BIOPHOTONICS – EXERCISE 3: HYPERSPECTRAL IMAGING

1 General Information

Please prepare the task until the deadline provided in the slides. Please upload your solution via Digicampus. Make sure that the following requirements are met:

- Hand in the experiment protocol as PDF
- The experiments and tasks, as well as results should be comprehensible from the protocol without having to consult this task assignment!
- List your sources!
- All source code must be available and executable for us (this implies that you specify the required packages; in addition, you have to provide necessary data and/or give advises if paths have to be adjusted; in order to do so, you should provide a reasonable README file).
- The usage of the Spectral Python (SPy) package is recommended to solve the exercise
- The used dataset can be copied from our institute's NAS server at [1], reachable from within the university network. The dataset is located under 'File Station/dsens-DFG_I/CvDb/0000_StudentDownload/Biophotonics/Exercise III'
  Username: *Biophotonics*
  Password: *bpWS2425!*

Submission: submission deadline is 12.01.2025, 23:59.
Please ask in case of any doubts regarding the contents or the forma procedure.

2 Background on the exercise

Hyperspectral Imaging describes the recording of images using cameras which capture intensity information at multiple narrow colour bands. For example, a common setup for visible light spectrum is to use 10 nm bands within 380 to 750 nm. But in this exercise we will use a dataset with wider range: The HyperBlood dataset has a spectral range from 377 nm to 1046 nm, equally partitioned into 128 spectral bands.
Though different materials of the same colour may seem to look similar to the human eye, their spectral signature differs. In medicine such data can be used for diagnostic purposes, e.g. detection of cancer tissue, retinal diseases, or research on microcirculation and tissue pathologies.
In this exercise you will learn about the perception of colour and how to work with spectral reflection data. In the end you will train a detector which is (hopefully) capable of distinguishing blood from other red substances.

3 Task(s)

One of the most common representations of colour images is the RGB colour space. It is based on the additive colour mixing and represents each colour as a composition of red, green and blue intensity.

(1) Read the RGB image file of ‘frame_image.png’ and visualize the colour image as well as grayscale representations of the red, green and blue channels.

Using standardised file formats makes it easier to exchange data and utilize public algorithms or libraries. Make yourself familiar with the SPy library. Published datasets do not always rely on container formats, as ENVI, but sometimes just present you with a collection of single images for each spectral band.

(2) Write a method to load the ENVI file of a HSI cube and save each spectral band as monochromatic image. Show the monochrome images of 4 spectral bands.

(3) Now reverse the conversion and load all the single spectral image files and recreate the datacube. Present the image of the datacube.

Research and implement a method to calculate an RBG image from its hyperspectral representation.

(4) Present your transformation algorithm and use it on the HSI data of the ‘HyperBlood’ dataset.

(5) Choose 3 images (two of them from the scenes ‘Frame’ and ‘Comparison’), and show your own reconstructed RGB images.

For the following tasks, you will need to load the ground truth annotations in the ‘anno’ folder. Create a simple user interface where it is possible to use the mouse to select two pixels in an RGB image. The UI should visualize a loaded scene as RGB and mark real and fake blood, e.g. outline the area with different colours.

(6) Load the data from scene A, B, C or D and select one pixel of real and fake blood. Show the image and your selected pixels.

(7) Visualize and compare the reflectance spectra of your pixels.

(8) Using the spectra, choose two spectral bands, where the real and fake blood differ strongly. Show the two monochrome images of the chosen spectral bands.

Expand your user interface to be able to label an arbitrary number of pixels either ‘fake’ or ‘real’.

(9) Load another scene and select several (>30) pixels for the blood and one fake blood class. Show collections of real and fake blood spectra in two plots.

(10) Create one plot, where both spectral collections are shown as average spectal line with indications of its standard deviation.

Develop a classification system, which aims to autonomously distinguish the fake from real blood in the dataset. For this part you can choose freely of various classification algorithms, e.g. SVM, clustering methods (k-means, GMM, …), neural networks, ….

When training your algorithm, pay attention to train and test splits of the data. The classification system should, given an HSI cube as input, assign a label (blood, fake blood, background) to each pixel.

(11) Present and explain your algorithm.

Investigate on performance metrics, which can be used to measure the quality of pixelwise classification. Evaluate the performance of your algorithm:

(12) Explain your performance metrics.

(13) Present your classification results and compare them to similar works from at least two other publications.

4 References (and recommended reading)

[1] https://dsens136.informatik.uni-augsburg.de:5001/
[2] http://www.spectralpython.net
[3] Michal Romaszewski, et al., “A dataset for evaluating blood detection in hyperspectral images”, Forensic Science International, 320, 2021.
[4] Peter Bajcsy and Rob Kooper "Prediction accuracy of color imagery from hyperspectral imagery", Proc. SPIE 5806, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XI, 2005.
[5] F. Yasuma, et al., "Generalized Assorted Pixel Camera: Post-Capture Control of Resolution, Dynamic Range and Spectrum," Technical Report, Department of Computer Science, Columbia University CUCS-061-08, Nov. 2008.
[6] Dietrich Zawischa, “How to calculate and render colours – thin films as an example”, https://www.itp.uni-hannover.de/fileadmin/itp/emeritus/zawischa/pdf/soapfilmcalc.pdf