	integer	[10, 500]	117				
	LearnRate	real	[0.001, 1]	0.849				
	Min Leaf Size	integer	[1, 970]	1				
	Max Num Splits	integer	[1, 1940]	1914				
	Split Criterion	categorical	gdi, twoing, deviance	twoing				
	Num Variables To Sample	integer	[1, 18]	n/a				
Regression ensemble method	Method	categorical	Bag, LSBoost	LSBoost	0.90	0.74	0.93	0.79
	Num Learning Cycles	integer	[10, 500]	267				
	LearnRate	real	[0.001, 1]	0.033379				
	Min Leaf Size	integer	[1, 970]	2				
	Max Num Splits	integer	[1, 1940]	43				
	Num Variables To Sample	integer	[1, 18]	9				

*cityblock, chebychev, correlation, cosine, euclidean, hamming, jaccard, mahalanobis, minkowski, seuclidean, spearman

Table S1. Classification models and hyperparameter search space, related to Table 1
Each hyperparameter is either a categorical variable, integer or real number, as indicated in the 'Type' column, with potential values listed in the 'Search Space' column; the numbers in square brackets indicate the lower and upper bounds of integer and real number ranges. The optimal hyperparameter values for each classifier listed in the 'Model' column is shown along with the corresponding accuracy and F1 scores on the training and hold-out test sets. Accuracy is calculated as $(TP + TN)/(TP + TN + FP + FN)$ and F1 is calculated as $2TP/(2TP + FP + FN)$, where TP, TN, FP and FN are the number of true positives, true negatives, false positives and false negatives, respectively. Optimal hyperparameters were estimated using the *bayesopt* algorithm (*bayesopt* Copyright 2016-2017 The MathWorks, Inc.).

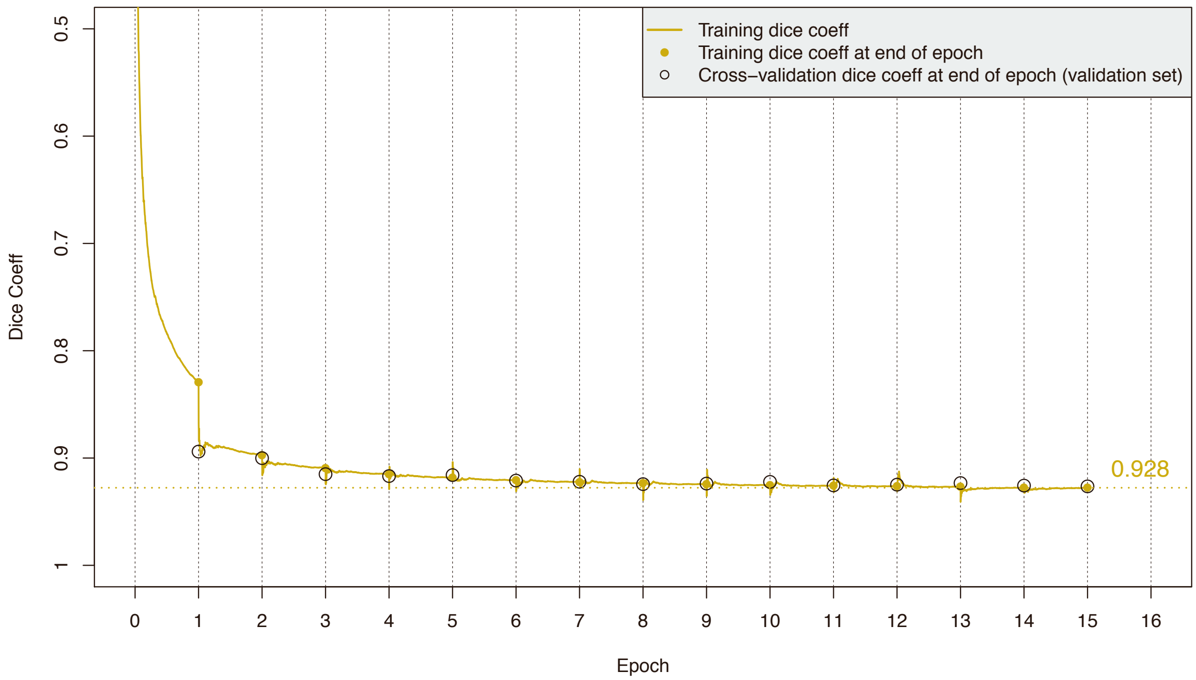

Figure S1. Dice coefficient vs epoch, related to Figure 1
The dice coefficient on the training set after 15 epochs (0.928) is indicated by the dashed horizontal line.

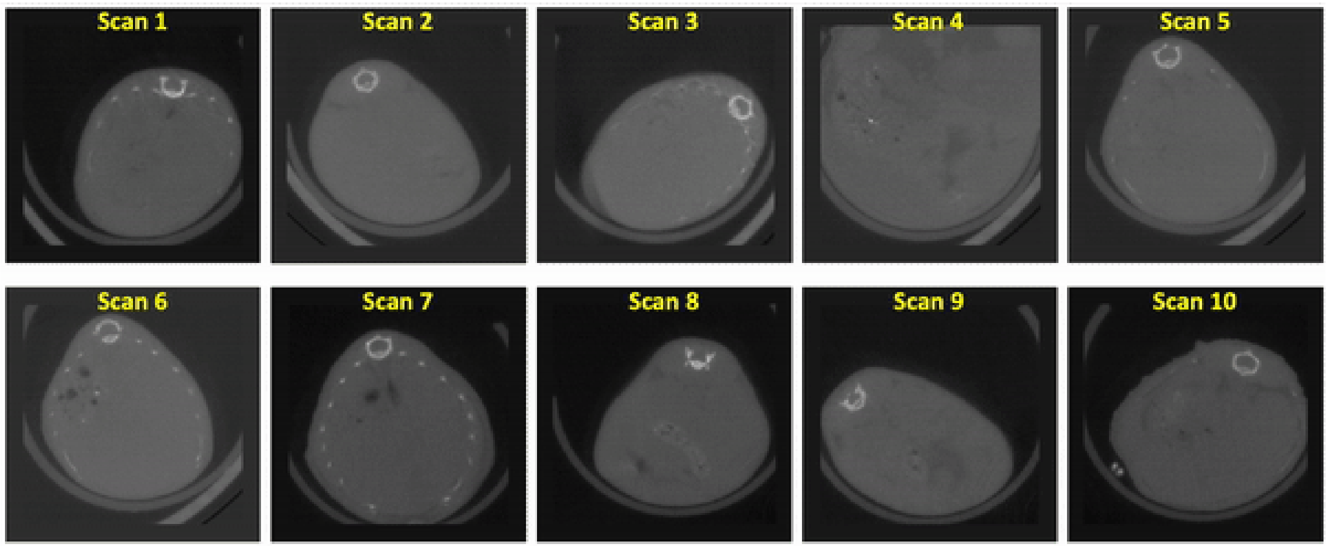

Figure S2. micro-CT images acquired by the CT120 scanner with lung ROIs, related to Figure 1
Comparison of $n = 10$ hand-drawn and 3D U-net-predicted lung ROIs in the transverse plane from the hold-out test image set, where the green ROIs were manually drawn by a human reader and the red ROIs were predicted by the trained 3D U-net. Click figure to animate.

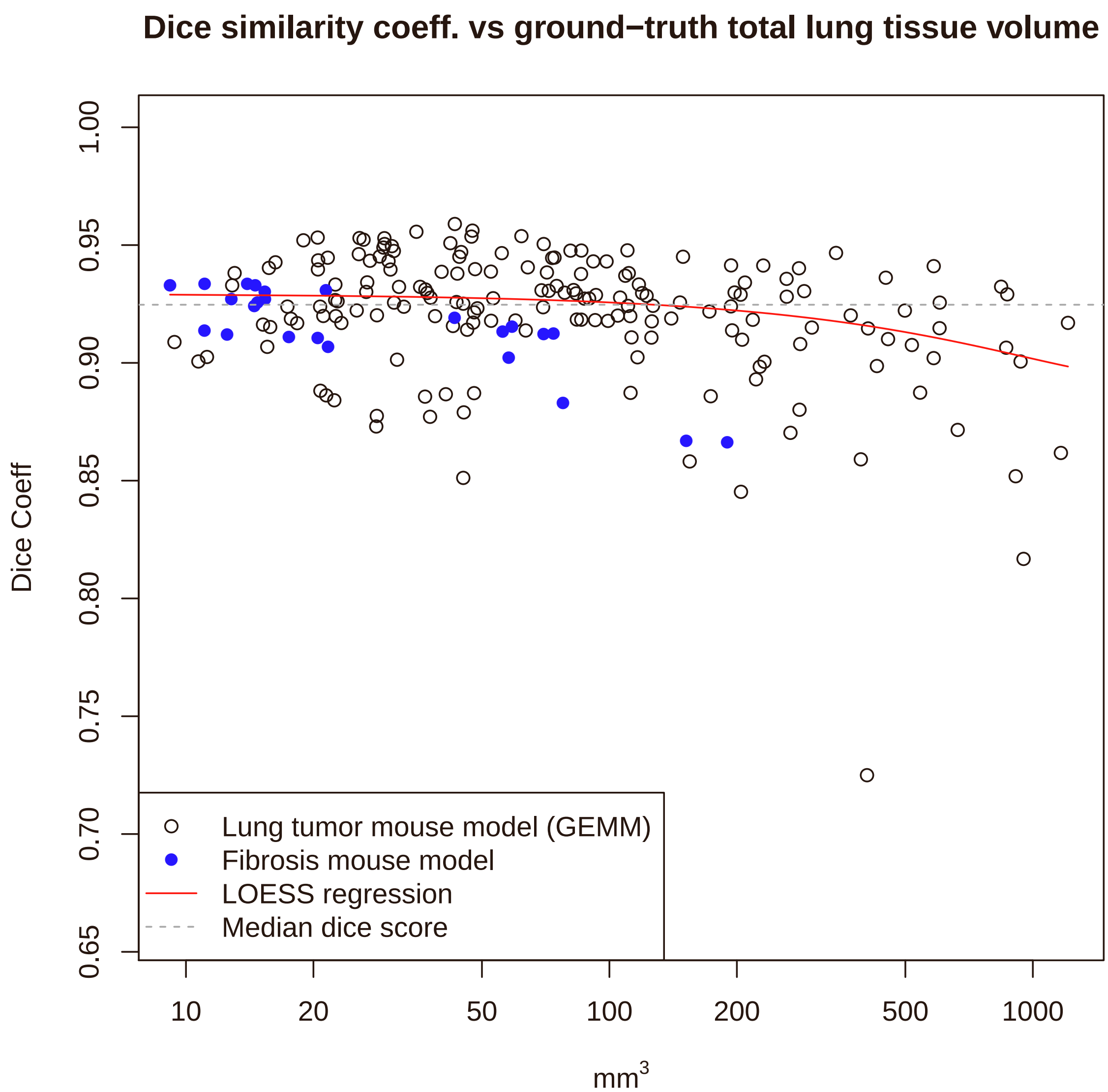

Figure S3. Dice coefficient vs. approximate total lung tissue volume for the 200-scan ct120 hold-out test set, related to Figure 2

Total lung tissue is primarily composed of blood vessels and tumors (GEMM lung tumor mouse model) or fibrotic tissue (fibrosis mouse model). For the $n = 200$ scans shown, open black circles represent values from GEMM mouse model scans, filled blue circles represent values from fibrosis mouse model scans, the median dice coefficient across all scans is shown by the dashed line and the average trend of dice coeffocient vs. tissue volume is captured by the LOESS regression curve shown in red.

```
Requirement already satisfied: nibabel in /gstore/home/ferlg/.local/lib/python3.6/site-packages
Requirement already satisfied: six>=1.3 in /gstore/apps/Anaconda3/5.0.1/lib/python3.6/site-packages (from nibabel)
Requirement already satisfied: numpy>=1.8 in /gstore/apps/Anaconda3/5.0.1/lib/python3.6/site-packages (from nibabel)
You are using pip version 9.0.1, however version 22.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
Requirement already satisfied: SimpleITK in /gstore/home/ferlg/.local/lib/python3.6/site-packages
You are using pip version 9.0.1, however version 22.3.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
MATLAB is selecting SOFTWARE OPENGL rendering.
Opening log file:  /gstore/home/ferlg/java.log.28381

                            < M A T L A B (R) >
                  Copyright 1984-2021 The MathWorks, Inc.
             R2021a Update 3 (9.10.0.1684407) 64-bit (glnxa64)
                                May 27, 2021


To get started, type doc.
For product information, visit www.mathworks.com.


Detected 29 image files in folder "files_in_CTscans"

Downsampling and padding images to 256x256x256 voxels...0.327925 minutes

============================
--------------------------------------------------------------------------------------------------
MATLAB Version: 9.10.0.1684407 (R2021a) Update 3
MATLAB License Number: 120997
Operating System: Linux 3.10.0-1160.66.1.el7.x86_64 #1 SMP Wed May 18 16:02:34 UTC 2022 x86_64
Java Version: Java 1.8.0_202-b08 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode
--------------------------------------------------------------------------------------------------
============================
MATLAB toolboxes used for this analysis:
image_toolbox
matlab
============================


Predicting lung masks for downsampled images using trained 3D U-net:
2022-11-15 08:10:17.074384: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Found device 0 with properties:
name: Quadro P6000 major: 6 minor: 1 memoryClockRate(GHz): 1.645
pciBusID: 0000:0d:00.0
totalMemory: 23.88GiB freeMemory: 23.73GiB
2022-11-15 08:10:17.192984: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1030] Found device 1 with properties:
name: Quadro P6000 major: 6 minor: 1 memoryClockRate(GHz): 1.645
pciBusID: 0000:0e:00.0
totalMemory: 23.88GiB freeMemory: 23.73GiB
2022-11-15 08:10:17.194457: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1045] Device peer to peer matrix
2022-11-15 08:10:17.194506: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1051] DMA: 0 1
2022-11-15 08:10:17.194514: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1061] 0:   Y Y
2022-11-15 08:10:17.194518: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1061] 1:   Y Y
2022-11-15 08:10:17.194540: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1120] Creating TensorFlow device (/device:GPU:0) -> (device: 0, name: Quadro P6000, pci bus id: 0000:0d:00.0, compute capability: 6.1)
2022-11-15 08:10:17.194547: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1120] Creating TensorFlow device (/device:GPU:1) -> (device: 1, name: Quadro P6000, pci bus id: 0000:0e:00.0, compute capability: 6.1)


Lung mask for animal 0 generated in 4.74268364906311 seconds
Lung mask for animal 1 generated in 2.161865472793579 seconds
Lung mask for animal 2 generated in 1.9781420230865479 seconds
Lung mask for animal 3 generated in 1.8721568584442139 seconds
Lung mask for animal 4 generated in 2.3398702144622803 seconds
Lung mask for animal 5 generated in 1.8457610607147217 seconds
Lung mask for animal 6 generated in 1.812135934829712 seconds
Lung mask for animal 7 generated in 1.927943229675293 seconds
Lung mask for animal 8 generated in 1.9030261039733887 seconds
Lung mask for animal 9 generated in 1.9549524784088135 seconds
Lung mask for animal 10 generated in 1.7855937480926514 seconds
Lung mask for animal 11 generated in 1.8462295532226562 seconds
Lung mask for animal 12 generated in 2.0528273582458496 seconds
Lung mask for animal 13 generated in 1.8189809322357178 seconds
Lung mask for animal 14 generated in 1.8579018115997314 seconds
Lung mask for animal 15 generated in 1.965744972229004 seconds
Lung mask for animal 16 generated in 1.8580067157745361 seconds
Lung mask for animal 17 generated in 1.8574190139770508 seconds
Lung mask for animal 18 generated in 1.9503605365753174 seconds
Lung mask for animal 19 generated in 1.8867170810699463 seconds
Lung mask for animal 20 generated in 1.8097031116485596 seconds
Lung mask for animal 21 generated in 2.0715324878692627 seconds
Lung mask for animal 22 generated in 2.3014371395111084 seconds
Lung mask for animal 23 generated in 1.9279725551605225 seconds
Lung mask for animal 24 generated in 1.9085676670074463 seconds
Lung mask for animal 25 generated in 1.807969331741333 seconds
Lung mask for animal 26 generated in 1.8122947216033936 seconds
Lung mask for animal 27 generated in 1.96575927734375 seconds
Lung mask for animal 28 generated in 1.8148307800292969 seconds


Lung masks for all animals generated in 58.841492652893066 seconds
Using TensorFlow backend.




MATLAB is selecting SOFTWARE OPENGL rendering.
Opening log file:  /gstore/home/ferlg/java.log.30251

                            < M A T L A B (R) >
                  Copyright 1984-2021 The MathWorks, Inc.
             R2021a Update 3 (9.10.0.1684407) 64-bit (glnxa64)
                                May 27, 2021


To get started, type doc.
For product information, visit www.mathworks.com.


Upsampling predicted lung masks to original image dimensions...0.684990 minutes

============================
--------------------------------------------------------------------------------------------------
MATLAB Version: 9.10.0.1684407 (R2021a) Update 3
Operating System: Linux 3.10.0-1160.66.1.el7.x86_64 #1 SMP Wed May 18 16:02:34 UTC 2022 x86_64
Java Version: Java 1.8.0_202-b08 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode
--------------------------------------------------------------------------------------------------
============================
MATLAB toolboxes used for this analysis:
image_toolbox
matlab
============================
```

Figure S4. Log file for forward prediction of lung masks, related to STAR Methods
Log file shown for $n = 29$ hold-out test scans. Lungs segmented using the trained 3D U-net and Matlab scripts for image pre- and post-processing.

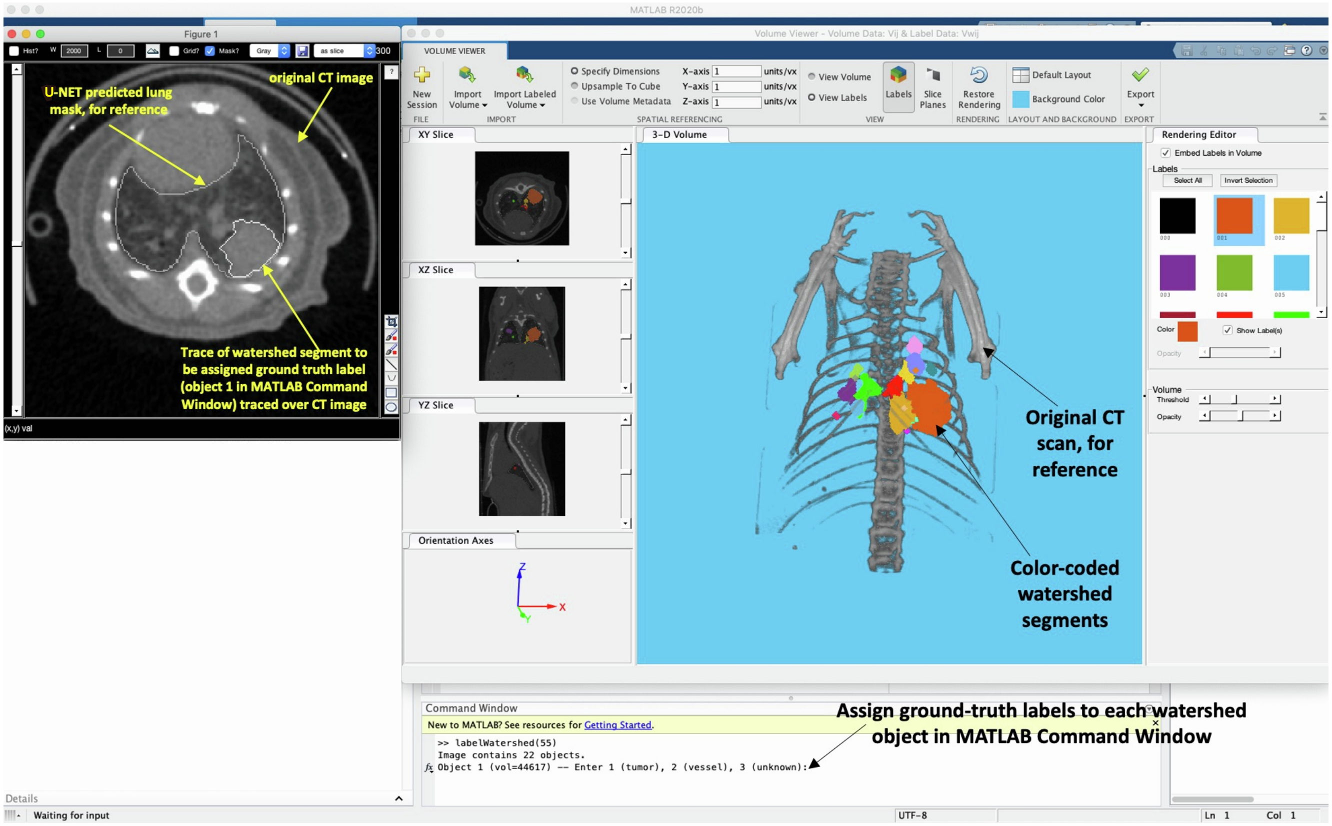

Figure S5. Screenshot of object labeling tool, related to Figure 3
For each scan of interest with n watershed segmentation-generated tissue objects (watershed objects), the original micro-CT image is opened (upper-left window), allowing the user to scroll through the transverse slices with both the 3D U-net-predicted lung ROI and watershed object i of n superimposed on each slice. The labeling tool also provides a 3D rendering of the original micro-CT image with all n watershed objects shown(right-hand window). A tissue label (tumor, vessel or other/unknown) is assigned to each watershed object in the Command Window, where the watershed object shown in the stack of transverse slices is continually updated to show only the object currently under consideration (watershed object i).

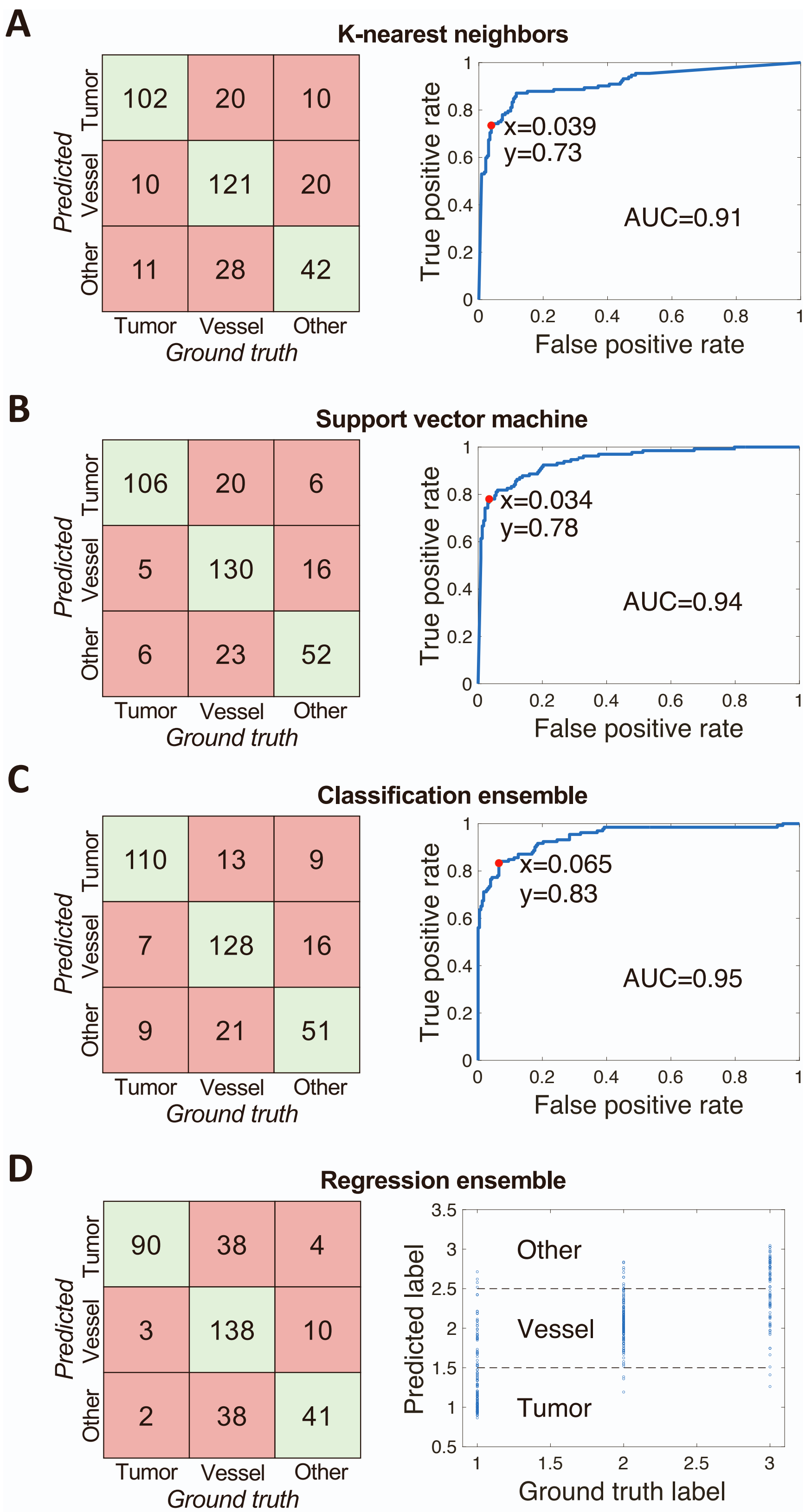

Figure S6. Confusion matrices and receiver operating characteristic (ROC) curves, related to Table 1
Left-hand column shows confusions matrices and right-hand column shows ROCs, for the hold-out test data sets ($n = 364$ tissue objects) corresponding to the trained A) K-nearest neighbors, B) support vector machine and C) classification ensemble models; D) confusion matrix and predicted vs. ground truth tissues labels (right-hand plot) are shown for the trained regression ensemble classification model, where predicted tissue class values of < 1.5, $1.5 - 2.5$, ≥ 2.5 are assigned tumor, vessel and other, respectively. The area under the curve (AUC) and x-, y-coordinates of the optimal operating point are shown for each ROC (*perfcurve* Copyright 2008-2021 The MathWorks, Inc.).

```
ferlg@nc232:/gnet/is2/p01/data/bmi/CT/ferlg/image_processing_pipeline % matlab -nodesktop -nosplash
MATLAB is selecting SOFTWARE OPENGL rendering.
Opening log file:  /gstore/home/ferlg/java.log.21512

                                                                                  < M A T L A B (R) >
                                                                        Copyright 1984-2021 The MathWorks, Inc.
                                                                   R2021a Update 3 (9.10.0.1684407) 64-bit (glnxa64)
                                                                                     May 27, 2021


To get started, type doc.
For product information, visit www.mathworks.com.

>> predictLungTumors


Detected 29 image files in folder "files_in_CTscans"

Performing image preprocessing...4.032079 minutes

Performing image coregistration...71.549929 minutes

Performing watershed segmentation...2.927589 minutes

Writing watershed objects to file...0.468160 minutes

Generating watershed object feature array...1.019190 minutes

Predicting watershed object tissue class...4.824518 minutes

============================
Entire analysis took 84.828159 minutes
============================
-----------------------------------------------------------------------------------------------------
MATLAB Version: 9.10.0.1684407 (R2021a) Update 3
Operating System: Linux 3.10.0-1160.66.1.el7.x86_64 #1 SMP Wed May 18 16:02:34 UTC 2022 x86_64
Java Version: Java 1.8.0_202-b08 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode
-----------------------------------------------------------------------------------------------------
============================
MATLAB toolboxes used for this analysis:
image_toolbox
matlab
statistics_toolbox
============================
>>
```

Figure S7. Matlab output for forward prediction of individual tumor ROIs, related to STAR Methods
Matlab terminal output shown for $n = 29$ hold-out test scans. Note that the watershed segmentation step is executed by the software package Analyze 12. All other steps are performed within MATLAB.

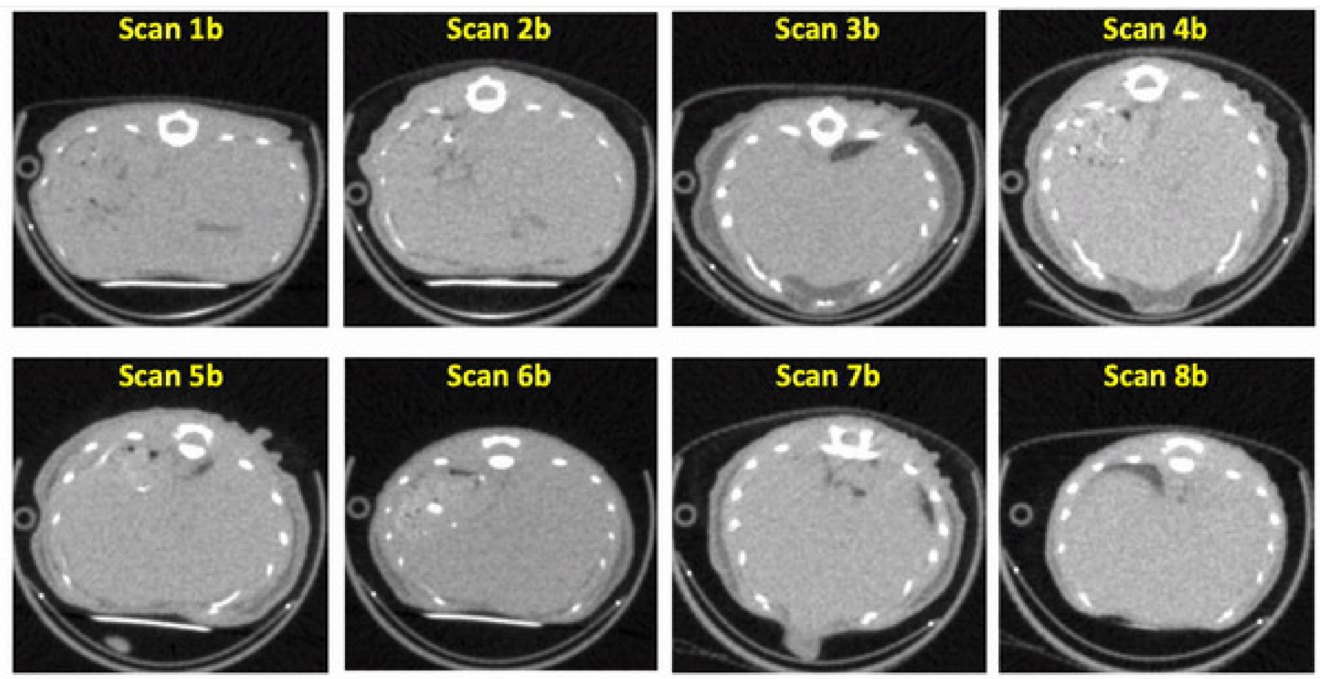

Figure S8. Lung ROIs for respiratory-gated scans acquired by the MILabs scanner, related to Figure 1
ROIs were predicted by the 3D U-net trained on $n = 8$ selected images acquired by the ct120 scanner. Click figure to animate.

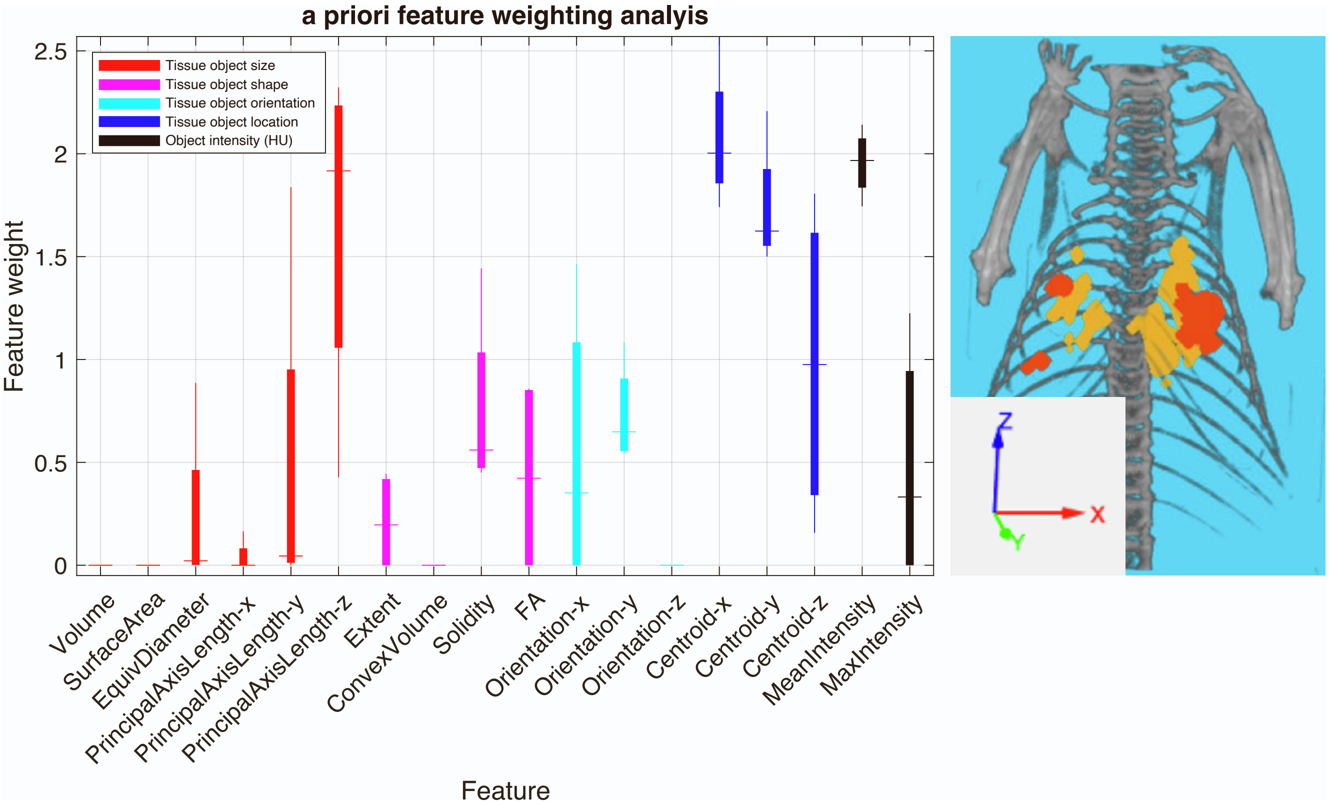

Figure S9. ***A priori*** **feature selection using neighborhood component analysis, related to Figure 3**
Analysis based on Matlab function *fscnca* (Copyright 2015-2016 The MathWorks, Inc.). Prior to training the classification algorithms summarized in Table S1, the full feature array (18 features, 1941 tissue objects) was analyzed to identify correlations between individual features and the three tissue classes (vessel, tumor, other). The 1941 tissue objects were split into 5 partitions of approximately equal size, for which feature weights were calculated. The boxplot shows median, 25% and 75% quantiles (boxes) and most extreme values (whiskers) of the calculated feature weights across the 5 partitions ($n = 5$). Most features have estimated weights greater than zero, with metrics of object size and location having the highest weights, where the principal axis length along the z-axis and object centroid on the x- and y-axis are strongly correlated to the tissue class of each object. Right-hand image shows image axes for a representative micro-CT scan where segmented blood vessels are shown in yellow and tumors in orange. The mean image intensity in Hounsfield units is also highly correlated to tissue class, but is also correlated with object size where larger objects will tend to have higher average image intensities. Generally, the features within each of the 5 classes shown in the boxplot legend will tend to be correlated to one another. All features analyzed here were included in the feature array used for classification algorithm training.

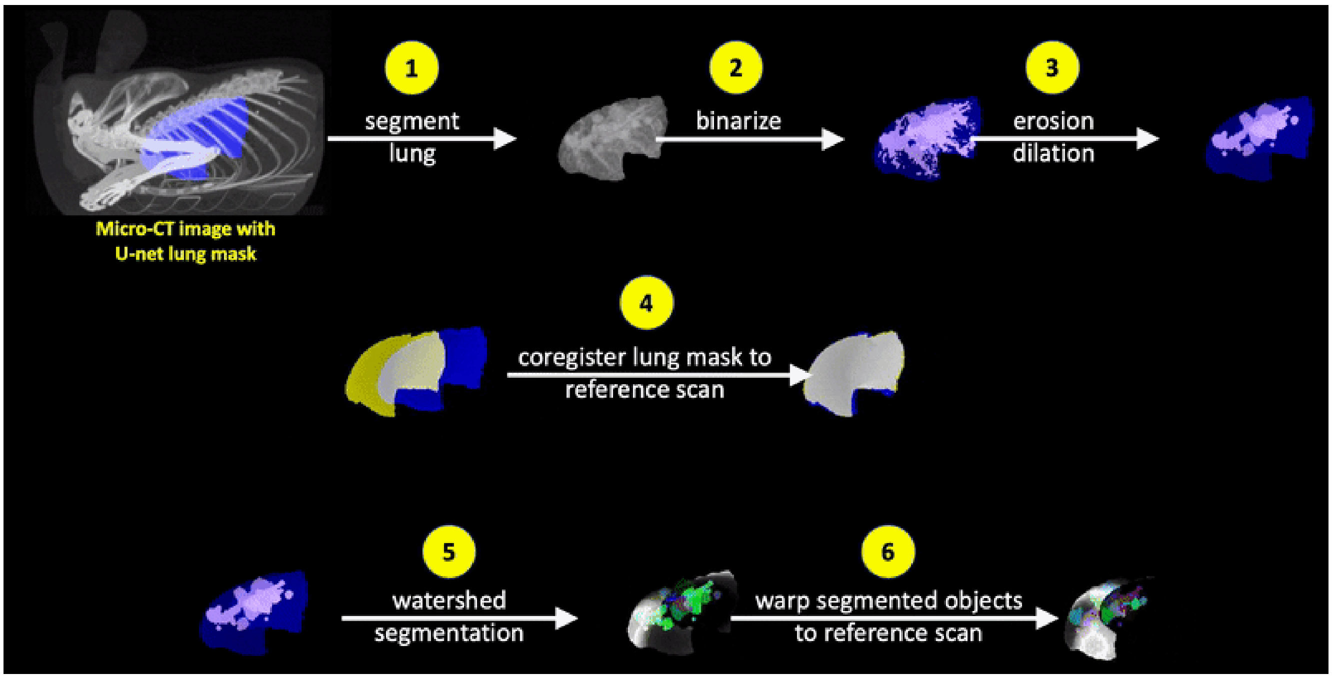

Figure S10. 3D image preprocessing steps for lung tumor segmentation, related to STAR Methods
The key image processing steps for the lung tumor segmentation algorithm are summarized here, where 1) the lungs are segmented using the region of interest estimated by the trained 3D U-net, the resulting lung CT image is 2) binarized and 3) small objects are removed via erosion/dilation operations before 4) coregistering the lung ROI to a reference scan lung ROI; 5) watershed segmentation is then applied to the image generated by step 3 and 6) the resulting objects are warped to the reference scan coordinates using the affine transformation parameters calculated in step 4. Click figure to animate.