**Statistical analysis of images to detect skin lesions and applied topical medicine**

**Introduction**

Mainly from manuscript

**2.0 Methods**

**2.1 generating manual classification set**

*2.1.1 finding roi*

*2.1.2 classifying roi*

**2.2 generating model**

*2.2.1 feature detectors and descriptors*

*2.2.2 generating bag of words*

2.2.2.1 binary descriptors

*2.2.3 generating histograms*

*2.2.4 testing detectors/descriptors*

*2.2.5 evaluating with tsne*

**2.3 cross validation**

*2.3.1 Bayesian optimization*

*2.3.2 training cheap model (k nearest neighbor)*

*2.3.3 error/cost function*

*2.3.4 with hyperparameters generate svm and or random forest*

*2.3.3 running on test set*

**2.4 implementing in iOS**

**Generating training set**

To recognize skin lesions with applied medicine, supervised machine learning algorithms can be used. Supervised means that these algorithms are trained on a predefined dataset. In our case this dataset should contain images of skin lesions with and without topical medicine applied. Since the differences between these two states will be slight, it is of importance to have a negative training set of skin lesions without medication.

We generated the training set using image analysis to fasten the process. First the image is segmented, in which the image is divided in regions based on similarity. Mathematically this is the same as clustering in a matrix. After the segmentation the segment containing a finger is recognized by averaging the color in the segment and looking for skin colors. Using the location of this segment the finger in the original image can be found. This finger is further analyzed on contrast and texture differences to detect a possible region of the skin lesion.

The region of interests the above-explained method will find are manually classified for containing warts and possible applied medication. This classified set of images will be used to train a supervised machine learning algorithm.

**0. Image representation**

To analyze an image, it needs to be represented in a mathematical way. This is most often done in matrices, but images can also be represented in other representations such as frequencies and wavelets. Matrices are useful since mathematical clustering algorithms can be used on them. A grayscale image of 20x20 pixels will have a matrix representation with 20 rows and 20 columns. The amount of gray on an individual pixel can be represented between 0-255. This means that at a certain value v in the matrix at row y and column x will be the grayscale value of the pixel at location x,y in the image. For color images multi-dimensional matrices are used. Each color channel (red, green or blue) will have its own dimension.

**1. Image segmentation**

We wanted to analyze the image to in a nonparametric manner, where the analysis makes no assumption about the underlying distribution. This has the advantage that in a nonparametric analysis a predefined number of clusters is not needed. Only a few algorithms could be used for image segmentation. We tested the watershed and mean shift algorithm.

* 1. **Watershed algorithm**

...

* 1. **Mean-shift algorithm**

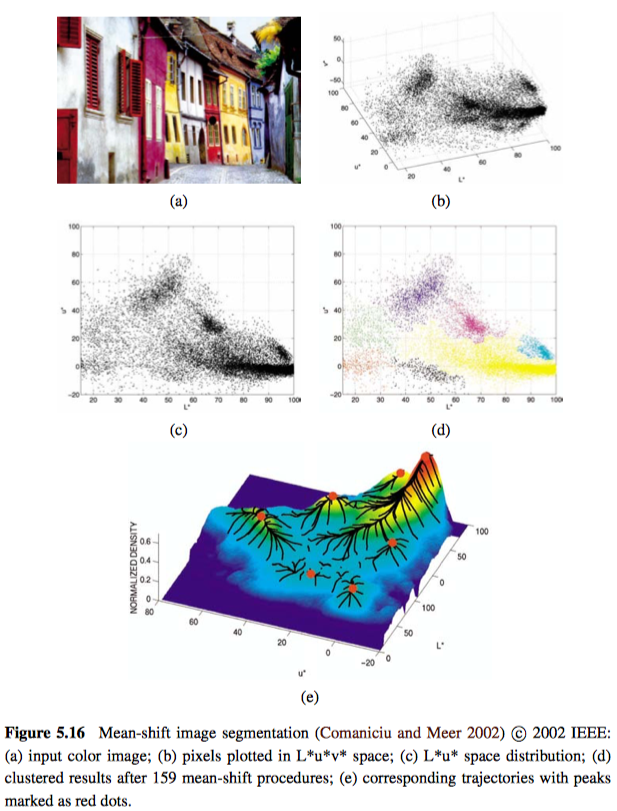
For the mean-shift algorithm the image representation is a multi-dimensional matrix by discarding spatial information and only looking at color. This resulting representation is called the *feature space*. Mean shift considers the feature space to consist of a probability density function. The probability density function is a function in which the area under the curve describes the probability of a certain value range giving the supplied data points. For example, a well-known *probability density function* (p.d.f.) is the normal distribution.

The mean shift algorithm uses a sample of points (the window) in the feature space as a sample of a pdf, calculates the pdf on these discrete values with a supplied kernel and calculates the mean shift () this pdf. This *mean shift* is calculated by the weighted mean of the change of the p.d.f.: by summing the derivative’s value at every discrete of the point in the window and dividing it by the weights provided by the kernel. Another way of looking at it is calculating the direction of the p.d.f. at a distinct point; the direction towards the mean can be derived from the derivative. This is done equal to creating a ‘gradient’ vector, a vector of the partial derivatives (the derivate considering the different variables).

The window in the feature space (containing the analyzed sample points) of the algorithm then moves by this calculated direction mean shift, thereby moving the denser area of the p.d.f. This continues until a finite number of steps or after the shift is below a certain threshold, which mean a dense region is reached (local maximum of the p.d.f.)

Applied on an image, the ultimate value of each pixel is set to the mean of the underlying probability density function. Instead of using this algorithm on every pixel (value in matrix), random location samples can be taken, the paths of these mean-shifts toward the mean are tracked. If in the end there are pixels not contained in these paths they can be classified using the nearest path.

The segmentation can then be based on the spatial information and different hues of colors in the picture with a certain threshold of difference in hue to define a segment.



Source: Szeliski R. **Computer Vision: Algorithms and Applications**, 2010

<http://imagelab.ing.unimore.it/imagelab/pubblicazioni/2013ElectronicImaging.pdf>