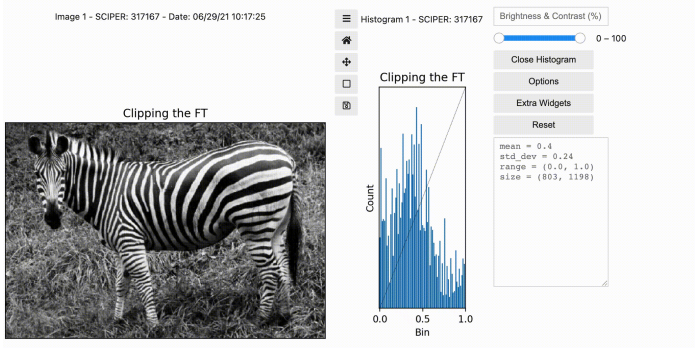
Remote and interactive image processing programming laboratories with Jupyter

Pol del Aguila Pla, Ph.D.

Research staff scientist
Center for Biomedical Imaging
Switzerland

Postdoctoral researcher
Biomedical Imaging Group
EPFL, Lausanne, Switzerland



IEEE Finland Joint Chapter of the Signal Processing and Circuits and Systems Societies

Webinar - June 30, 2021

1

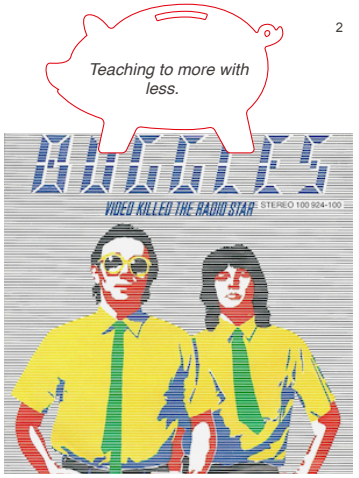
A visual summary

2

2000

2020

- Java, ImageJ → jupyter, JS
- 



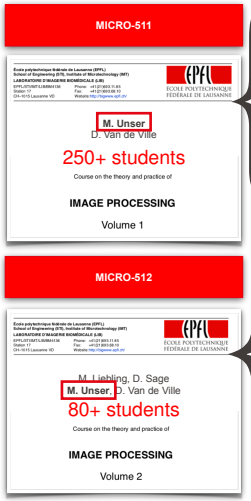
Video killed the radio star, by The Buggles (1979)

Image: Owned by studio, fair use to represent work.

2

Image Processing @ EPFL - Biomedical Imaging Group (BIG)

3



1. Continuous images and the Fourier transform
2. Image sampling, acquisition, quantization
3. Discrete images and filtering
4. Point-wise, filtering, and morphological operations
5. Segmentation, edges, and detection
6. Directional image processing
7. Continuous vs discrete: interpolation and splines
8. Image transforms, wavelets, denoising, and coding
9. Image deconvolution
10. Tomography: reconstructing from projections
11. Statistical pattern classification and neural networks
12. Image analysis: shapes and textures

Study programmes:

A. Life sciences eng.

B. Micro-eng.

C. Electrical eng.

D. Computer eng.

E. and more.

3

Image Processing Programming Laboratories (2000s)

4

Pedagogical goals

- Student engagement
  - Theory ↦ practice. Abstract formulas to concrete code
  - Application of IP algorithms to real images
  - Interactive visual feedback
  - Provide a setup for students to implement their own pipelines
- Adaptation to diverse backgrounds
  - Programming *by example*: minimal prerequisites
- Learning outcomes: *The student should be able to*
  - Implement important IP algorithms down to the pixel level
  - Implement important IP pipelines end to end
- Democratization of education: levelling the field
  - Based on open-source technology (no need for software licenses for revision / reuse)

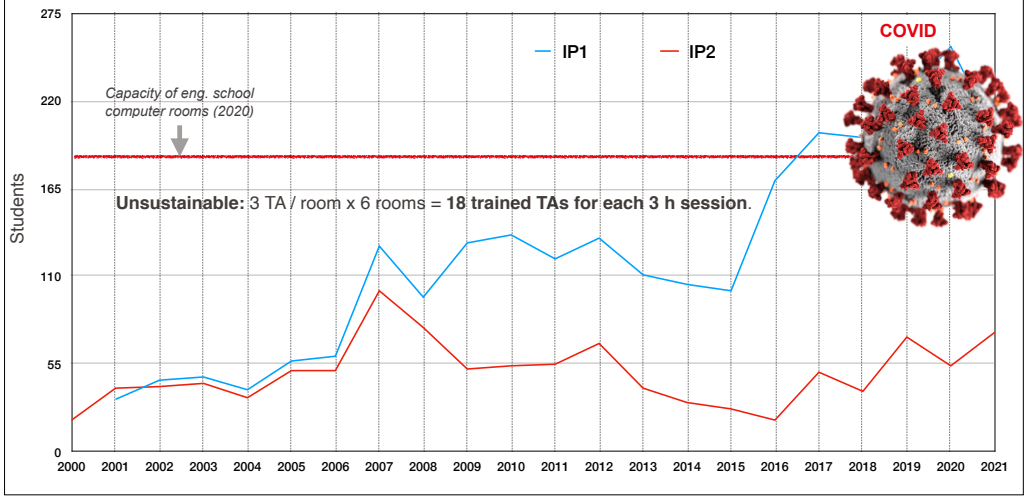
"The best way to understand an algorithm is obviously to code it and to test it"

Teaching Image-Processing Programming in Java
D. Sage, M. Unser
IEEE Signal Processing Magazine, 2003

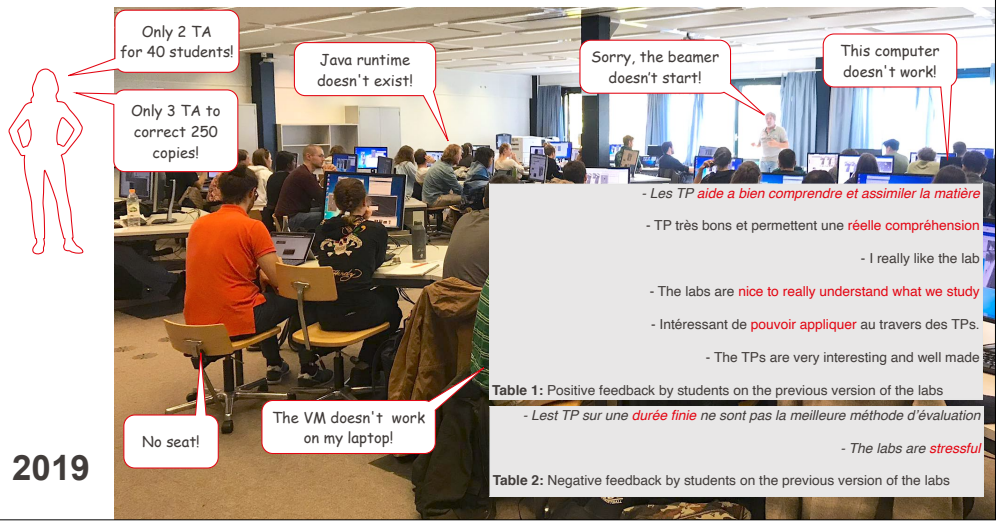
Dr. Daniel Sage, Scientist and Head of Software Development at BIG

4

Need for scalable solutions - Hard limits



Need for scalable solutions - Resource allocation



- *Les TP aide a bien comprendre et assimiler la matière*
- TP très bons et permettent une réelle compréhension
- I really like the lab
- The labs are nice to really understand what we study
- Intéressant de pouvoir appliquer au travers des TPs.
- The TPs are very interesting and well made

Table 1: Positive feedback by students on the previous version of the labs

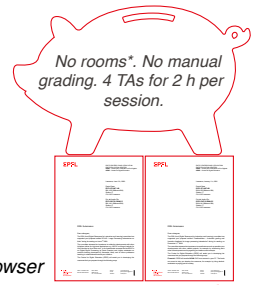
- *Lest TP sur une durée finie ne sont pas la meilleure méthode d'évaluation*
- *The labs are stressful*

Table 2: Negative feedback by students on the previous version of the labs

Remote Image Processing Programming Laboratories (2020s)

Pedagogical goals

- Student engagement
  - Theory ↦ practice. Abstract formulas to concrete code
  - Application of IP algorithms to real images
  - Interactive visual feedback
  - Provide a setup for students to implement their own pipelines
  - Work from home
  - Virtually no time limits
- Adaptation to diverse backgrounds
  - Programming *by example*: minimal prerequisites
  - No installation hurdles - *just a browser*
- Learning outcomes: *The student should be able to*
  - Implement important IP algorithms down to the pixel level
  - Implement important IP pipelines end to end
  - Use industry-relevant technologies for IP programming
- Democratization of education: levelling the field
  - Based on open-source technology (no need for software licenses for revision / reuse)
  - Need of only the most basic hardware to reuse the content

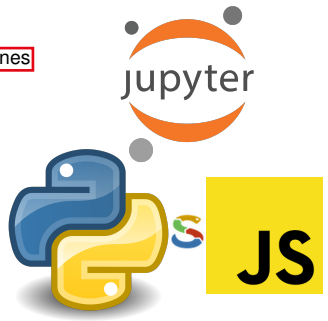


EPFL DRIL Fund (CHF 44k)

- IP1, Lab 0: Introductory
- IP1, Lab 1: Pixel-wise operations and the Fourier transform
- IP1, Lab 2: Digital filtering
- IP1, Lab 3: Morphological operators
- IP2, Lab 4: Directional image analysis
- IP2, Lab 5: Geometric transformations and spline interpolation
- IP2, Lab 6: Wavelet transform
- IP2, Lab 7: Deep learning and backpropagation

Polyglot Notebooks for Image Processing Programs

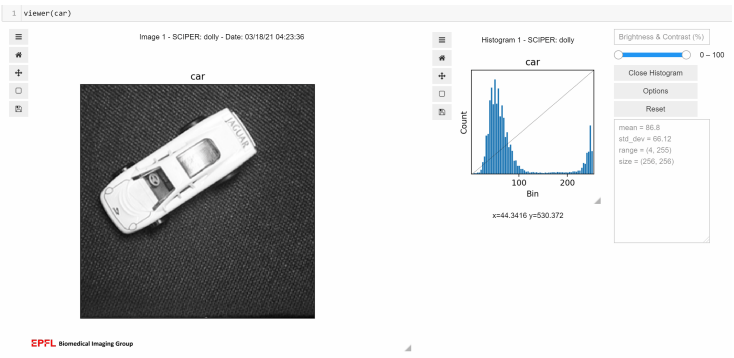
- Learning outcomes: *The student should be able to*
  - Implement important IP algorithms down to the pixel level
  - Implement important IP pipelines end to end
  - Use industry-relevant technologies for IP programming
- Student engagement
  - Provide a setup for students to implement their own pipelines



An Interactive Viewer for Image Processing Programming Labs

- Simplify image visualization for students
- Offer a broad range of native capabilities

replacing ImageJ



Alejandro Nogueron Aramburu
Master Student at EPFL
Core developer

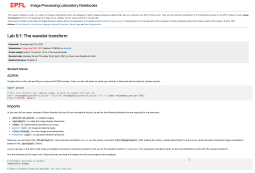
jupyter

ipywidgets: Interactive HTML Widgets github.com/jupyter-widgets

matplotlib

An Interactive Viewer for Image Processing Programming Labs

- Reduce the length of the lab notebooks



replacing ImageJ

- Reduce time spent on code that is not strictly related to the content

```
1  # IPLabViewer
2  viewer(boats, hist = True, axis = True)     1 line, simple and clear
3
4  # MatPlotlib
5  hist, bins = np.histogram(boats, bins = 70, range = (boats.min(), boats.max()))
6  fig, axs = plt.subplots(1, 2)
7  im = axs[0].imshow(boats)                   7 lines, complex object-oriented representation
8  im.set_cmap('gray')
9  axs[0].axes.yaxis.set_visible(True)
10 axs[0].axes.xaxis.set_visible(True)
11 axs[1].bar(bins[:-1], hist, width = (bins[1] - bins[0]) / 1.5)
```

Alejandro Nogueron Aramburu
Master Student at EPFL
Core developer

jupyter

ipywidgets: Interactive HTML Widgets github.com/jupyter-widgets

matplotlib

An Interactive Viewer for Image Processing Programming Labs

- Maximize the didactic capabilities of the labs

replacing ImageJ

Context: IP2, Lab 6: Wavelet transform *Learning wavelet denoising*

Alejandro Nogueron Aramburu
Master Student at EPFL
Core developer

jupyter

ipywidgets: Interactive HTML Widgets github.com/jupyter-widgets

matplotlib

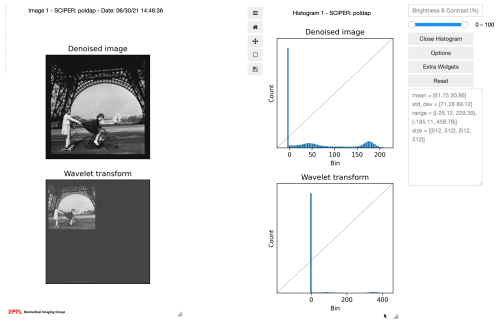
An Interactive Viewer for Image Processing Programming Labs

- Maximize the didactic capabilities of the labs

replacing ImageJ

Context: IP2, Lab 6: Wavelet transform *Learning wavelet denoising*

```
soft_thr_viewer = viewer([doisneau_noise, doisneau_noise], title=['Denoised image','Wavelet transform'], new_widgets=new_widgets,
                         callbacks=[callback_wt, callback_iwt], hist = True, subplots = [2, 1])
```



Alejandro Nogueron Aramburu
Master Student at EPFL
Core developer

jupyter

ipywidgets: Interactive HTML Widgets github.com/jupyter-widgets

matplotlib

Soon to be found in PyPI python Package Index *as part of* `interactive-kit`.

A toolkit for interactive visualization of signal and image processing on Jupyter Notebooks.

- Provides with dedicated error handling (very much lacking in JavaScript)

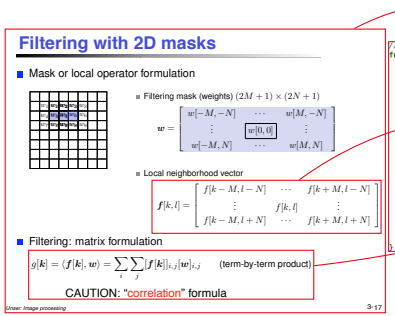
```
In [2]:  1 var a = [1, 2, 3, 4, 5]
         2 var b = a[1.5];
         3 typeof(b)

Out[2]: 'undefined'

In [3]:  1 var a = new Image(1, 5);
         2 var b = a.getPixel(1.5, 0);
         3 typeof(b)

Error: Non-integer index provided in getPixel
    at IPLabImageAccess.getPixel (C:\Users\kay-1\OneDrive\EPFL\Image Processing Lab\IPLabI
0)
```

Kay Lächler
Master Student at EPFL
Core developer

JS

- Is designed to work with neighbourhoods (simplified implementation of IP workflows)
- Handles boundary conditions
- Easy access to pixels, columns, rows, and neighbourhoods

- `getPixel(x, y)`
- `setPixel(x, y)`
- `getNbh(x, y, w, h, padding='mirror')`
- `getRow(y)`
- `putRow(y, newRow)`
- `getColumn(x)`
- `putColumn(x, newColumn)`



Kay Lächler
Master Student at EPFL
Core developer

JS

Context: IP1, Lab 2: Digital filtering

Learning the most common techniques and possible bugs when implementing digital filters in image processing


```
function erode_raw(img, b){
    // initialize the output array with the first dimension
    var output = new Array(img.length);
    // for every element of the first dimension, add the second dimension
    for(var i=0; i < output.length; i++){
        // create the second dimension
        output[i] = new Array(img[i].length);
        for(var j=0; j < output[i].length; j++){
            // initialize output value to 0
            output[i][j] = 0;
        }
    }
    // Loop through every pixel of the image
    for(var y = 0; y < img.length; y++){
        for(var x = 0; x < img[y].length; x++){
            var valmin = Number.MAX_VALUE;
            // Loop through every pixel of the neighbourhood
            for(var k = 0; k < b.length; k++){
                for(var l = 0; l < b[k].length; l++){
                    if (b[k][l]){
                        var y_ = y - parseInt(b.length/2) + k;
                        var x_ = x - parseInt(b[k].length/2) + l;
                        // Apply boundary conditions
                        if(y_ < 0){
                            // apply mirror folding
                            y_ = -y_ - 1;
                        }
                        if(y_ >= img.length){
                            // apply mirror folding
                            y_ = img.length - 1 - (y_ % img.length);
                        }
                        if(x_ < 0){
                            // apply mirror folding
                            x_ = -x_ - 1;
                        }if(x_ >= img[y].length){
                            // apply mirror folding
                            x_ = img[y].length - 1 - (x_ % img[y].length);
                        }
                        // perform erosion (calculate new minimum value)
                        valmin = Math.min(img[y_][x_], valmin);
                    }
                }
            }
            // Set the eroded pixel value in the output image
            output[y][x] = valmin;
        }
    }
    return output;
}
```

Context: IP1, Lab 3: Morphological operators

Learning the most common techniques and possible bugs when implementing morphological operators

```
function erode(img, b){
    var output = new Image(img.shape());
    // Loop through every pixel of the image
    for(var x = 0; x < img.nx; x++){
        for(var y = 0; y < img.ny; y++){
            // extract the neighbourhood around pixel (x,y)
            var nbh = img.getNbh(x, y, b.nx, b.ny);
            var valmin = Number.MAX_VALUE;
            // Loop through every pixel of the neighbourhood
            for(var k = 0; k < nbh.nx; k++){
                for(var l = 0; l < nbh.ny; l++){
                    if (b.getPixel(k, l)){
                        // perform erosion (calculate new minimum value)
                        valmin = Math.min(nbh.getPixel(k, l), valmin);
                    }
                }
            }
            // Set the eroded pixel value in the output image
            output.setPixel(x, y, valmin);
        }
    }
    return output;
}
```

Example: Implementing erosion

Kay Lächler
Master Student at EPFL
Core developer

JS

- Tools for image comparison and display in the JavaScript console

```
1 var img1 = new Image([[1, 2, 3], [4, 5, 6]]);
                ...
1 img2 = new Image([[3, 6, 9], [12, 15, 18]]);
2
3 if(!img1.imageCompare(img2, err)){
4     console.log(err.msg);
5 }

Number of mismatched elements: 6 (100%)
Max error: 12
Normalization error: The image should be normalized by a factor of 3

>> console.log(img.visualize())

[[   0   0   0   0   0 255   0   0   0   0   0 ]
 [   0   0   0   0   0 255   0   0   0   0   0 ]
 [ 255 255 255 255 255 255 255 255 255 255 255 ]
 [   0   0   0   0   0 255   0   0   0   0   0 ]
 [   0   0   0   0   0 255   0   0   0   0   0 ]]
```

Soon to be found in Node.js as `image-access`.

A class for didactic implementation of image processing in JavaScript.

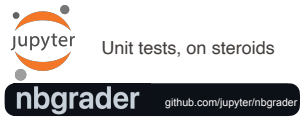
Kay Lächler
Master Student at EPFL
Core developer

JS

EPFL **Automatic Grading and Feedback for Image Processing Programming Labs**

Grading and Summative Feedback

the grading library



Unit tests, on steroids

```
### BEGIN HIDDEN TESTS
### END HIDDEN TESTS
### BEGIN SOLUTIONS
### END SOLUTIONS
```

Context: IP2, Lab 7: Deep learning and backpropagation

Learning to backpropagate without AutoGrad

```
msg_wrong = 'Your value for $\partial L/\partial b$ is incorrect.\
    The right solution is $ \\frac{\\partial L}{\\partial b} = \\frac{\\partial}{\\partial b}\\left\\lbrace (y-b)^2 \\right\\rbrace = 2 (b-y) $, \
    which, evaluated at <code>y = 20</code>, <code>b = 2 + (-4)(-2) = 10</code>, results in <code>dLdb = np.array([-20.])</code>.'
_, _, _ = grading.check_array_almost_equal(np.array(dLdb), np.array(answers.dLdb_TA), msg_wrong=msg_wrong,
                                msg_correct='Well done! The value for $\\partial L / \\partial b$ is correct.',
                                points_lost='You did not get points for your answer.', raise_exception = True)
```

Your value for $\partial L/\partial b$ is incorrect. The right solution is $\frac{\partial L}{\partial b} = \frac{\partial}{\partial b}\left\{(y-b)^2\right\} = 2(b-y)$, which, evaluated at `y = 20`, `b = 2 + (-4)(-2) = 10`, results in `dLdb = np.array([-20.])`. You did not get points for your answer.

Well done! The value for $\partial L/\partial b$ is correct.

EPFL **Automatic Grading and Feedback for Image Processing Programming Labs**

Grading and Summative Feedback

*the grading library - **advanced usage***


```
### BEGIN HIDDEN TESTS
### END HIDDEN TESTS
### BEGIN SOLUTIONS
### END SOLUTIONS
```

Context: IP2, Lab 6: Wavelet transform

Learning to implement a wavelet transform and to use the most popular libraries in Python.

A challenge for grading!

```
# Check that the students do not use PyWavelets
# First, declare a list with the students function and the necessary parameters
student_function = [analysis, lowlight, answers.analysis_hp, answers.analysis_lp, 1]
# Call our grading function IP_grading
check_error, times, _ = ip_grading.function_used(student_function, '__main__.pywt.dwt2', pywt.dwt2)
if check_error:
    msg = f'You should not have used PyWavelets for this exercise. You used it {times} times.'
    ip_grading.display_html([msg, loss_msg], 'danger', raise_exception = True)
```

You should not have used PyWavelets for this exercise. You used it 2 times. You lost 1 point because of this.

EPFL **Automatic Grading and Feedback for Image Processing Programming Labs**

The holy grail: *Formative* feedback

Context: IP2, Lab 5: Geometric transformations and spline interpolation

Learning the importance of prefiltering for spline interpolation

```
%use javascript
// Define test image and expected output
var impulse = new Image([[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]]);
var solution = new Image ([[-0.006, 0.012, -0.042, 0.158, -0.588, 2.196, -0.588, 0.158, -0.042, 0.012, -0.006]]);
// Get the coefficients
var test = cubicSplineCoefficients(impulse);
console.log('Input sequence:\n' + impulse.visualize())
console.log('Your cubic spline coefficients:\n' + test.visualize())
// Compare the output to the solution
if(test.imageCompare(solution, tol=1e-3) == false){
        throw new Error("The recursive exponential filter is not working properly, the expected cubic spline coefficients are:\n" +
                        solution.visualize());
}
console.log('Congratulations! Your implementation passed the sanity check.' +
            ' However, this is not a guarantee that it works well for any input! Revise your code! ')
```

```
Input sequence:
[[ 0 0 0 0 0 1 0 0 0 0 0 ]]

Your cubic spline coefficients:
[[ -0.006  0.012 -0.042  0.158 -0.588  2.196 -0.588  0.158 -0.042  0.012 -0.006 ]]

Congratulations! Your implementation passed the sanity check. However, this is not a guarantee that it works well for any input! Revise your code!
```


EPFL **Automatic Grading and Feedback for Image Processing Programming Labs**

The holy grail: **Visual** *Formative* feedback

Context: IP2, Lab 6: Wavelet transform

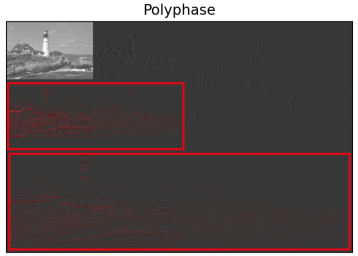
Learning the polyphase implementation of the Haar wavelet transform.

```
# Test polyphase implementation
error_haar = False
try:
    np.testing.assert_array_almost_equal(lighthouse_pywt, lighthouse_poly, decimal = 10)
except Exception as e:
    print('Your polyphase implementation is not correct. Look at the following message for details.\n')
    print(e)
    error_haar = True
    plt.close('all')
    viewer([lighthouse_poly,lighthouse_pywt],
           title = ["Polyphase", "PyWavelets (Ground truth)"], compare = True, widgets = True)
```

```
Your polyphase implementation is not correct. Look at the following message for details.

Arrays are not almost equal to 10 decimals

Mismatched elements: 231417 / 393216 (58.9%)
Max absolute difference: 684.
Max relative difference: 8.54247233
 x: array([[ 4.9900e+02,  5.0050e+02,  4.9500e+02, ...,  2.5000e+00,
         2.0000e+00, -2.5000e+00],
       [ 5.0750e+02,  5.0725e+02,  5.0875e+02, ...,  3.5000e+00,...
 y: array([[ 4.9900e+02,  5.0050e+02,  4.9500e+02, ...,  2.5000e+00,
         2.0000e+00, -2.5000e+00],
       [ 5.0750e+02,  5.0725e+02,  5.0875e+02, ...,  3.5000e+00,...
```

Image 1 - SCIPER: 317167 - Date: 06/29/21 22:03:58

Polyphase



Brightness & Contrast (%)

0 – 100

Show Histogram

Options

Reset

Prev Next

```
mean = 28.58
std_dev = 118.8
range = (-300.25, 1002.25)
size = (512, 768)
```

EPFL Biomedical Imaging Group

21

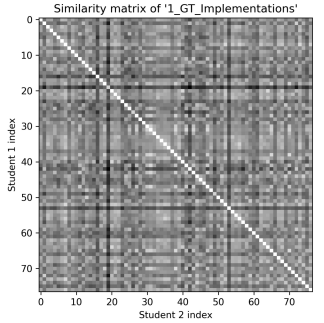
Plagiarism Detection for Image Processing Programming Labs

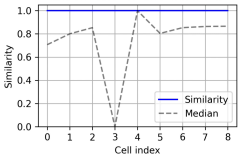
. . . a work in progress

```
plt.close('all')
PV = Plagiarism_visualization(lab_name = 'Geometric_Transformation_Lab', notebook_name = '1_GT_Implementations')
```

executed in 126ms, finished 09:59:23 2021-06-30

Figure 1



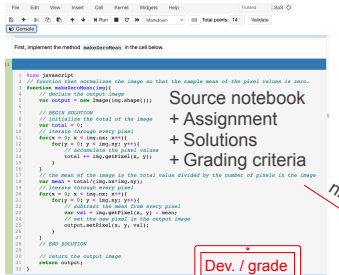


Student 1: 321084
Student 2: 321084
Similarity: 100.00 %

Compare

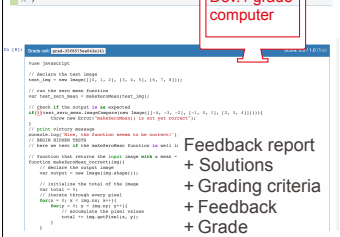
22

Infrastructure for the Image Processing Programming Labs



Source notebook
+ Assignment
+ Solutions
+ Grading criteria

Dev. / grade computer



Feedback report
+ Solutions
+ Grading criteria
+ Feedback
+ Grade

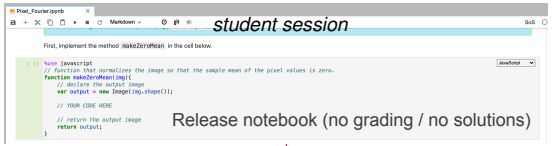
nbgitpuller

custom-made scripts

The model we've mostly used

NOTO EPFL JupyterLab servers

student session



Release notebook (no grading / no solutions)

students, manually

EPFL MOODLE

23

Infrastructure for the Image Processing Programming Labs

NOTO EPFL JupyterLab servers

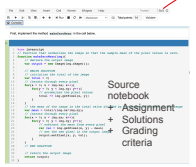
The model we're using now

nbgitpuller

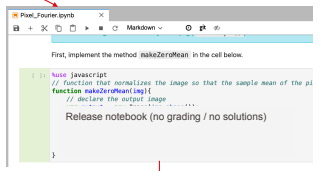
instructor session

student session

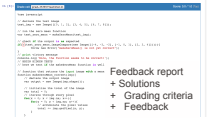
Development and grading



Source notebook
+ Assignment
+ Solutions
+ Grading criteria



Release notebook (no grading / no solutions)



Feedback report
+ Solutions
+ Grading criteria
+ Feedback

students, manually

custom-made scripts

EPFL MOODLE

24

Student feedback for the Image Processing Programming Labs

- The labs are amazing! It is the first time in my EPFL life that I enjoy doing these kind of exercises.

- (The labs) are by far the best I have encountered.

- Super labos, j'adore le format et l'interactivité

Table 3: Examples of feedback from the **Course evaluation for MICRO-511 (Image Processing I)** by anonymous students, November 2020

- it could be nice to allow submitting a little after the deadline, even for no points, ***to still get feedback***

- The labs are amazing and really helpful to understand and visualize the main concepts. Thank you for that !

Table 4: Examples of feedback from the **Course evaluation for MICRO-512 (Image Processing II)** by anonymous students, May 2020

Student feedback for the Image Processing Programming Labs

(General impression) My general impression of the Lab was Not selected (-2)Very bad (-1)Bad (0)Neutral (1)Good (2)Very good Edit

(Learning) How much do you feel you have learned completing this lab? Not selected (-2)Nothing (-1)Not much (0)A fair amount (1)Quite a bit (2)A lot Edit

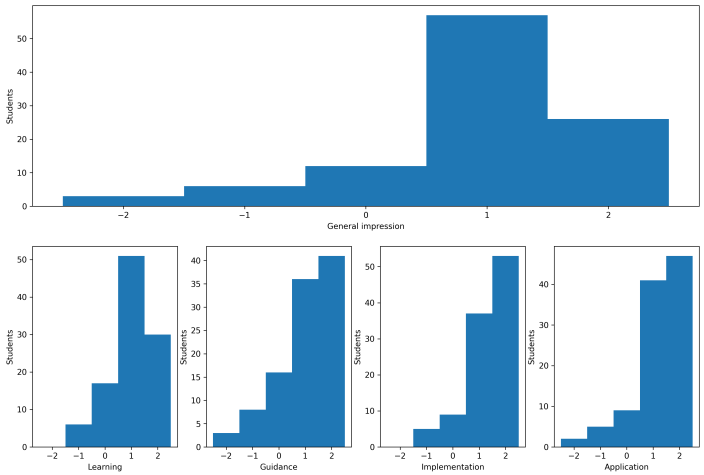
(Guidance) The level of guidance and documentation in the Lab was optimal for learning Not selected (-2)Strongly disagree (-1)Disagree (0)Neutral (1)Agree (2)Strongly agree Edit

(?????? implementation) The lab has helped me to understand how to implement the ?????? in image processing Not selected (-2)Strongly disagree (-1)Disagree (0)Neutral (1)Agree (2)Strongly agree Edit

(?????? application) The lab has helped me to understand how to use ?????? in image processing Not selected (-2)Strongly disagree (-1)Disagree (0)Neutral (1)Agree (2)Strongly agree Edit

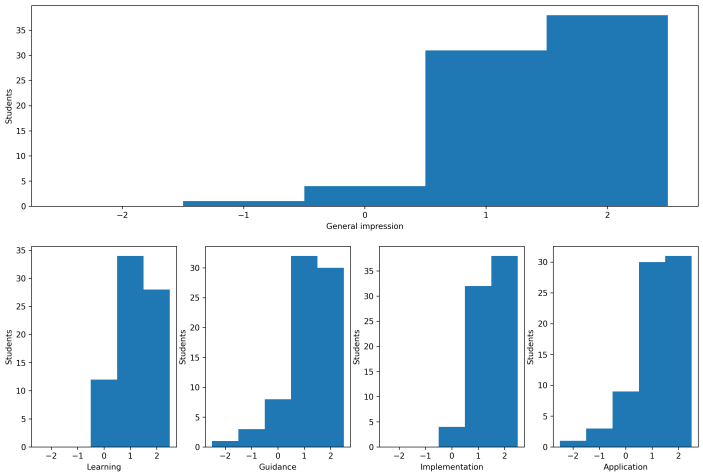
Student feedback for the Image Processing Programming Labs

Student feedback: IP1, Lab 2: Digital filtering - 104 responses



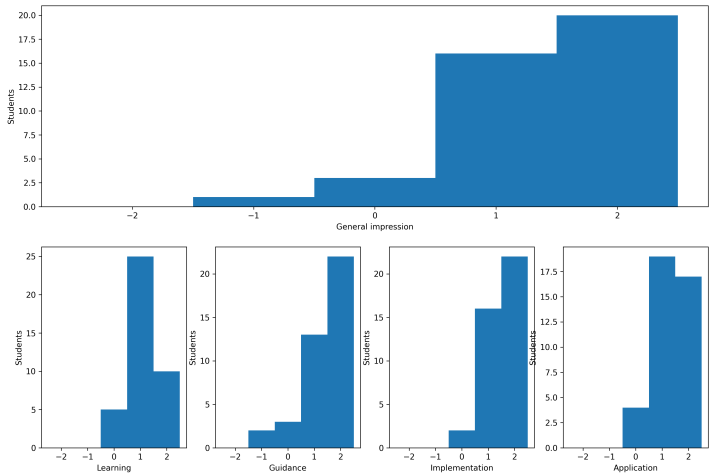
Student feedback for the Image Processing Programming Labs

Student feedback: IP1, Lab 3: Morphological operators - 74 responses



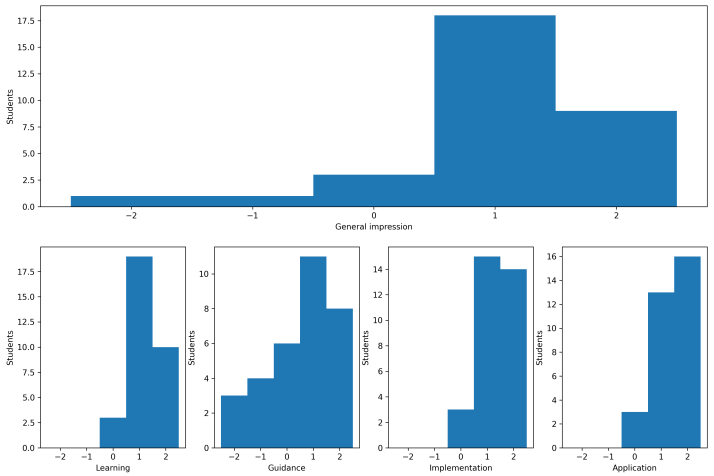
Student feedback for the Image Processing Programming Labs

Student feedback: IP2, Lab 4: Directional image analysis - 40 responses



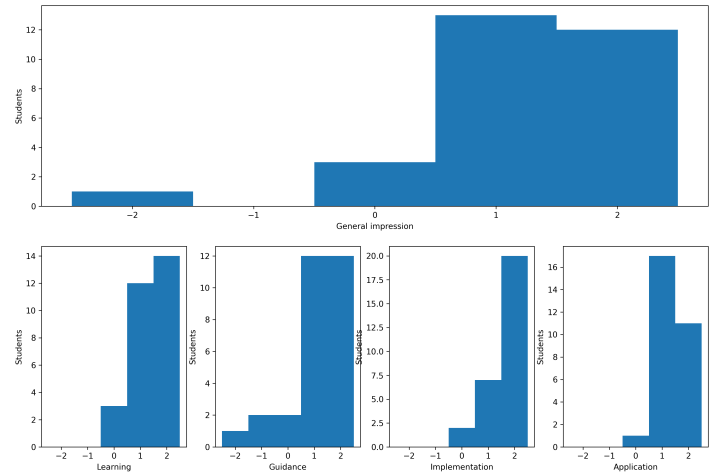
Student feedback for the Image Processing Programming Labs

Student feedback:IP2, Lab 5: Geometric transformations and spline interpolation - 32 responses



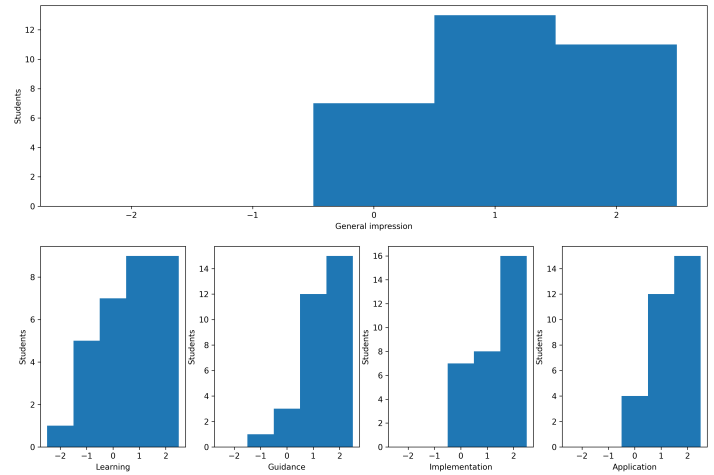
Student feedback for the Image Processing Programming Labs

Student feedback: IP2, Lab 6: Wavelet transform - 29 responses

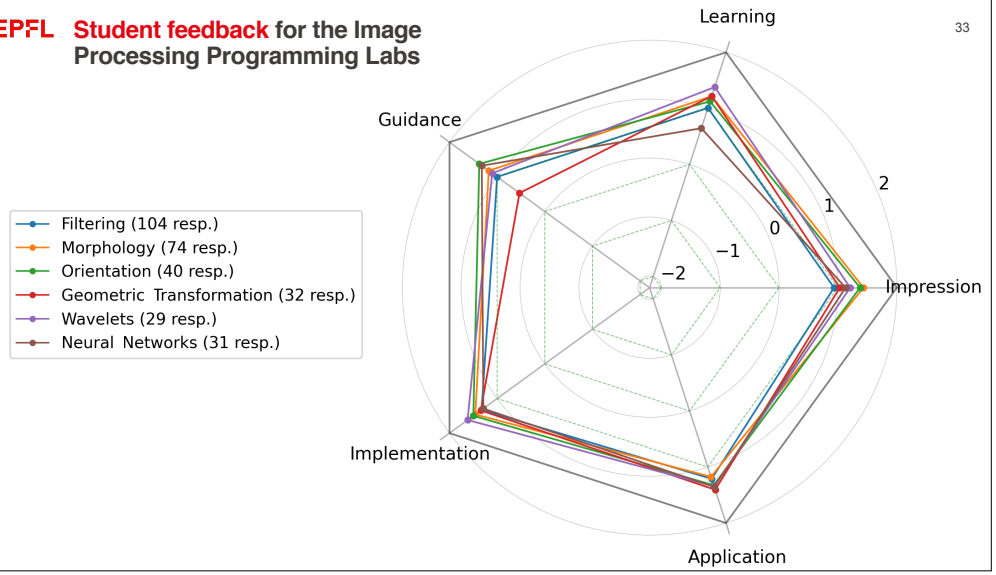


Student feedback for the Image Processing Programming Labs

Student feedback: IP2, Lab 7: Deep learning and backpropagation - 31 responses



Student feedback for the Image Processing Programming Labs



Want to use our labs in your course? Contact me!
pol.delaguilapla@epfl.ch, @polsocjo, epfl.zoom.us/my/poldap
poldap.github.io, www.linkedin.com/in/poldap

Thank you!

Our giants to stand upon

ipywidgets: Interactive HTML Widgets github.com/jupyter-widgets

IJavascript github.com/n-riesco/ijavascript

Core team

Kay Lächler
Master Student at EPFL
Core developer

Alejandro Noguerón Aramburu
Master Student at EPFL
Core developer

Dr. Daniel Sage, Scientist and Head of Software Development at BIG

Contributors

Kamil Seghrouchni
Master Student at EPFL
Developer for Lab 5

Dr. Jaejun Yoo
Post-doc at BIG
Developer for Lab 7

Funding

EPFL DRIL Fund (CHF 44k)

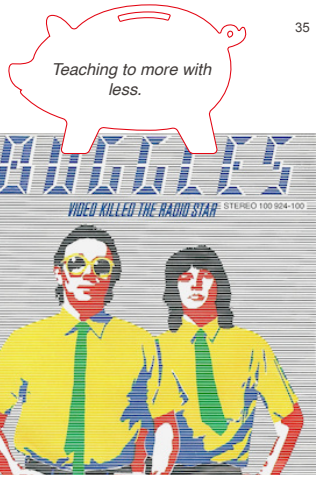
Questions?

- 2000 Java ImageJ → 2020 jupyter JS

Image: EPFL students in a computer room

Image: Rutmer Zijlstra, obtained from Noun Project, CC License

Want to use our labs in your course? Contact me!
pol.delaguilapla@epfl.ch, @polsocjo, epfl.zoom.us/my/poldap
poldap.github.io, www.linkedin.com/in/poldap



Video killed the radio star, by The Buggles (1979)

Image: Owned by studio, fair use to represent work.