

BACKGROUND

Research software engineers exist on a **continuum of different backgrounds** and skill sets, often using different tools and approaches for programming. Preferred development platforms each have their own advantages which cater to various users, but when programmers **collaborate closely** it can be beneficial to use a shared platform to encourage the shared use of shared practices. JupyterLab, an extensible platform, is capable of supporting a wide variety of development approaches and practices. We propose **JupyterLab as an ideal platform for collaboration** among programmers that prefer platforms ranging from Jupyter Notebooks to more traditional IDEs. We introduce JupyterIDE, a set of tutorials to assist interdisciplinary teams in converging on a set of shared best practices centered around JupyterLab.

POTENTIAL NEEDS:

Domain Researchers

may benefit from a solid understanding of basic version control

can benefit from a shared coding paradigm for research software

Software Engineers

may want a development tool with more features

SOLUTION

JupyterIDE provides **short modules** explaining core software development practices, like **Git version control**, implemented in JupyterLab. For individuals who want more features in JupyterLab, there are **independent modules** explaining other features available through **curated extensions for JupyterLab**. Lastly, there is a short primer on the **intersection of iterative and literate programming** approaches that is designed with research contexts in mind.

File

Edit

View

Run

Kernel

Tabs

Settings

Current Repository

jupyterlab-ide

Current Branch

main

Branches

Tags

Filter

New Branch

main

thresholding

preProcessingInterface

Changes

History

Staged

Changed

test.ipynb

Untracked

sensitiveData

Summary (required)

Description

Commit

test.ipynb

These libraries are useful standards for image processing in python, so we import them here for use during our dashboarding.

[1]:  
import cv2  
import PIL  
import numpy as np

Now we will import a test image from our data folder. In this case, we will use an image which is not too far from our other manuscripts. The page will not have any great blemishes, just to see if we can get a standard pre-test some spots going. We will also convert it to a numpy array and use some methods.

[3]:  
wellspelt\_var = cv2.imread("path/to/file")  
wellspelt\_array = np.array(misspelt\_var)

With this image, we will try a couple different 'kernels' for the different convolutions we will be using for any sort of different tasks like edge extraction, binarization, etc. We will also test a number of different boundary conditions to see if we end up served better with one of them or another. First, we will use the cv2 library for edge extraction. You can see below some of the different methods already built into the library for this sort of purpose.

Diagnostics Panel

Message	Code	Severity	Source	Cell	Line
undefined name 'misspelt_var'		Error	pyflakes		
Name 'misspelt_var' is not d...		Error	mypy		
PIL imported but unused		Warning	pyflakes		

Literate and iterative programming, used together, create documentation for dashboarding that can be replicated afterwards

jupyterlab-lsp

This extension allows users to use language server processing, including autocompletion, code linting, and diagnostics.

JUPYTER-IDE [ALSO] HAS:

Domain Researchers

a self-paced tutorial on the Git extension and version control

a short primer and example of literate programming for research contexts

Software Engineers

a curated list of extensions with tutorials giving users access to features like language server processing

REPO

Follow the link to access the GitHub repository and give it a

ACKNOWLEDGMENTS

**better scientific software**  
This work was supported by the Better Scientific Software Fellowship Program, funded by the Exascale Computing Project (17-SC-20-5C), a collaborative effort of the U.S. Department of Energy (DOE) Office of Science and the National Nuclear Security Administration; and by the National Science Foundation (NSF) under Grant No. 2154495.  
Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DOE or NSF.