

Code used: **image\_analysis\_for\_python3\_adapted\_new\_v2.py**

Run with: **pycharm community edition** (check the interpreter has files in it)

<https://www.jetbrains.com/edu-products/download/other-PCE.html>

Inputs into script:

- line 125: path to folder
- line 126: name of the image
- line 127: image type (.jpg)

Right click in script and click run – will get prompts in the “Python Console” window located at the bottom of pycharm

- Geometric correction - say: y and hit enter
- Pop up of image
- Click on the top right corner to set the first reference point
- Pop-up closes on its own
- In console for X(Rahmen) enter: 0 and hit enter
- Y (Rahmen): 0 and hit enter
- Additional point: enter: y and hit enter
- Pop up reappears and click in bottom left corner and repeat the process
- Use the profile dimensions from the field notes paper for x and y
- Then do bottom right corner and then top right
- After top right corner coordinates are entered for “Additional point” enter: n and hit enter
- Pop up with geometric corrected image appears
- Close the pop-up and code will continue
- For Recut image enter: y and hit enter
- Pop up appears
- Click on top left and then bottom right corner of the area you are interested in
- Pop up changes to be the cropped image
- Close the pop up
- For stop script here, enter: n
- lower boundary hue, enter a number – usually around 160-175
- upper boundary hue, enter a number – usually around 170-190
- lower boundary value, enter a number from 0 to 1 – always used 0
- lower boundary y, enter a number from 0 to max plot height – always used 0

-pop up of classified image will appear

-close pop up

-code will ask Repeat classification, saying y will repeat the steps for entering the classification values, and n will finish the script and generate a set of files in the folder entered on line 125

- “image name\_data.txt” is the classification matrix – 1 for blue and 0 for not blue

- “image name\_classification\_values.txt” is documenting the 4 values used to classify

- “image name\_rowtots.txt” is a list giving the total number of blue pixels per row

- “image name\_klassifiziert.jpg” is the classified image – black represents blue pixels

- “image name\_bearb.jpg” is the cropped image

-the photos that were longer than wider weren't read in properly in the code. Using Paint an extra white space was added and the photo was saved as “image name\_wide.jpg” and used in the processing. **The resolution was 96 DPI and this might cause a problem?**

-after the image was cropped a “colourcheck.png” was saved and uploaded to a hue picker website to choose the range of hue values. **This is very subjective**