Assignment 5 – Feature vectors and dimensionality reduction

This assignment will count towards your grade.

Summary

In this assignment, you will create feature vectors representing imaging datasets into other domains. This is the cornerstone of modern pattern recognition pipeline based on machine learning. You will also experiment the appropriate technique to represent data in a lower dimensional space.

Introduction

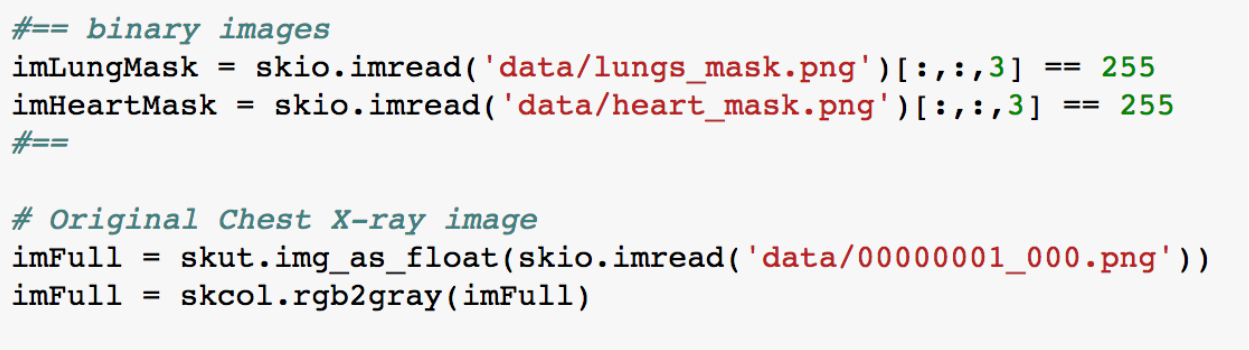
The following assignment needs to be presented as a Jupyter notebook. However, it is suggested to write as much as possible of code as an external module and use Jupyter notebooks for visualization and code description only. This will allow you to easily debug the code. A zip file containing the data used in this assignment is available on the web.

We will be using the following shorthand for the libraries:

```
import assign05 as a5
import matplotlib.pyplot as plt

import numpy as np
import skimage as sk
import skimage.io as skio
import skimage.transform as sktr
import skimage.filters as skfl
import skimage.feature as skft
import skimage.color as skcol
import skimage.exposure as skexp
import skimage.morphology as skmr
import skimage.util as skut
import skimage.measure as skme


import sklearn.decomposition as le_de
import sklearn.discriminant_analysis as le_di
import sklearn.preprocessing as le_pr
```

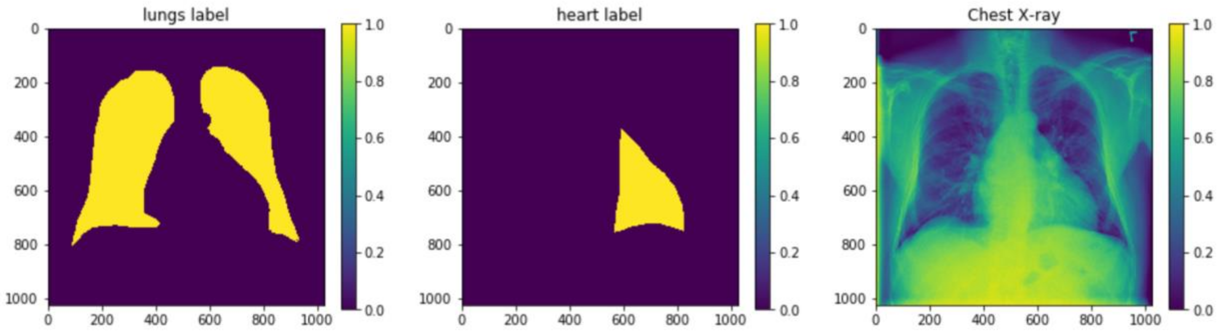
LBP Feature Generation (3 points)

- We will load a chest X-ray image and two binary maps representing heart and lungs.

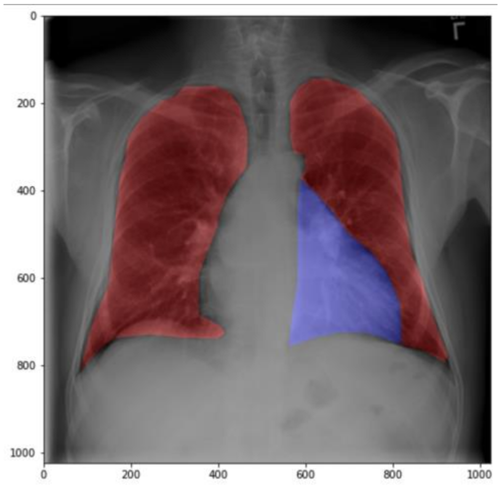


```
#== binary images
imLungMask = skio.imread('data/lungs_mask.png')[:,:,3] == 255
imHeartMask = skio.imread('data/heart_mask.png')[:,:,3] == 255
#==

# Original Chest X-ray image
imFull = skut.img_as_float(skio.imread('data/00000001_000.png'))
imFull = skcol.rgb2gray(imFull)
```

- Plot the images

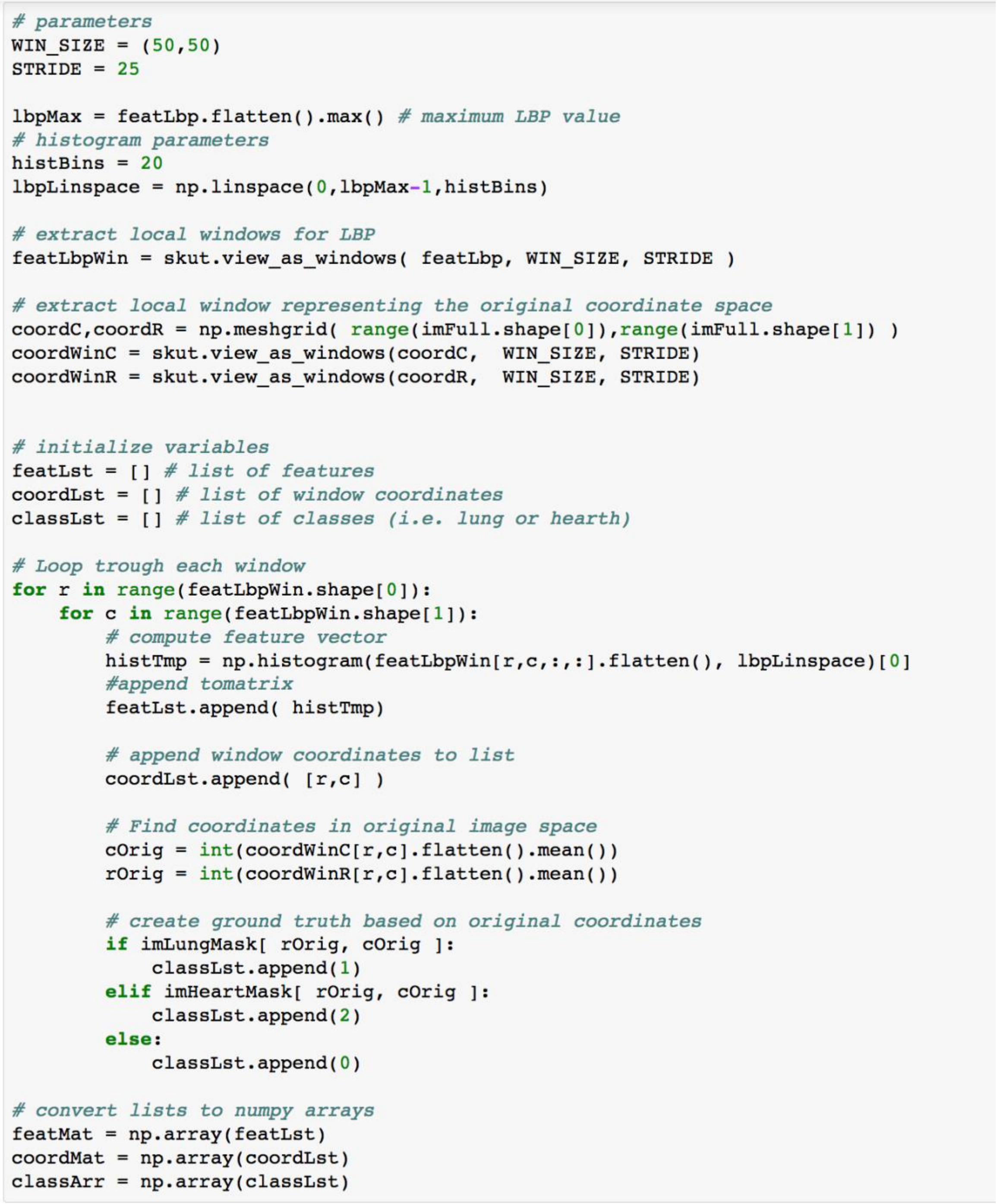


- Overimpose the labels onto the chest X-rays using skcol.label2rgb



- Compute the local binary patterns for the whole image (radius 3 and 15 points for the circular neighborhood). Check the skft.local_binary_pattern function. Store the result in featLbp
- Compute local features:
  - Split the the featLbp figure into local windows of size 50x50 (stride 25).

  - For each window compute the local histogram (20 equally spaced bins, going from 0 to the maximum of featLbp -1). This will be our feature vector.
  - Store each feature vector into a feature matrix featMat (dimensionality: samples x features)
  - Create a numerical array (classArr) representing the ground truth for each window. The array will be coded as follows:
    - 1: if the center of the window belongs to the lungs
    - 2: if the center of the window belongs to the heart
    - 0: anything else
- In the next page, you have a code snippet to compute exactly what described above in the point above (i.e. split the image in local widows, compute the feature vectors and ground truth). You can either copy it into your code or try to perform the same operation in alternative ways (maybe employing a manual for loop to slice the windows or using parallelism with skimage.util.apply_parallel).

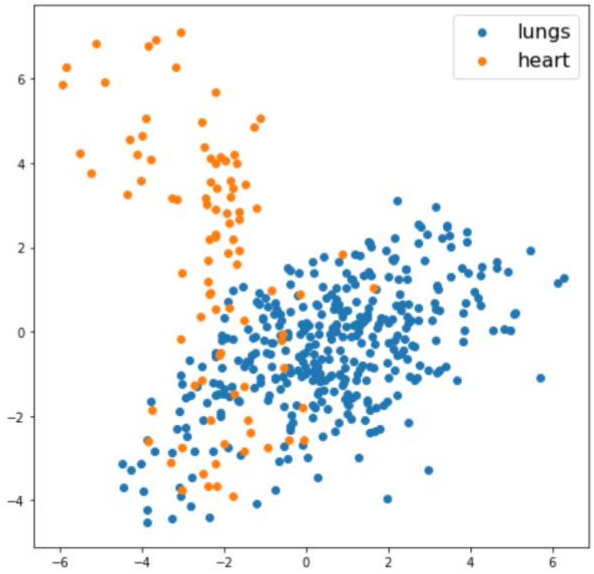


```
# parameters
WIN_SIZE = (50,50)
STRIDE = 25

lbpMax = featLbp.flatten().max() # maximum LBP value
# histogram parameters
histBins = 20
lbpLinspace = np.linspace(0,lbpMax-1,histBins)

# extract local windows for LBP
featLbpWin = skut.view_as_windows( featLbp, WIN_SIZE, STRIDE )

# extract local window representing the original coordinate space
coordC,coordR = np.meshgrid( range(imFull.shape[0]),range(imFull.shape[1]) )
coordWinC = skut.view_as_windows(coordC,  WIN_SIZE, STRIDE)
coordWinR = skut.view_as_windows(coordR,  WIN_SIZE, STRIDE)


# initialize variables
featLst = [] # list of features
coordLst = [] # list of window coordinates
classLst = [] # list of classes (i.e. lung or hearth)

# Loop trough each window
for r in range(featLbpWin.shape[0]):
    for c in range(featLbpWin.shape[1]):
        # compute feature vector
        histTmp = np.histogram(featLbpWin[r,c,:,:].flatten(), lbpLinspace)[0]
        #append tomatrix
        featLst.append( histTmp)

        # append window coordinates to list
        coordLst.append( [r,c] )

        # Find coordinates in original image space
        cOrig = int(coordWinC[r,c].flatten().mean())
        rOrig = int(coordWinR[r,c].flatten().mean())

        # create ground truth based on original coordinates
        if imLungMask[ rOrig, cOrig ]:
            classLst.append(1)
        elif imHeartMask[ rOrig, cOrig ]:
            classLst.append(2)
        else:
            classLst.append(0)

# convert lists to numpy arrays
featMat = np.array(featLst)
coordMat = np.array(coordLst)
classArr = np.array(classLst)
```

Principal Component Analysis (PCA) (4 points)

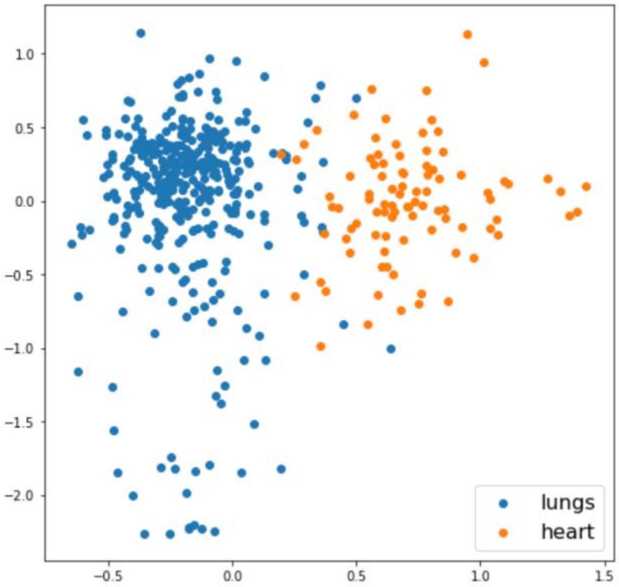
- Filter all feature vectors not belonging to heart or lungs. In other words, we will be analyzing only windows sampled over the lungs or heart.
- Scale the feature matrix such that each feature has 0 mean and 1 as standard deviation. (you can do it manually or using le_pr.StandardScaler).

- Compute PCA and project each feature vector to the first two components (you can use manual eigen decomposition in numpy or the integrated function in scikit-learn).
  1 bonus point if you compute PCA without employing scikit-learn
- Plot the result with the samples labelled according to their class. Do not worry if your resulting image is flipped.

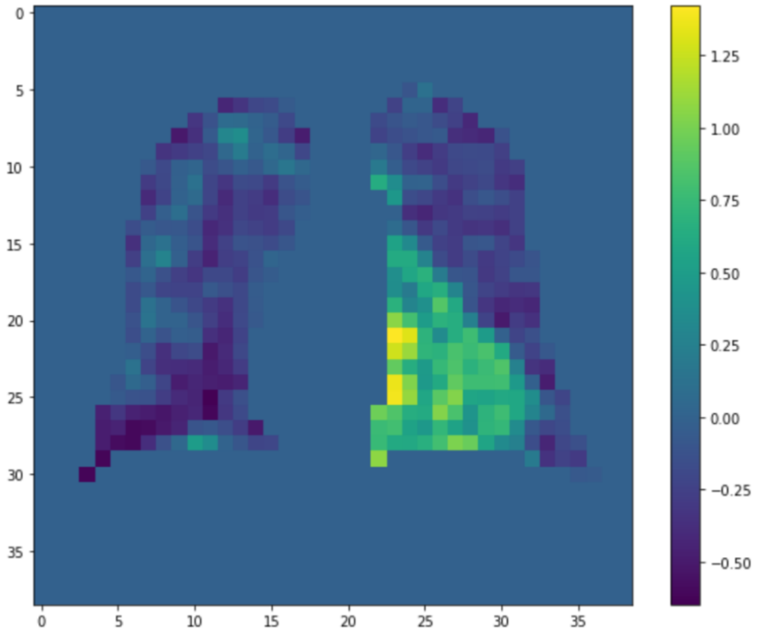


Linear Discriminant Analysis and Visualization (3 points)

- Use Local Discriminative Analysis (LDA) to project each feature vector and their class to a 2-dimensional space. You can use a5.lda function provided in assign05.py



- Plot the first LDA component as an image. You can use the coordinates of the local windows (stored in coordMat in the example code provided at the beginning).



- Note how the heart correspond to higher values and the lungs to lower. You could also get the exact opposite! (i.e. low values for the heart and high values for lungs), there is nothing wrong with that.

Submission

- Now zip the folder containing your code and the notebook file as you did in assignment 01
- Upload yourname-assignment05-nb.html and yourname-assignment05-src.zip to canvas before the deadline