Toward a User Friendly JuliaImages

Google Summer of Code 2021 Proposal

Ashwani Rathee

ab669522@gmail.com

April 2021

Abstract

JuliaImages is a state of the art image processing ecosystem available to the community. Being written purely in Julia and having a huge number of algorithms - this ecosystem provides both almost-complete and easy interface environment for users. Being said that, there is lack of organized setup for worked out demonstrations. There are several mini examples scattered around in the documentation of various packages in JuliaImages. This project aims to solve this very issue by organising the examples, reworking the current examples and adding a good number of new examples.

1. Description of the Project

Image Processing is an important part of science. Image processing plays a very important role in science from helping Hubble telescope to explore the wide universe, to helping researchers segment HeLa cells. JuliaImages is one of the most feature rich and optimized ecosystems for solving image processing problems. Performance has always been a top priority for this ecosystem. It is highly optimized towards memory and speeds given the clean architecture and efficient code design. But still, there is not yet a well organized set of demonstrations to allow new and experienced users to explore capabilities of JuliaImages similarly with well-written clear and concise demonstrations. Here I propose ideas on how the growth of this ecosystem can be further fueled by this project which focuses on the addition of new fully-worked out examples, reworking old examples and organising them under **Demonstrations**.

1.1 Project : Improved Demonstrations Section

JuliaImages ecosystem hosts numerous packages that provide equivalently complete set of highly performant algorithmic features as provided by scikit-image, opencv, and MatLab image processing frameworks.
Images.jl is an umbrella package that exports a set of packages that are useful for common image processing tasks.One particular line that caught me attention :

> ”Elements of this package descend from ”image.jl” that once lived in Julia’s extras/ directory. That file had several authors, of which the primary were Jeff Bezanson, Stefan Kroboth, Tim Holy, Mike Nolta, and Stefan Karpinski.”

From image.jl to the widespread JuliaImages,that’s a impressive story of persistence I think . I would like to be part of it in some way.

Overview of the project

The main motivation for this project is that lack of examples create a gap between the user of JuliaImages and the plethora of capabilities provided by JuliaImages. Even though the doc string of the algorithms provided by the developers are well written in most cases, but still docstring cannot replace the online demonstrations for easy and quick usage examples.

Current Status

Demonstrations section made with Democards.jl currently holds 12 examples and several others scattered in particular package documentations.
In Demonstrations :

- 4 related to general handling of image data
- 4 related to spatial transformations
- 3 related to contours
- 1 related to quality indexes

7 tutorials are currently present in ImageFeatures.jl docmentation.2 demonstrations are present in ImageFiltering.jl documentation. Demonstrations of algorithms are present in ImageSegmentation.jl all cluttered in one page.

Project Details

Project is divided in 4 sections:

- Section 1: Demonstrations on Image Binarization and Morphology
- Section 2: Demonstrations on Image Features
- Section 3: Demonstrations on Image Segmentation
- Section 4: Demonstrations on Image Quality Indexes and Contrast Adjustment

According to the current project plan, most of the work for first 2 sections should be completed before the first evaluation and similarly section 3 and section 4 work

would be done in week 6-10. Since the work done in any open source community is asynchronous,so some demonstrations/ideas might take longer to get merged/added depending on the reviews,feedback and updating the PRs with according to the feedback.
With this current project plan,12-14 demonstrations will be added covering most of capabilities in ImageBinarization.jl, ImageMorphology.jl, ImageSegmentation.jl, ImageQualityIndexes.jl and ImageFeatures.jl packages.
Project will be discussed in more details with my mentor in community bonding period.

Note: There are several projects in JuliaImages like ImageTransformations.jl, ImageFiltering.jl and from people in Julia's image processing like ImageProjectiveGeometry.jl, Augmentor.jl that haven't been touched in this project, so those might be potential demonstrations ideas for the community bonding period and of course in post GSOC period. There are several low level packages and utilities package like ImageDraw.jl,ImageView.jl, TiledIteration.jl that haven't been covered in this project plan due to time constraints and delivering limited quantity-high quality work.

1.1 Section 1: Demonstrations on Image Binarization and Morphology

ImageBinarization.jl + HistogramThresholding.jl

These packages mainly provide algorithms to convert grayscale images into black or white(binary image). Image binarization is one of the most important steps in pre-processing of images to save all or maximum sub-components such as objects, text, background, and image.
Histogram Thresholding also provides similar functionalities and a little more in terms of abilities to get more than black or white. Long demonstration will come from this section.
There are currently 12 algorithms available to perform image binarization/thresholding:

- Some Algorithms perform better for text data i.e. Niblack and Sauvola
- Some Algorithms perform better for distinct contrast between foreground and background i.e. Adaptive Threshold

Points to be covered:

- What if I don't know which one is best algorithm for my specific use case?How to check that fast?

  ```
  function binarize_methods()
  .
  .
  end
  ```

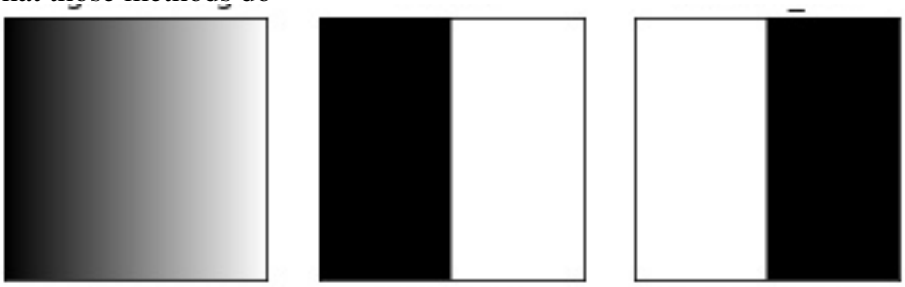
  By using this function, which applies all the methods to test image and shows all of them in one place with MosaicViews.jl

- Cases for 2-D and 3-D images

  For the 3-D case, we can use ImageView.jl to show the effect in the

3-D image to the viewer.

- Cases for textual and high contrast images need to be taken, to show Niblack and Sauvola use cases.
- For simpler methods, we can use simple linear gradients to effectively show what those methods do



ImageMorphology.jl

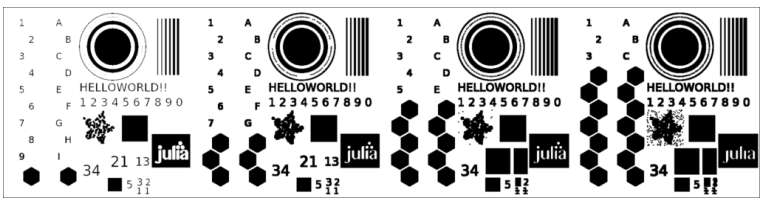
ImageMorphology.jl exports JuliaImages's collection of non-linear operations related to the shape or morphology of features in an image.

1. Basic methods : Erosion, Dilation, Closing, Tophat, Bothat, Morphological Gradient, Morphological Laplace
2. Other methods : convexhull, imfill(for small holes filling), clearborder(to clear image's border), thinning algorithm.

2 Long demonstrations: One of basic methods, and other to show usage of convex hull, image hole fill, clear border, thinning.

1. Long Example : Implementation of the basic algorithms and then comparing the results with the other results and discussion on how the threshold of Gray affect the performance.

```
img_erosion = erode(Gray.(img) .< 0.7)
```



2. Long Example with usage of convexhull, image fill, thinning, clear border and usage of polygon filling algorithm after convex hull points extraction

   Input image would be the image of a coin with unclear border, noisy with Noise.jl

3. Emphasis on input methods and conversion, since some of the above methods only take boolean images.

4. Emphasis on creation of boolean image from RGB and GRAY image using thresholding and visualizing them properly afterward

To effectively show the capabilities of the Image Binarization and Morphology with Democards for 3d Image, Democards.jl should be able to effectively show 3d image which is not the case yet. So in the community period,this problem also needs to be resolved.

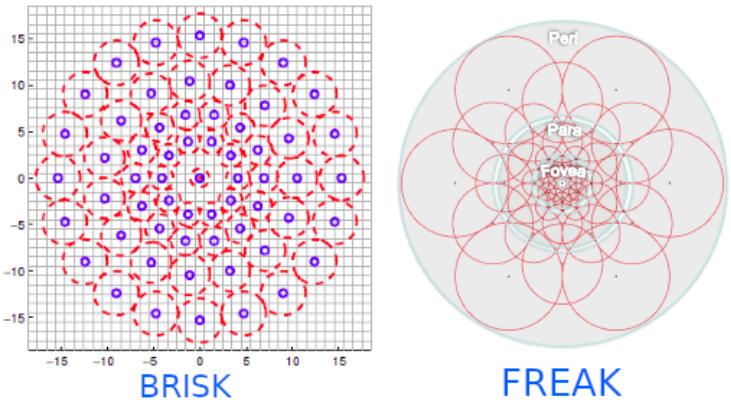
1.2 Section 2: Demonstrations on Image Features

ImageFeatures.jl

This package is used to identify and characterize key points in images.
The collection of key points can be matched between two images as shown in the tutorials.
There are 7 tutorials currently available in ImageFeatures.jl that need to be transferred to the demonstrations section and they need some bug fixing. GLCM and Local Binary Pattern tutorials need to be reworked/created.

Points to be covered :

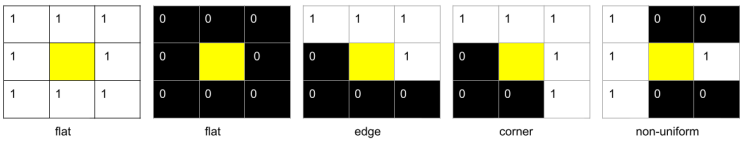
1. BRIEF, ORB, FRISK, Object detection using HOG, FREAK, Local Occurrence Matrix, Local Binary Patterns need to be transferred to Demonstrations section and reworking them to be more concise.
2. BRIEF examples need to be updated with test images other than Lena. ORB, FRISK, BRISK has issues with output calculation due to RotMatrix since Rotations.jl isn't imported.
3. Show the differences in performances of these algorithms for a image and compare results.
4. Show how these algorithms approach a problem like FREAK,BRISK has defined sampling pattern.In BRISK,pixels are sampled over concentric rings and FREAK approaches it differently.



5. Demonstration on GLCM Texture Analysis :

- How to create a GLCM matrix and how matrix is related to image
- Using distance, angles for directional analysis in GLCM
- Deriving properties like correlation, variance, mean etc
- Use of symmetric version of GLCM

6. Long Demonstration on usage of local binary patterns



Use rotated images of faces, other objects at different orientations and then compare them and check if they are similar with Image differences methods

- Show how the LBP are calculated, computation of histogram from frequency of cells
- Show how LBP can be used in texture and pattern recognition

1.3 Section 3: Demonstrations on Image Segmentation

ImageSegmentation.jl

This package provides the algorithms related to segmentation of images. Although documented quite extensively in the documentation, it needs to be broken into short demo pieces that are quick,concise to use for a new user.
Most of the demos can be considered under the reworking section since they are pretty well documented but needs to be broken down into pieces.
Since this part needs a lot of repetitive work, there is need to discuss on where the emphasis should on certain part of the algorithms.
For example : In watershed segmentation, the distance to the background is calculated which should be emphasised on.
Points to be covered :

- Watershed Segmentation Algorithm
  - Focus on showing distance transform,labelcomponents,labelsmap and showing segmentation accuracy with various degree grayscale images
- Seeded + Unseeded Region Growing Algorithm
  - Emphasis on suitable selection of seed points, minimum area threshold, similarity threshold value is important
- Felzenswalb's Region Merging Algorithm
- Mean Shift Segmentation Algorithm

  - Emphasis on how meanshift doesn't scale well with size of image and why we use iters,eps keywords,how kernel size i.e. spatial radius,range radius affects resolution of mode detection

- Fast Scanning Segmentation Algorithm
  - Emphasis on how fast it can be and how choice of threshold affects results
- Fuzzy C-means,K-means Algorithm
- Creating Region Adjacency Graphs
- Creation of Region Trees and Region Splitting using RegionTrees

Also, Comparison demonstration/tutorial of these segmentation and superpixel algorithms would also improve the user's understanding on how and where a particular algorithm would be most efficient.

1.4 Section 4: Demonstrations on Quality Indexes and Contrast Adjustment

ImageContrastAdjustment.jl

This package provides algorithms for contrast adjustments. Most of its exported functions use cases are shown in histogram equalization example and histogram matching example.
A good idea would be to rework both the histogram matching demo and histogram equalization demo to more accurately show
Points to be covered:

- Use a different test image than moon surface, probably mountain stream png or lakecolor tiff, as they will be more clearly able to show how gamma correction among others can be used for images clicked in poor lighting conditions.
- Also,it would be a good idea to take a matrix and show the effect of all the algorithms in one place.

255	255	255	108
20	124	162	168
198	212	58	14

Contrast Increased

72	123	66	27
5	31	41	42
50	53	15	4

Contrast Decreased

144	245	132	54
10	62	81	84
99	106	29	7

Initial Image

- Also, examples need to be reworked to show the histogram calculated through buildhistogram functionality

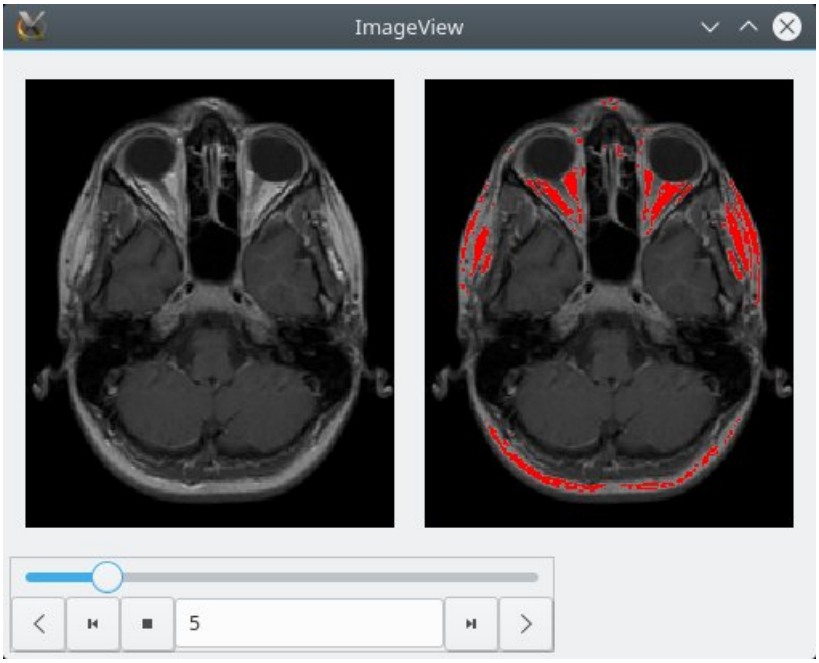
ImageQualityIndexes.jl

ImageQualityIndexes provides the basic image quality assessment methods. There are currently 3 methods that return ratios:

- Peak signal-to-noise ratio
- Structural similarity
- Multi-scale SSIM
- Colorfulness

Even though SSIM has a good demonstration already available on the website, we should make a QualityIndexes Complete example for a movie-like array of Images using ImageView.jl as shown in its documentation for human brain MRI. Points to be covered:

- All 4 Quality Indexes to be calculated for two movies(arrays of images) run side by side like shown below:



ImageFiltering.jl

ImageFiltering.jl implements blurring, sharpening, gradient computation, and other linear filtering operations, as well as nonlinear filters like min/max. ImageFiltering.jl provides 2 demos:

1. Custom median filters (shows use of median filter with mapwindow)
2. Max min filters(show use of min,max filter)

These tutorials need to be moved to demonstrations section.

2. Road Map for the project

I will not be available for two or three separate days in June (traveling). Apart from this, I have no prior commitments during this period.
I have been participating in discussions and issues for sometime now, and have learned about how codebase is organized and works to some extent.
I have worked on previous demos and had received a lot of review from johnny. So I understand the layout on how to approach demos and present them in reasonably well manner.
I will be able to devote 40-45 hrs a week during the coding period. After GSOC,I will devote 15+ hrs a week since my interest fall in similar regions and Julia is worth exploring with the run time speed it provides for signal processing and machine learning.

Potential Mentors

Tim Holy, Zygmunt L. Szpak

Timeline

- **Buffer Period: April 13 - May 17** I would try to complete my remaining PRs or tasks (if any) and try to finish them before the community bonding period. I will try to go through on details of algorithms for image segmentation, GLCM and LocalBinaryPatterns.
- **Community Period : May 17 - June 7** I plan to get to know more people in the Julia Community during this time period. I will need to brush up any theory that may be required during the coding phase.
- **Coding Period Starts:**
  - Section 1: Demonstrations for Binarization and Morphology
    - Week 1 : Long Demonstration on Image Binarization and Histogram Thresholding
    - Week 2: Long demonstration on basic morphological methods and second demonstration combined for all other morphological methods
  - Section 2: Demonstrations on Image Features
    - Week 3: Transferring and reworking/updating tutorials of BRIEF,ORB,FRISK and FREAK,Object detection using HOG to be more concise and clear.
    - Week 4: Long Demonstration on usage of Local Binary Patterns as an object detector and demonstration on usage of GLCM and statistics related to it.
  - First Evaluation: July 12 - 16
    - Review on the work that has been done and updating the work flow if there's a need.
  - Section 3: Demonstrations on Image Segmentation
    - Week 6: Demonstrations on seeded,unseeded region growing, and watershed segmentation

    * Week 7: Demonstrations on Felzanszwalb,Mean Shift, and fast scanning algorithms
    * Week 8: Demonstrations on Region Adjacency Graphs and Region Trees.
  - Section 4: Demonstrations on Image Quality Indexes, Image Contrast Adjustments
    * Week 9: Demonstration on Image Quality Indexes which shows usage for 2-D and 3-D images
    * Week 10: Reworking Image Contrast related examples and transferring examples from ImageFiltering.jl and completing the WIP tasks from past weeks
  - Final Evaluation: August 16 - 23
    * Review on the work that has been done ,submitting the final work product and write final report

  * Depending on how PR review progresses, some demos might take longer to be added/merged.

Deliverable

- 2 Demonstrations covering ImageMorphology.jl capabilities and 1 demonstrations covering ImageBinarization.jl
- 2 new Demonstrations covering GLCM and LBPs and transferring/reworking ImageFeatures.jl tutorials
- 6-8 new Demonstrations covering ImageSegmentatio.jl algorithms from majorly reworking current documentation of the package and a comparison demonstration
- Reworked ImageContrastAdjustment.jl tutorials to improve coverage and 1 final tutorial on quality indexes from ImageQualityIndexes.jl

Post GSoC

I have learned a lot and picked up a lot of important skills like testing and benchmarking by contributing to Julia and even after Google Summer of Code,I plan on continuing my contributions to this organization and working on new interesting projects and working on open issues.
JuliaImages provides me with a good platform to hone my Julia programming skills and put my mathematical skills to good use. Since my interests lie in Signal Processing and Machine learning, several things come up in my mind:

- Work toward benchmarking JuliaImages algorithms with other Image Processing frameworks
- Work toward my major project(in early phase) Libranya.jl which provides audio signal processing ability which involves a lot of image processing too. Julia doesn't provide any proper high level audio signal processing packages which other languages provide like librosa,essentia etc
- Work toward image restoration and inpainting related algorithms since I have worked with these topics manually,it's always better to have algorithms.

3. About the Author

I am Ashwani Rathee,Second year student studying Information Technology for my Bachelors of Engineering at University Institute of Engineering and Technology,Punjab University,Chandigarh.
I'm currently working as flutter software intern for a project-based learning company Upepo(Our app is available on playstore,the one my team and me directly contributed too)
I am one of the mentors for GirlScript Summer of Code,2021 2 which is 3-month Open Source Program for a Machine Learning Project AlgorScriptML.
This is my first time being a mentor,I'm trying my best and have learned several things about being a mentor.
So yeah,I am trying to be a good mentee and a good mentor both at the time(At the time of writing,GSSOC'21 is halfway through)

AlgoScriptML : AlgorScriptML

I have been playing taekwondo since 2011 and have played at both state and national level and have represented Chandigarh in School National Games.I am Black Belt(DAN 1) in taekwondo and practicing for DAN 2.

Contact Details:

E-mail: ab669522@gmail.com

GitHub: https://github.com/ashwani-rathee

Website: https://ashwani-rathee.github.io

Programming Experience and Mathematical Background

I have been programming from last two years, and from the past year in Julia. Apart from Julia I have experience in Python, Dart,Flutter,Java. For version control, I have been using git.
I work on Ubuntu 20.04 with VScode as my primary editor. I like vscode in most cases because its fast and productive.
I have completed courses on Julia Scientific programming(coursera), Linear Algebra and have keen interest in signal processing(image and audio).

Code Portfolio

- Conscious - Utility tools webapp for people with special needs
- Sinfork - Audio feature extraction desktop tool
- Alz-help - Detection of Alzheimer's disease by using standardised tests like MOCA,SAGE,etc with flutter mobile app

All of my projects (including others) can be found on my github page.

Contributions to the community

I started using Julia in last june and I made my first contribution to JuliaMusic in June where I made their website with Franklin.jl,fixed minor bugs in Music Processing.jl and MIDi.jl . I also later helped in the transfer of th sciml website to franklin.jl with chris rauckaus and tlenart.
In October,I started contributing to JuliaImages and have been contributing ever since then.
Main discussion thread : JuliaImages Project Discussion

Merged PR's

- ImageDraw.jl 52 Added rectangle point drawable
- Juliaimages.github.io 167 - Spatial Transformations Demo which included examples for cropping,resizing and rescaling
- TestImages.jl 94 96 - Uploaded artificats chelsea and coffe to TestImages.jl and made them loadable.
- Juliaimages.github.io 183 177 - Canny Edge Filter Demo which showed example of creating a image with JuliaArrays and then using Canny Filter from ImageEdgeDetection.jl
- Juliaimages.github.io 178 185 - Histogram Matching Demo which showed example of histogram matching.
- SciML Website : setup the base for the transfer of the website from Jekyll to Franklin.jl,transferred most of the data from old website to new one.
- MIDI.jl 125 : Fixed issue with file handling,files ending with .MID were saved as nameofthefile.MID.mid,helped resolve it.

Unmerged PR's and Relevant issues

- ImageDraw.jl 57 Polygon Filling API for polygon filling algorithms like boundaryfill, floodfill, scan-line fill algorithm.[Initial Implementations]
- ImageDraw.jl 51 - **[WIP]** Fill,Thickness keyword for drawbale type of Circle,Ellipse,and Circle Thre Points
- PaddedViews 49 -**WIP** Additon of Dims keyword to pad only in specified direction
- TestImages 53 - **WIP** Image with numbers created with Luxor.jl to show the morphological operations in a standard way.
- Image Edge Detection 19 - Resolves problem of incorrect handing of NaN values,raised the issue in my main discussion thread.