User manual

Table of Contents

1. Introduction **2**

2. Detection view **2**

3. Evaluator view **2**

4. Color model training view **3**

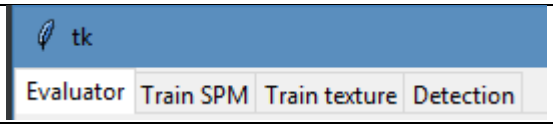
5. Texture model training view **4**

6. Common configuration frames **4**

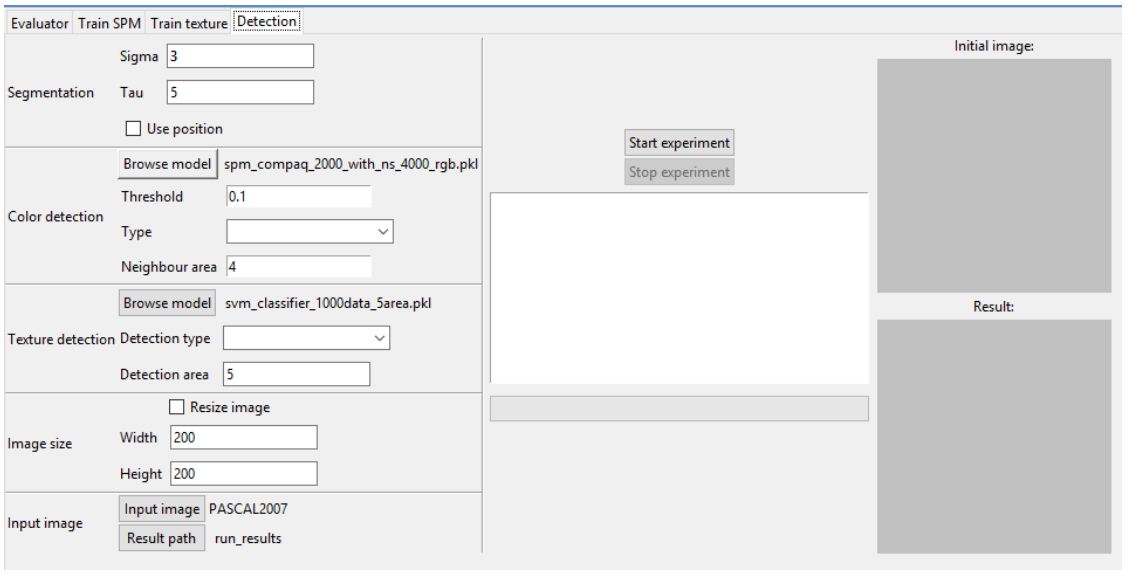
7. Feedback frame **5**

1. Introduction

The goal of this application is to provide an easy-to-use and highly-configurable interface for training and evaluating skin detection models that are share a core structure of image segmentation, color and texture analysis. While the number of parameters for each task could be overwhelming the user can opt to use the default values that have been already set in each field.

The application has 4 main windows accessible through a tab-view set in the top-left corner.	

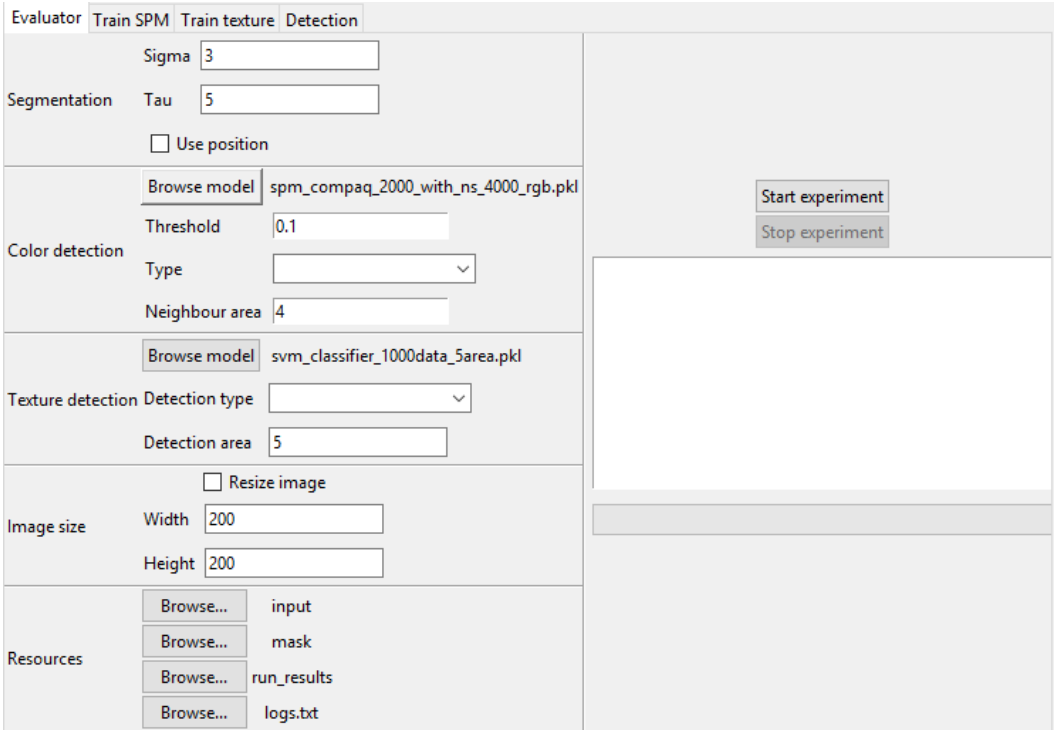
2. Detection View



If you want to apply the model to a single image you must specify the detection parameters, or go with the default ones, provide a path to the input image (bottom frame on the left side of the screen) and click 'Start experiment'. The initial image and the result will be displayed on the frames on the right side of the screen. The other configuration frames are explained in detail in section 6. On the middle of the screen, below the buttons for controlling the experiment, you will see updates of the detection process.

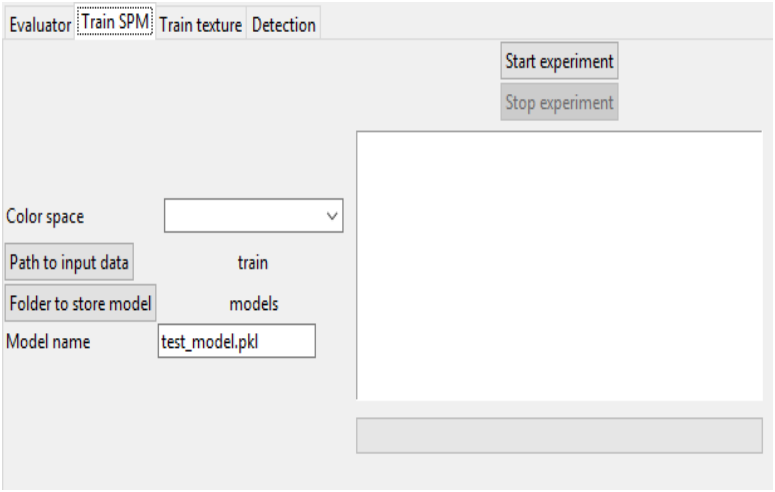
3. Evaluator view

This screen allows the user to evaluate the performance of the final model and follow the progress in real time. The selected model can be configured in various ways which will be presented in detail in the common configurations section (section 6).



You should provide paths for the input images, masks for those images and location where to store results and logs. To begin evaluation you must press 'Start experiment'. The frame below the button displays information about the currently processed picture, the operation in progress and evaluation results.

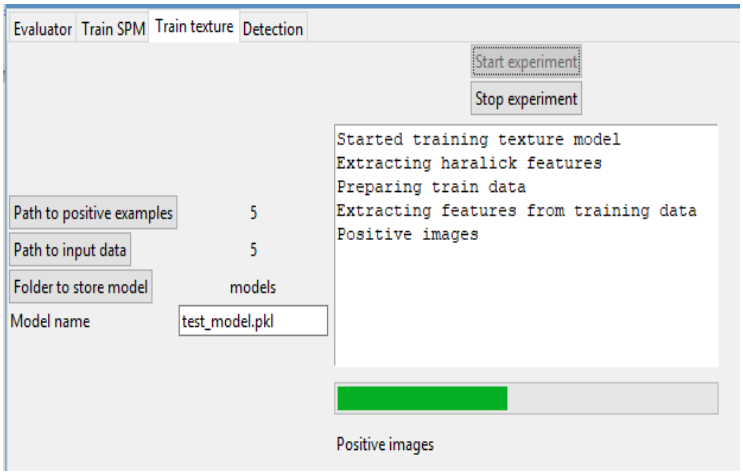
4. Color model training view



This window is used to train a color detection model. You can select a color space, the path to input data (a folder which should contain the following subfolders: train_images – images containing skin, train_images_ns – images not containing skin, train_masks – masks for images containing skin), and a folder in which to store the resulting model and its name.

To start training the model you must press the 'Start experiment' button. In the frame below you will see status updates for loading images and extracting features from them as well as a progress bar for each operation.

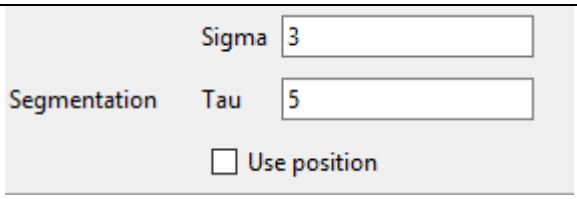
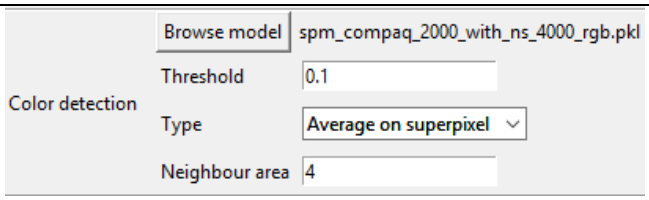
5. Texture model training view

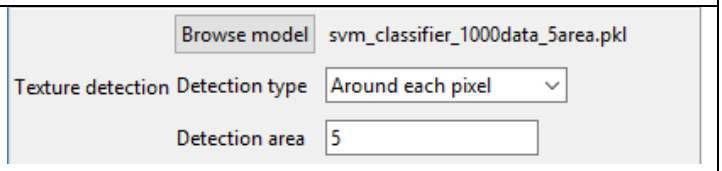
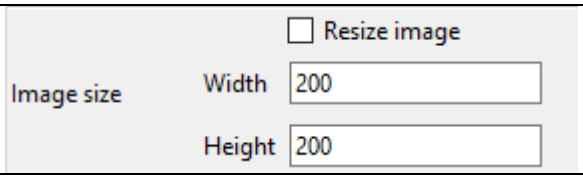


The process of training a texture detector is simple. You must select folders with images of skin, images not containing skin and a path to store the resulting model.

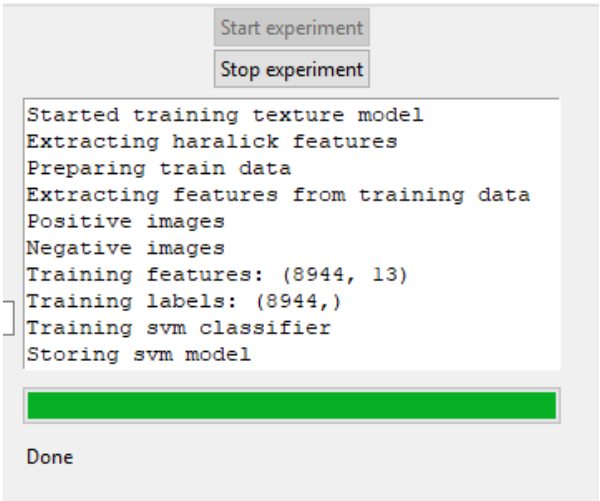
After pressing 'Start experiment' the training begins and you will see updates for feature extraction and SVM fitting.

6. Common configuration frames

This window is used for tuning the parameters of segmentation. Sigma controls the shape of the superpixels and Tau the size, and you can set them by typing in their adjacent textboxes. The 'Use position' checkbox indicates whether the algorithm should consider pixel position in image as a feature for density calculation.	
You have the following configurations for the color detection module: • the SPM model (serialized as a pickle file) • the detection Threshold, the type (average probability over superpixel or probability for the root of the superpixel) • Neighbor area (the neighbors considered when calculating the	

probability of a pixel).	
By clicking on the 'Browse model' button you can select the texture analysis model, which is a SVM classifier stored as a pickle file. From the 'Detection type' dropdown you can select the texture detection strategy: • Window scan, which slides a window over the image, is fast but produces jagged edges • Around each pixel, which builds a window around each pixel, is slow but obtains smoother results The size of these windows is set from the 'Detection area' text field.	
If you wish to resize the image you must check the 'Resize image' box and then specify the given parameters. This is usually done for increasing computation speed.	

7. Feedback frame



The feedback view looks like this and is present on each window. It is a text box that displays feedback from the application.