GSoC 2019 Project
Towards Better Images Ecosystem

Jiuning Chen✉
Github: Johnnychen94
Slack: Johnny Chen

March 21, 2019

Abstract

This project aims to achieve a better ecosystem for Images.jl, an image-processing toolbox in Julia. Main contributions consist of definition of Images.jl ecosystem and its scope, a manual for developers, and a more consistent API.

The author is currently a third-year graduate student and Ph.D candidate in School of Mathematical Sciences, East China Normal University, Shanghai, China. His current research interests are image processing and computer vision, convex optimization, and machine learning. More information about him is listed in section 3.

1 Project Introduction

Images.jl is a Julia image-processing toolbox, including several packages such as: ImageCore.jl, ImageTransformations.jl, ImageAxes.jl. It provides a collection of out-of-box functions[1] to do image processing tasks just like scikit-image and MATLAB Image Processing Toolbox.

However, despite of the performance, this toolbox at present is still not friendly to both users and developers; Unlike other mature julia packages such as JuMP.jl and GPUArrays.jl, Images.jl requires potential users to understand the very details of its mechanism and architecture, and this is even harder for them without comprehensive documentation on it. Under this circumstance,

[1] An overview of currently implemented image-processing functionalities is shown at api comparison.

most image-processing researchers are still using Python and MATLAB for their daily work.

Some apparent causes for its poor usability are:

- there's few demos or recipes in Images.jl for new users to start with;
- the APIs lack of consistence and don't match the julian style well;
- there's no style guide on naming and programming;
- there're too many temporary helper functions defined everywhere;
- Images.jl is an ecosystem but it lacks of a comprehensive illustration of its packages;
- coverage of trait functions are not fully tested.

Fundamentally this is because that it is still in the progress of finding the most suitable programming style to process images using Julia.

Fortunately this problem is well-concerned in the community. Issues such as

- JuliaImages/Images.jl#766
- JuliaImages/Images.jl#767
- JuliaImages/Images.jl#772
- zygmuntszpak/ImageBinarization.jl#23

dicuss the coding styles in the most generic way, and packages such as

- HistogramThresholding.jl
- ImageBinarization.jl

are examples validating the effectiveness of style consensus reached in those issues. In ImageBinarization.jl, one could binarize an image using any implemented methods[2] with one unified API:

```
binarize(::BinarizationAlgorithm, ::AbstractArray{T,2}) where {T}
```

With these existing work, it's in the right time to revisit the whole Images.jl ecosystem and head towards a more easy-to-use Images.jl package. This project aims to solve this problem by:

[2] At the time of writing, there're 12 methods implemented.

- providing more comprehensive and integrated documentation on both style guide and ecosystem illustration,
- pruning codebase of the ecosystem according to the provided documentation

Writing demos of Images.jl is not included in this project since it belongs to a totally different project. Basically, this is a project on documentation and code refactoring.

2 Delivery and milestone

TODO: (I can't estimate the timeline.)

As described in the end of section 1, the project will be delivered in two stages: drafting a style guide in the first stage as a preperation, and cleaning the codebase in the second stage.

2.1 Documentation and Guide

The main purpose of this stage is to provide trackable documents for the next stage's pruning work. *The documents will be in three forms: ecosystem documentation, developer manual, and RFC.* Ecosystem documentation illustrates the scope of image ecosystem and relationships between different relavent packages, it helps users and developers to understand what package belongs to Images.jl and what package doesn't. Developer manual consists of style guide and best practice as well as other related community-operating rules. According to these two documents, a RFC with a detailed list of API changes and operation will be proposed, which will be stick onto in the next stage's pruning work.

To reach the milestone, following work will be done:

1. dig into the source code of image-processing-related Julia packages as well as other mature Julia packages such as JuMP.jl to filter out good and bad julia programming practices.
2. compare the APIs of Images.jl with that of scikit-image and MATLAB Image Processing Toolbox to find the balance between the native julian style and the familiar style for users from other languages.
3. discuss with the cummunity and draft the ecosystem documentation and style guide, define the scope of the ecosystem.

4. dig into the source code of Images.jl ecosystem, and draft the RFC[3]
5. discuss with the community and make the RFC approved.

The approval of RFC indicates the end of this stage.

2.2 Prune Codebase

With the RFC approved, the prune stage is to clean the codebase following the RFC operation guide. A milestone will be set in Images.jl to track the progress.

60% completeness rate shall be enough to indicate the end of this stage, since many repositories will be involved in this stage.

TODO: (I don't know what should be take care of in this stage)

3 About the Author

My name is Jiuning Chen. I'm currently doing research related to image processing, computer vision, convex optimization and machine learning.

3.1 Programming Background

I started to use MATLAB to do research on image processing in the end of 2016, met and immediately fell in love with Julia at Aug 9, 2018[4], and learned Python during the Spring Festival of 2019.

Although my programming career is only about three years, however, I think I'm qualified to achieve the project milestones for the contributions I've done to members in the lab of my supervisor:

- Set up the whole self-hosted research platform independently from scratch for my supervisor's laboratory[5];
- *De facto* maintainer of the deep learning servers of the School of Mathematical Sciences, and that of a laboratory in Computer Sciences Department;

[3] Tensorflow community provides a RFC Template to begin with, and 20180827-api-names.md is a good example.

[4] It's the day after the historic announcement of Julia v1.0.

[5] The platform includes but not limited to homepage, documents for users and administrators, server monitor, gitlab, jupyterhub, sharelatex, DNS servers, and VPN servers. I'd like to show you how this looks like but it's all built in a LAN environment.

- Proudly create and maintain the homepage of my supervisor, prof. Fang Li;

to undergraduate students in the university:

- Head teaching assistant of courses of ”Deep Learning and its Practice” (Fall 2018) and ”Digital Image Processing” (Spring 2019).
- Unofficially mentor talented students with all the best programming practices I learned from the open-source community and from the English world[6].

and to the open-source community:

- PR: JuliaImages/ImageTransformations.jl#58 reviewed by Evizero and Tim Holy;
- PR: JuliaImages/ImageTransformations.jl#59 reviewed by Evizero and Tim Holy;
- PR: JuliaLang/julia#29626 reviewed by Matt Bauman;
- PR: FluxML/Flux.jl#371 reviewed by Mike J Innes;
- Repo: DeepLearning_Tutorial
- Repo: Digital-Image-Processing-Gonzalez

The following is a some kind of self-evaluation to let you have a more structural overview of myself:

- **Mathematics & Image Processing (8/10)**: my current research is on image denoising based on hybrid method of variational model and deep learning;
- **Linux (8/10)**: heavy usage of docker, bash, git and vim in my daily work;
- **Matlab (8/10)**: the only programming language used throughout my early-stage of research;
- **Julia (6/10)**: fully understand and stick to the philosophy of Julia, but lack of real project experience;
- **Related packages (6/10)**: familiar with other open-source image-processing packages but haven’t dig into them yet;

[6] Most Chinese students are afraid of reading English since it’s not their native language, however, almost all best materials are in forms of it. My role here is to learn and to preach.

3.2 Education Background

2016-Present (Postgraduate) [7] Study on image processing and computer vision in School of Mathematical Sciences, East China Normal University, and supervised by Prof. Fang Li.

2013-2016 (Undergraduate) Bachelor of philosophy, Department of Philosophy, Shanghai University.

2011-2013 (Undergraduate) Study on metal material in School of Material Sciences, Shanghai University.

[7] The author just passed the Ph.D qualification examination and will be a Ph.D candidate in September 2019.