cropLung(S)

parameter(s): S 🡪 x-ray path

output: Lung image inside the image(S)

Initialization:

get image(S)

find body image inside the image(S) 🡪 findBody(image(S))

find lung image inside the body image 🡪 findLung(body)

if the result returns

do save the lung image as new image.

findBody(S)

parameter(s): S 🡪 x-ray image

output: Body image inside the S

Initialization:

X,Y = x and y dimensions of S

O keep the original image

convert S to black and white (240, 255)

for i = 0 to Y

do for j = 0 to X

do If the S[i,j] are white, paint O[i,j] black.

T convert and keep O to black and white (50, 255)

blur T ((X/10), (Y/10))

convert T to black and white (100, 255)

for i = 0 to Y

for j = 0 to X

do XS Find the largest j value where T[j,i] == 255 (white)

do

for j = X to 0

do XE Find the smallest j value where T[j,i] == 255 (white)

for i = 0 to X

for j = 0 to Y

do YS Find the largest j value where T[j,i] == 255 (white)

do

for j = Y to 0

do YE Find the smallest j value where T[j,i] == 255 (white)

return O[YS:YE, XS:XE]

findLung(S)

parameter(s): S 🡪 body image

output: Lung image inside the S or S

Initialization:

search lung in S with related templates 🡪 findTemplate(S, ”fullLung”, 95)

if the result returns

do return lung image

search left lung in S with related templates 🡪 findTemplate(S, ”leftLung”, 95)

if the result returns

do search right lung in S with related templates 🡪 findTemplate(S, ”rightLung”, 95)

if the result returns

do return lung image

search top left lung in S with related templates 🡪 findTemplate(S, ”topleftLung”, 95)

if the result returns

do search bottom left lung in S with related templates 🡪 findTemplate(S, ”botLeftLung”, 95)

if the result returns

do search top right lung in S with related templates 🡪 findTemplate(S, ”topRightLung”, 95)

if the result returns

do search bot right lung in S with related templates 🡪 findTemplate(S, ”botRightLung”, 95)

if the result returns

do return lung image

return S

findTemplate(S, P, A)

parameter(s): S 🡪 body image

P 🡪 path of related object templates

A 🡪 minimum template match rate

output: coordinates of the object to be searched

Initialization:

convert S to RGB image

XS,YS = x and y dimensions of S

TAll fetch all templates from where P is

foreach T in TAll

XT,YT = get values of image size before template cutout from T name

compare XT with XS and YT with YS

if comparing rates are bigger than 1.1 or smaller than 0.9

do resize T by comparison ratio

R match S to T and keep results

do

for i = 100 to A

Get match rate as i value

if there is a match and the i value is higher than the highest match rate

do

do B save match coordinates as best match

save highest match rate value as i

return B