

GSoC 2019 Project

Towards Better Images Ecosystem

Jiuning Chen 

Github: [Johnnynchen94](#)

Mentors: [Zygmunt L. Szpak](#), [Júlio Hoffmann](#),
[Christof Stocker](#), [Tim Holy](#)

March 23, 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. 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 functions¹ 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

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

them without comprehensive documentation on it. Under this circumstance, 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](#)
- [zygmuntzpak/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 methods² 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:

²At 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.

TODO: (How many packages will be involved in this project?)

2 Delivery Schedule

As described in the end of section 1, the project will be delivered in two stages: **documenting** and **pruning**.

Documenting and recording in the first stage as a preperation, and cleaning the codebase in the second stage to make real changes to Images.jl ecosystem. With regard to GSoC timeline, **Phase 1** evaluates the documentation work, and **Phase 2** and **Final** evaluations focus on the pruning stage.

2.1 Documenting

Stage Expectations

The main purpose of this stage is to provide trackable records for the next stage's pruning work. There'll be three types of records: **ecosystem documentation**, **developer manual**, and **RFCs** (Request For Comments).

Ecosystem documentation illustrates the scope of image ecosystem and relationships between different relevant 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, it gives a documented reference to developers to solve potential conflicts. RFCs with detailed list of API changes and porting operation will be proposed as trackable records for the pruning work in next stage.

There'll be three side effects in this stage:

- partially rewriting of [user documentation](#) in a more meaningful way;
- potential bug reports and patches to all related **Julia** repositories;
- introduction of a new image denoising package, [ImageNoise.jl](#), as a concept-validation experimental field.

Stage Workflow

Ideally, this stage ends after the **Phase 1 Evaluation** with regard to GSoC timeline. However, since a lot of repositories will be involved in this project, which makes the timeline hard to be stucked to, the timeline serves in a flexible way.

From April 22 to June 24³, this stage will continue for ten weeks, which will be divided into two periods: **discussion period** and **RFC drafting period**.

The discussion period begins from April 22 to June 9 (weeks 1 to 7). In this period the community will share ideas and thoughts on the future of APIs and on best practices. Documenting work will be included in this period as well. The RFC drafting period begins from May 27 to June 23 (weeks 6 to 9), in this period, one or more RFCs will be drafted and discussed, as well as the detailed porting schedule. Basically, the content of RFCs come from previous discussions. The last week is used for evaluation, merge and announcement.

With a quick discussion among the community, an ecosystem documentation will be add to juliaimages.github.io as soon as possible to reach a consensus on the future of Images.jl, this consensus shall be the fundamental principle to all future discussion. The current Images.jl maintainer, i.e., [Tim Holy](#), is supposed to participate⁴.

Many discussions will happen simultaneously in the following way:

1. **Code Review:** dig into source codes of repositories of images ecosystem to find anything that's likely in need of changing. Other mature Julia packages, and image-processing libraries in other languages such as [scikit-image](#) and [MATLAB Image Processing Toolbox](#) are references.
2. **Issue Open:** open an issue for anything that is worth a discussion, e.g., legacy codes, misplaced codes, codes with bad practice, and undocumented practices and decisions.
3. **Decision Make:** the purpose of discussion is to make decision on API and practice. The conventional principles are taken: a decision is made when consensus is reached, otherwise the current maintainer of Images.jl make the decision. If a decision can't be made before June 16 (Week 8), it'll be dropped as future work.

³Although the coding officially begins from May 27, the author will start this project when he's available.

⁴In case of maintainer being busy on other work, the author will draft a document based on my understanding and post it to the community.

4. **Record:** all approved, rejected and future-work proposals will be documented in a temporary repository - [GSoC2019_Document](#). Developer manual will be drafted to [juliaimages.github.io](#) when there're enough decisions made.

RFC drafting⁵ will also happen simultaneously in the following way:

1. **Code Review:** for each approved proposal, find all involved code pieces, and give a solution to it according to developer's manual. The principle of code review is to rigorously sticking to decisions made in the discussion period – either there's one principle or no principle.
2. **RFC Post:** post the draft-version of RFC in [GSoC2019_Document](#).
3. **Discuss:** if there's any issue with any item in the proposed RFC, suspend the related items and go back to the discussion workflow until a decision is made.
4. **Merge and Announcement:** RFC merge and announcement will only happen in last two week in case there're more items to be added.

Details on porting operation will be in section [2.2](#).

Stage Evaluation

Four items are evaluated during this stage, i.e., Phase 1 Evaluation:

- 2/10: issues activity
- 2/10: ecosystem documentation
- 3/10: developer manual
- 3/10: RFCs

A score of 6/10 stands for Evaluation Pass.

2.2 Pruning Codebase(WIP)

Stage Expectations

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.

⁵A [RFC Template](#) is available in the Tensorflow community, and [20180827-api-names.md](#) is a good API-renaming RFC example.

Stage Workflow

Todolist:

- TODO: (how to take care of API changes and function moving?)
- TODO: (What part of porting work should be take?)
- TODO: (How to deal with WIP?)
- TODO: (How to deal with backward compatibility?)

Stage Evaluation

60% completeness rate shall be enough to indicate the end of this stage, since many repositories will be involved 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, 2018⁶, 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 laboratory⁷;
- *De facto* maintainer of the deep learning servers of the School of Mathematical Sciences, and that of a laboratory in Computer Sciences Department;
- Proudly create and maintain the homepage of my supervisor, prof. [Fang Li](#);

to undergraduate students in the university:

⁶It's the day after the historic announcement of Julia v1.0.

⁷The 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.

- Head teaching assistant of courses of “[Deep Learning and Action \(Fall 2018\)](#)” 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 world⁸.

and to the open-source community:

- PR: [JuliaImages/ImageTransformations.jl#58](#) reviewed by [Christof Stocker](#) and [Tim Holy](#);
- PR: [JuliaImages/ImageTransformations.jl#59](#) reviewed by [Christof Stocker](#) 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 an informal self-evaluation to let you have a more structural overview of my skill:

- **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 to maintain the servers;
- **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;

⁸Most 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) ⁹ 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.

⁹The author just passed the Ph.D qualification examination and will be a Ph.D candidate in September 2019.