# README for patch generator from ILD DB

Sylvain Kritter May 2nd 2016

# Tool used

1. RadianDICOM viewer to view dcm files.
2. Can also convert Dicom images in jpg, bmp,
3. Total Image Converter to create jpeg, bpm and JPEG 2000 from dcm.

I have created in each ILD\_DB-textROIs dataset (35, 65,…) the directories

bmp,jp2k,jpg to store equivalent format from dcm files.

1. Spider Python 2.7 for Python 2.7

# To run the patch generator

1. 3 python files needed:
2. final.py (this is the top, parameters are defined in it, no need to touch the 2 others)
3. fillshape.py
4. generatetabc.py
5. Preparation: 2 python file needed
6. genima.py
7. renomsca.py

To run the patch generator, we need first to have the directory ILD\_DB-textROIs in the place where Python is launched.

In each dataset in directory ILD\_DB-textROIs, we need to add a directory named bmp or jpg to store bmp or jpg files generated from dcm

Bmp or jpg file are generated by TotalImageconverter, at top level keeping hierarchy.

Then genima.py is run to create the bmp or jpg directories and to put all the image files in bmp or jpeg directory.

For some scan, the scan are not following the number convention: scan number in the format <name>-<number in 4 digit>.dcm.

renomscan.py is adding this number at the end of the scan file.

1. Customization in final.py
2. Patch image format: bmp or jpg
3. Dicom file size: 512 \* 512
4. Patch size: 32\*32
5. Threshold in % : 0.8 for patch area over ROI

# program generate files

1. From where it is launched:
2. ‘Jpeg’ directory , where jpeg images and text files of all ROI with patches is stored
3. ‘patch’ directory, where patches are store in sub directories, named after label and localization names in each scan, according to image format declared in top python (jpg or bmp). I think bmp is more accurate.
4. in each ILD\_DB-textROIs dataset (35, 65,…) , a directory named ‘patchfile’ where intermediate data is stored. Can be deleted afterward.

# Database Analysis

1. The Dcm files are identical in ILD\_DB\_txtROIs, ILD\_DB\_volumeROIs and ILD\_DB\_lungMasks
2. A ROI text file is in each data set In ILD\_DB\_txtROIs. This Text file corresponds to ROI in ILD\_DB\_volumeROIs roi\_mask of each dataset.
3. There is a set of ROI in roi\_mask in each dataset of ILD\_DB\_lungMasks: this is lung area

# Scan preparation for prediction

File to run:

1. prepanscanmask.py : from scan image, generates bmp files according to patient database and lung mask. A patch is considered valid if the recovering area is above a threshold (by default 0.8, can be changed in the py file: variable “thr”.)
2. Format:

master directory name for bmp: patch\_test\_mask

sub directory: data set name (example 107)

Each patch has the format:

p\_"scan number"\_"xcoord upper left" \_"ycoord upper left".bmp   example: p\_00001\_32\_96.bmp

Jpeg files are also recorded in directory jpeg\_test\_mask, with dicom files and patches over.

1. prepanscan.py

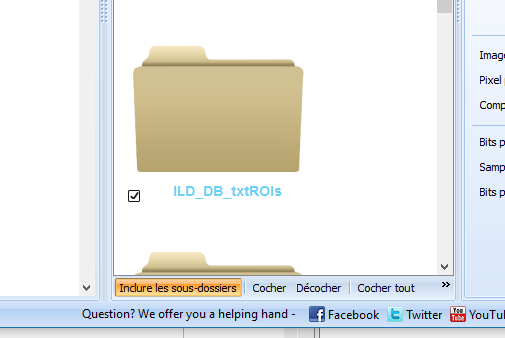
Generate patches, without lung mask: patches are generated over the full scan area

Same naming than above, but patch directory is patch\_test, jpeg directory jpeg\_test

# Global bmp file generation on a full patient database.

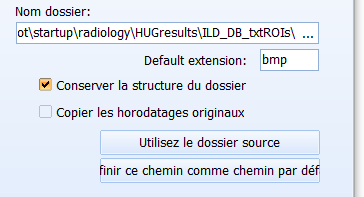
1. The tool Total Image Converter is run on directory top containing all the patient database.

Parameters should be: “inclure les sous dossiers”



Then, after clicking on BMP:

Activate “Conserver la structure des dossiers”



Then: launch bmp generation

1. File structure creation

genima.py file creates in proper place all the bmp directories and move bmp images of scan in them.

# List content of patient database

The program listlabel.py go through the patient database and put in a file named “listlabel.txt” the complete list of label and localization per dataset

# DCM Scan numbering

In some dataset, the scan are not numbered from 0 to n. The tool renomscan.py rename the .dcm file by appending a number starting from 1 in 4 digits (example -0021) just before the .dcm

# Calculate number of patches per label

The tool comptepatch.py run over the “patch” directory generated by final.py to calculate the number of patches per label and localization. Result is in file totalnbpat.txt

# Program to visualize the predict results

The program to visualize the predict results is visuapredict.py

1. Only prediction with probability above a threshold (default 0.8) are taken into account.
2. When several patches are recognized in a same scan, the image after visualization indicates the label with the average of probability (if above threshold)
3. I have attributed colors for ach label, used also for the name of labels as printed in the image.
4. The original patient database with dicom and bmp generated is in a directory at top level,

named : ILD/<patient number>/bmp

1. The patches in bmp extracted from patient database for prediction are in a directory at top level

named: predict/<patient number>

1. The result of prediction, are 2 pkl files named predicted\_classes.pkl and predicted\_probabilities.pkl

These files are in the directory named: predict\_out/<patient number>/predict

1. The visualisation tool generate a directory named: predict\_out/<patient number>/jpeg

and put the generated images in jpeg format

All can be changed, as per variables at top of py program.