Toolbox 3: Image and Patter Recognition

Project 2: Automatic classification of skin lesion

(Melanoma detection)

Rida LEFDALI

The aim of this project is to propose a methodology to automatically classify the skin lesions, based on image analysis and machine learning.

1 Data set:

Our dataset is composed from 200 samples, for each sample, there is three kinds of images: the original image, the segmented image and the superpixels image. Each kind will be used to extract specific features to build our classifier with high accuracy. Also, we dispose a csv file which contains the label (the class) of every sample.

Superpixels image | Binary segmented image | original image

Figure 1: A sample of the dataset.

2 Features extraction:

2.1 Extraction of geometrical descriptors:

As we did in the previous project, we are going to extract geometrical features from the binary segmented images. To extract these features, we will use the function ***regionprops_table()*** from the library ***skimage***.

```
label_img = label(B) # B : binary segmented image
props = regionprops_table(label_img, properties=['area','convex_area','bbox_area','eccentricity','extent','filled_area'\
                                ,'minor_axis_length','major_axis_length','solidity','perimeter','equivalent_diameter'])
```

The function ***extract_prop(B)*** has been coded to return these features in an ***np.array***.

2.2 Extraction of texture descriptors:

To extract these descriptors, we are going to use ***the Local Binary Pattern (LBP)*** which is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

we are going to extract the **LBP** histogram for the red component of each original image in the data set. So, to compute the **LBP** histogram, we will use the code done from the *practical work 13* in the *UP: computational geometry.*

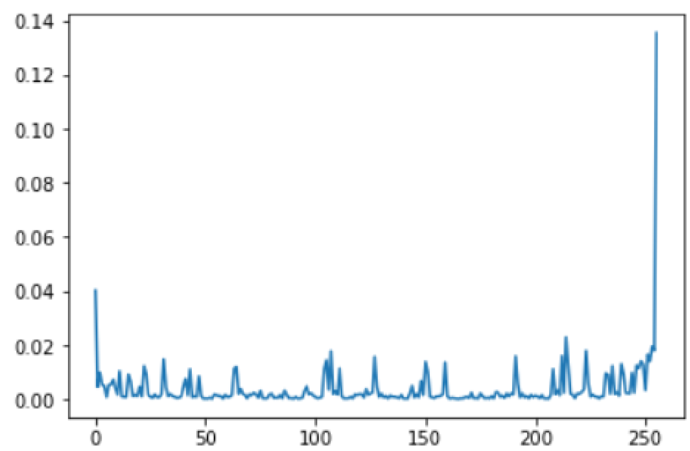

Figure 2: The LBP of the blue component of a tumor not malignant

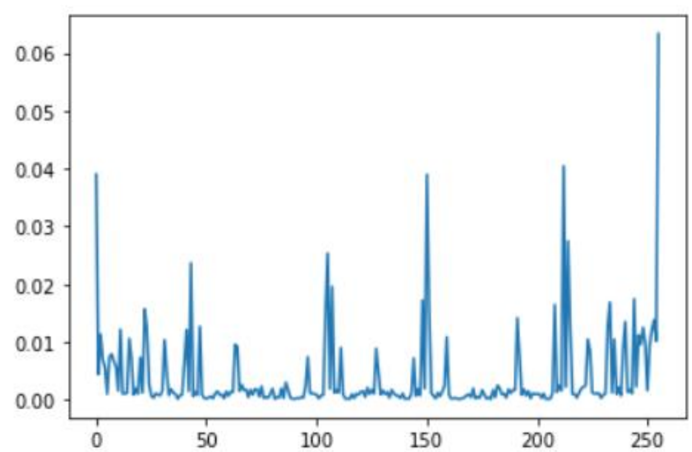

Figure 3: The LBP of the blue component of a malignant tumor

2.3 Extraction of other features:

According to some research papers, color variegation, which is the number of color present in a tumor, is a feature that can be extracted from the original images. Normal lesion consists of a uniform color. Usually they are characterized by brown, black etc. But one important sign of a lesion to be melanoma is color variation in the lesion. Malignant melanomas contain three or more types of colors. So, we are going to compute the number of color present in the lesion. For each color we compute the number of pixels that have this color and if it is great than **5%** of total pixels in the lesion, then the color is assumed to be present. In our case, we are going to compute then the presence of six colors which are: white, light brown, dark brown, red, blue gray and black.

The code of the function that extract this feature:

```
def color_variegation(I):
    I = I/255
    m, n, k = I.shape
    white, black, red, light_brown, dark_brown, blue_gray = 0, 0, 0, 0, 0, 0
    for i in range(m):
        for j in range(n):
            R, G, B = I[i,j,0],I[i,j,1],I[i,j,2]
            if (R > 0.8 and G > 0.8 and B > 0.8) :
                white +=1
            if (R >= 0.588 and G < 0.2 and B < 0.2) :
                red +=1
            if ((R >= 0.588 and R <= 0.94) and (G > 0.2 and G <= 0.588) and (B > 0 and B < 0.392)) :
                light_brown +=1
            if ((R > 0.243 and R < 0.56) and G < 0.392 and (B > 0 and B < 0.392)) :
                dark_brown +=1
            if (R <= 0.588 and (G >= 0.392 and G <= 0.588) and (B <= 0.588 and B >= 0.490)) :
                blue_gray +=1
            if (R <= 0.243 and G <= 0.243 and B <= 0.243) :
                black +=1
    number_pixels = m*n
    colors = [white,red,light_brown,dark_brown,blue_gray,black]
    C = 0
    for i in range(6):
        if colors[i] > 0.05*number_pixels:
            C += 1
    return C
```

To get an exact value of this feature, it is preferable to work on images that contain just the tumor. So, I coded the function **get_tumor (I, B)** which combine the binary segmented image and the original image to return an RGB image contains just the part that interest us.

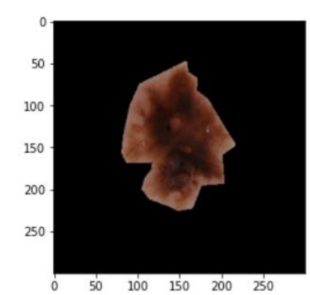

Figure 4: the original image combined with the segmented binary image

Another feature is computed is the ***number of superpixels*** of the given superpixels images.

Finally, we combine all these features in one matrix. The next step will be the classification, we will try many classifiers and choose the best one based on the common metrics, accuracy, **Fscore** and **ROC** curve.

3 Classification:

All the features have been extracted and our data is ready to build classifiers. We are going to use the following classifiers: *Logistic Regression, naïve bayes, SVM, Decision Trees and stochastic gradient descent*.

Before start building our classifiers, we are going to split our data randomly into training set and test set (75% for training and 25% for test)

At first, we trained our models just with the texture descriptors **(LBP histogram)** and we have got the following results:

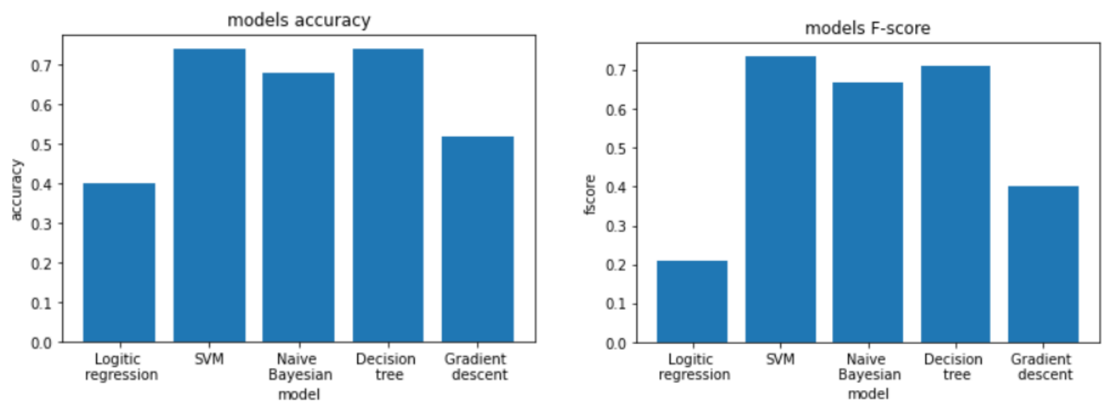

Figure 5: The performance of the classifiers

We see that the best classifiers are the **SVM** and **decision trees**, both of these classifiers has an *accuracy* equal to 0.74 but the **SVM** has an *Fscore* equal to 0.73 which higher than the *Fscore* of Decision tree. So, we can conclude that the best model is the SVM.

Until Now, we have a classifier that work relatively well with an accuracy not bad, but we will try to train also with other features to see if the accuracy will increase or not.

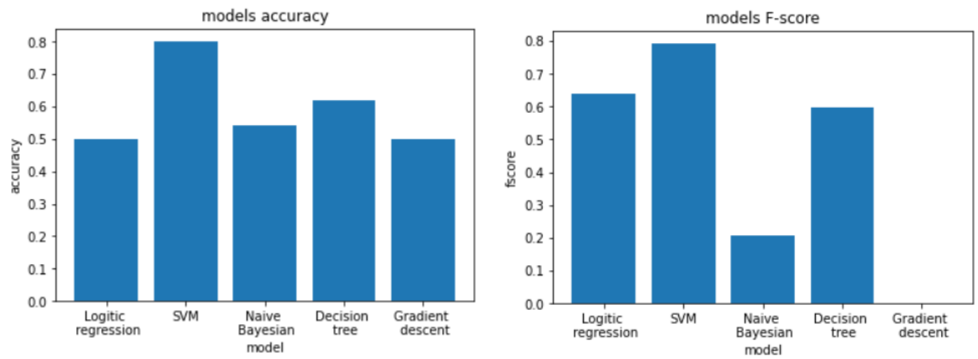

Figure 6: The performance of the classifiers

We see that when we used all the features the SVM classifier has been improved and we get an accuracy = 0.8 and Fscore = 0.79 which is quite well.

4 Conclusion:

To sum up, to build a good model to classify skin lesion images, it is important to extract from the images the features that describe very well the data. For our case we have extracted the texture descriptors, the geometrical descriptors and other features such as the number of super pixels and the color variegation. We have tried many classifiers such as logestic regression, naïve bayesian, stochastic gradient descent, SVM and decision tree. We found that the best classfier is SVM with an accuracy equal to 0.8.

We see that all the methodologies tried requires to extract many features from the segmented binary images, the superpixels images and the original ones. But, if we have just the original images it will be very complicated to apply the previous methodologies. So, the solution is to use the convolutional neural network, which is a strong tool used to classify images. The number of samples that we have is not enough, so we could have problems to get a good convolutional neural network.