Homework 4

Hengtao Guo

February 19, 2019

https://github.com/Tonight1121/Biology-Image-Analysis

1 Co-occurrence Matrix

Co-occurrence matrix is used to describe one pixel's texture from one image. The skimage.features library contains the greycomatrix() function to calculate the grey-level co-occurrence matrix:

greycomatrix(image, distances, angles, levels=None, symmetric=False, normed=False)

Where distances is the list of pixel pair distance offsets, angles is the list of pixel pair angles in radians and level is basically [0, 255] aka 256.

The five different textures in the ultrasound cardiac image are: right ventricular, left ventricular, right atrium, left atrium and cardiac walls, as pointed in Fig. 1. The corresponding textures descriptions can be seen in Fig. 2.

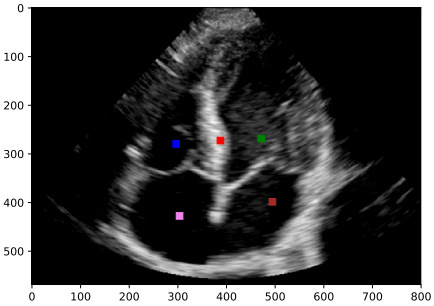

Figure 1: Points on different structures.

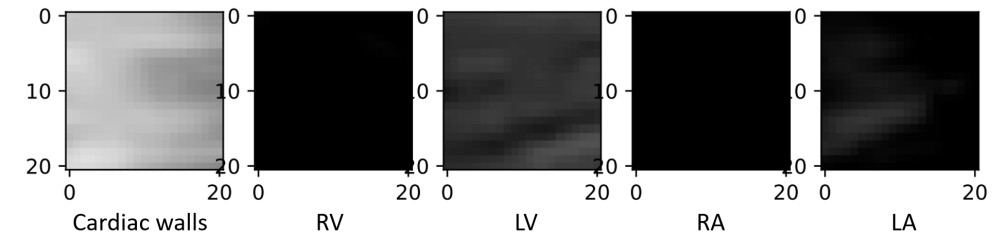

Figure 2: Textures of five structures.

2 K-Means

We basically want to segment the raw image into 6 parts: right ventricular, left ventricular, right atrium, left atrium, cardiac walls and other background, aka 6 clusters.

To apply K-Means as well as Support Vector Machine (SVM) and random forest algorithms, we have to design a feature vector for each pixel to identically describe its characteristics. Some of the features coming to my mind are:

1. The grey scale intensity of each pixel;
2. The average grey scale intensity of one pixel's $n \times n$ neighbourhood;
3. The standard deviation of grey scale intensity of one pixel's $n \times n$ neighbourhood;
4. The (x, y) coordinates of each pixel with or without scaling.

2.1 Pre-processing

For automatic clustering algorithms like K-Means, I have to use np.reshape to directly form the output mask with the same size of input. However, the fan-shaped scan is quit troublesome during the progress, so I just manually cropped out a rectangular region out of the meaningless background as shown in Fig. 3. I used this region only for the K-Means classification. Using a fan-shaped mask can also tackle this problem, but that is not the priority today.

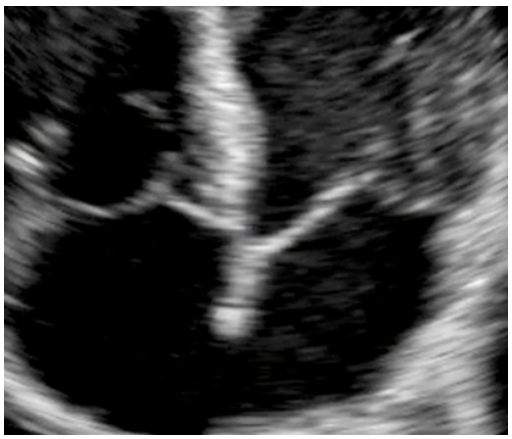

Figure 3: Cropped patch

2.2 Intensity with AVG & STD

From my own understanding, when we combine simple features such as grey scale intensity value, neighbourhood average intensity and standard deviation, we can roughly describe the texture information of one pixel. In such way, each pixel will have a feature vector of only three values: pixel intensity, the average intensity of its 9×9 neighbourhood, and the standard deviation of intensity of its 9×9 neighbourhood. K-Means results are shown in Fig. 4

At this moment, we can see that different components of heart have been roughly highlighted in different random colors. As the RV, RA and LA are acutally similar in texture, they are more or less clustered into the same group. On the other hand, cardiac walls and LV have been successfully distinguished from the rest.

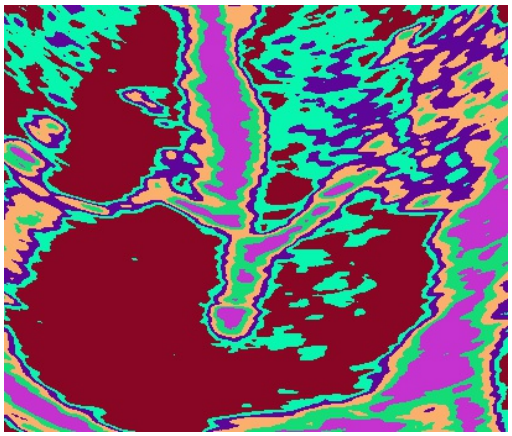

Figure 4: K-Means texture

2.3 Coordinates with/without Scaling

The motivation of using coordinates as features is to maintain the spatial location relationship of neighbourhood tissues. Tissues are more likely to be segmented together if they are close.

Using coordinates without scaling produces the results in Fig. 5. As it can be seen, the result is quite obvious for each components. As the location of each section is spatially ordered, the segmentation mask appears very convincing. Even the cardiac walls in the middle of four heart sections is segmented out.

The scaling procedure is to divide the coordinates by a ratio (I set 5 here) to make the pixels which are far away from each other can still be grouped together. Here Fig. 6 we can see the cardiac walls far from each other can also be grouped into one class.

Figure 5: With unscaled coordinates

3 SVM Classification

3.1 Preparation

First, I manually selected some patches representing right ventricular, left ventricular, right atrium, left atrium and cardiac walls. The mask can be seen in Fig. 7.

To be more specific, I only cropped a small portion of each actual region. As I only have one image available, I have to perform both training and testing on this exact image. The labeled regions

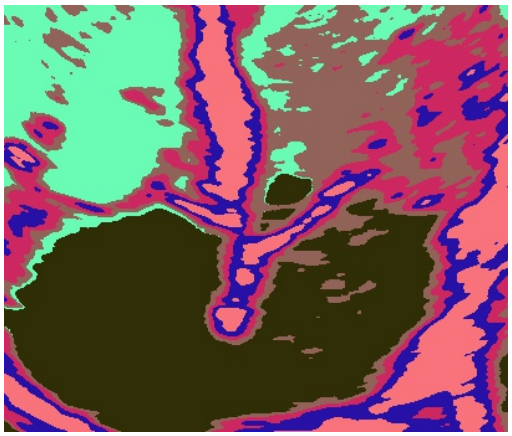

Figure 6: With scaled coordinates

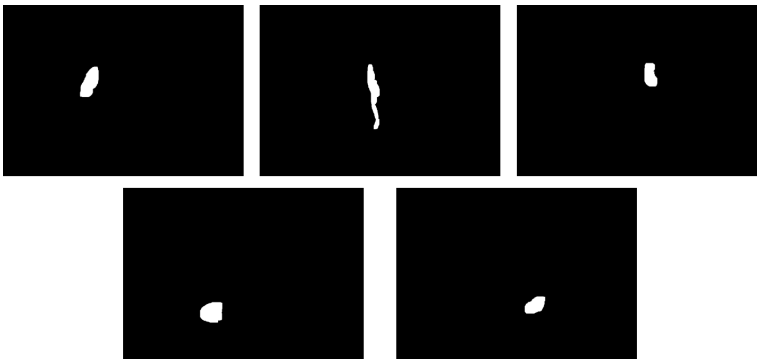

Figure 7: Some selected label patches. In the order of right ventricular, left ventricular, cardiac walls, right atrium and left atrium.

are used for training, and rest of the regions are for validation.

3.2 Results

The feature vectors I form for each pixel is the same as discussed in K-Means: pixel intensity, the average intensity of its 9×9 neighbourhood, and the standard deviation of intensity of its 9×9 neighbourhood, x and y coordinates with/without scaling.

The sklearn.SVC is capable of classifying multi-class targets in one time, so I can just label all five target region feature vectors as 0, 1, 2, 3 and 4. The labeled feature vectors form the entire training dataset, and just us them to train the SVM model to distinguish from each other. After the SVM is trained, this model is applied to the whole image feature vectors to evaluate the performance. The SVM model can perform 100% accuracy on the training dataset.

In this case I used a mask to limit the scope of the entire image as Fig. 8. The result of SVM classifier can be seen in Fig. 9. As it can be seen , the result is quite descent from my observation. Considering the actual background is quite complex and may mix with cardiac walls, I did not additionally crop the mask of 'background'. But once provided with enough training samples, the presentation of background can be learnt in the same way. The result of SVM is already satisfying, so no need to dig into the effects of each features.

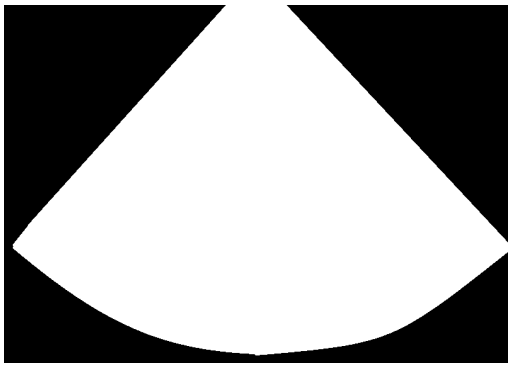

Figure 8: Fan mask.

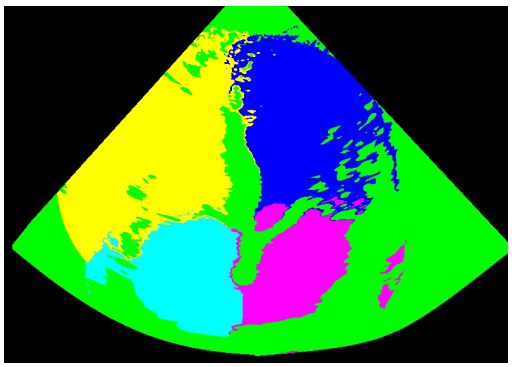

Figure 9: **SVM results. All features with scaling coordinates are used.**

4 Random Forest Classification

4.1 Results

The results of random forest are not as good as SVM's. Here are the listed results.

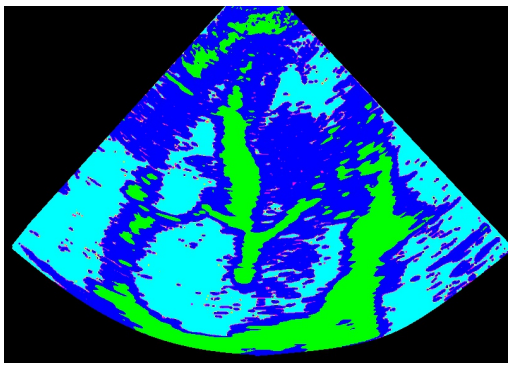

Figure 10: Only use intensity, neighbourhood average intensity and neighbourhood intensity standard deviation.

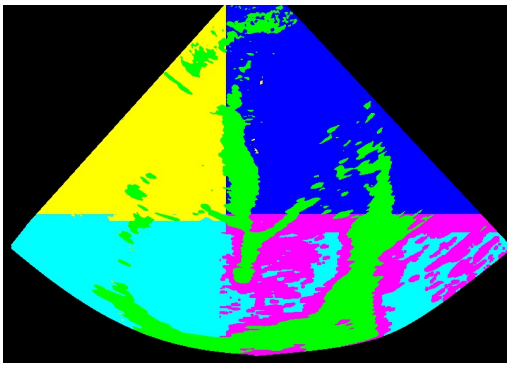

Figure 11: Adding unscaled coordinates as features.

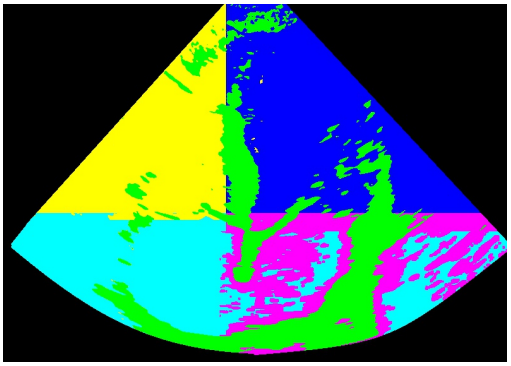

Figure 12: Adding scaled coordinates as features. Barely exists difference with former.