Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Bayesian Matting Final Presentation

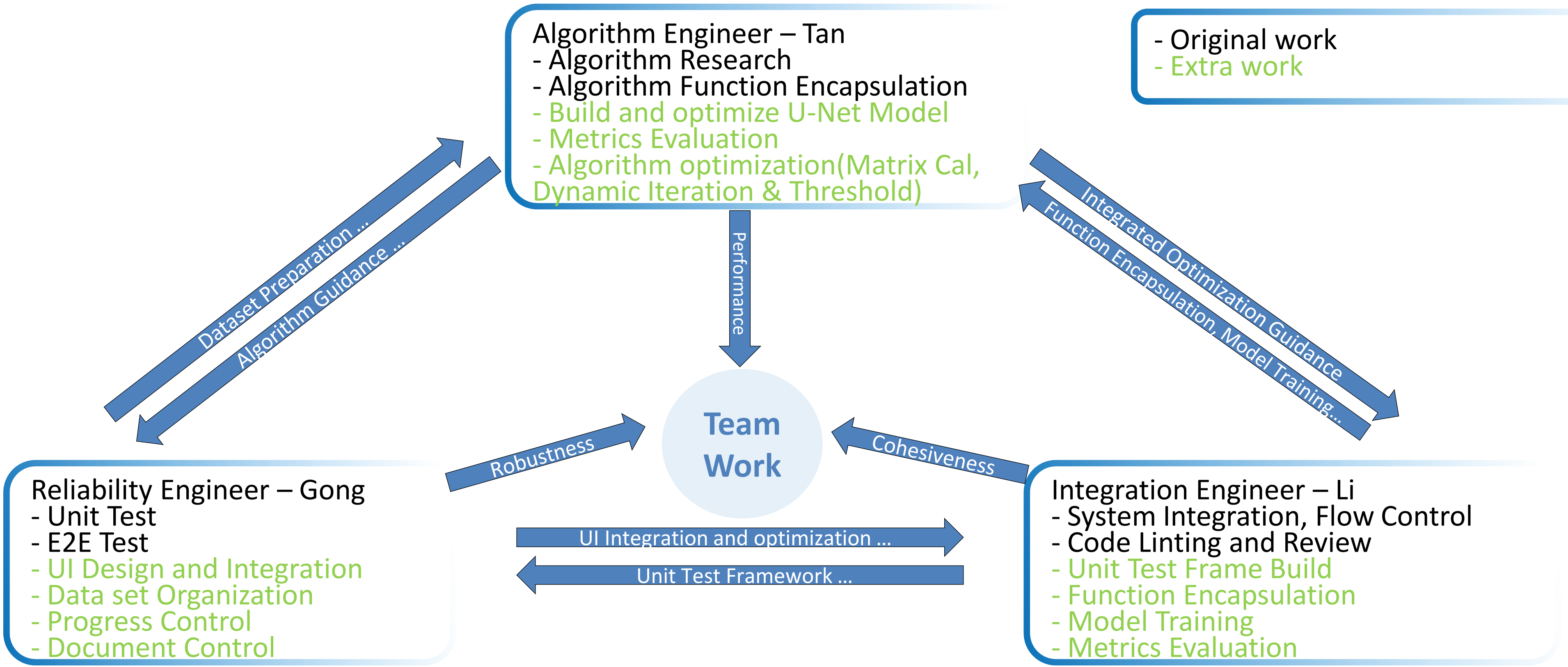
Matting based on Bayesian algorithm

Lingyu Gong, Changhong Li, Qiwen Tan
E3 School

Date 21/03/2024

CONTENTS

Group Overview - Group Roles & Changes



Flexible roles, Proactive advancement, Mutual assistance

Group Overview - Implementation vs plan

No	Period	Task	Result	Res	1	2	3	4	5	6	7	8	9	10	11	12
1	PLAN(1-4)	Group Name and Membership	Project Membership structure map	/												
2		Roles of everyone in the group[algorithms, i/o, testing]	Project Membership structure map	/												
3		Describe the key mathematical steps in easy-to-understand terms	3-5 pages slides handcraft	Tan												
4		Show how that leads to a flow diagram of the implement														
5		Present a plan showing the core algorithmic functions and development .														
6		Define an e2e test and how it will be implemented														
7		Define unit tests for your core functionality	unit test instances list	Gong												
8		Declare some milestones and timeline	This plan	Li												
9		Slides integration	Plan slides	Gong				★								
10	MATLAB IMP(5-6)	Implement Core Algorithm function block	Functional Code block	Tan												
11		Integrate the code and compare the result with Laplacian matting	Matlab Project	Li												
12		Code Linting and review	Matlab Project	Li												
13		Unit Test and e2e test								★						
14	PYTHON IMP & PROGRESS UPDATE(7-9)	Implement Core Algorithm function block														
15		Integrate the code														
16		Code Linting and review														
17		Unit Test and e2e test	Test Report	Gong												
18		Slides update Algo	3 pages slides	Tan												
19		Slides update Inte	3 pages slides	Li												
20		Slides update Test	3 pages slides	Gong									★			
21	FINAL PRESENTATION(10-11)	Slides update Algo	5 pages slides	Tan												
22		Slides update Inte	5 pages slides	Li												
23		Slides update Test	5 pages slides	Gong												
24	TEST OTHER CODE(12)	Lint check	Lint Report	Li												
25		Unit Test	Unit test Report	Tan												
26		E2E Test	E2E test report	Gong												
27		Test Report	Integrated Test Report	Li												★

	Plan
	Finish
	Delay
	Advance
★	MileStone

Problem: Task **seriality** is high
Solution: **Redistribute** and **collaborate** to drive progress

Problem: Milestone **node requirements in advance**
Solution: Re-plan and carry out the next phase of work **in advance**

Benchmark plans on time, Modify plans in time

Performance Measurement

Accuracy

MSE(Mean Squared Error): 0.02169
The average squared difference between estimated and ground truth matting results in Bayesian matting

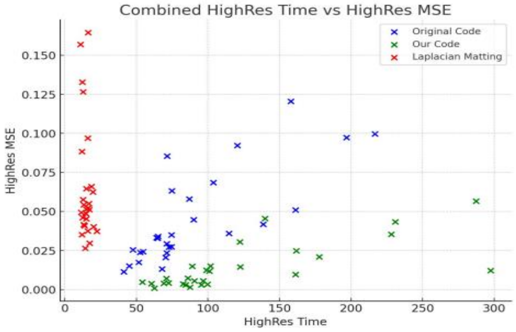
PSNR(Peak Signal-Noise Ratio): 16.63610
The quality of the matting result by comparing the signal power to the noise power.

Algorithm Comparison (Laplacian vs Ori Bayesian vs Our Bayesian)

Laplacian Matting:
Time advantage, not suitable for complex tasks

Original Bayesian Matting:
Relatively balanced

Our Bayesian Matting:
Relatively long time, but the effect is the best



Complexity

Elapsed Time: 4.66284 s
Time consumption of Bayesian algorithm in specified operating environment

Memory Usage: 555567 bytes
Memory resource usage of Bayesian algorithm in specified operating environment

Algorithm Comparison (Bayesian + U-Net vs Bayesian + Tri-map)

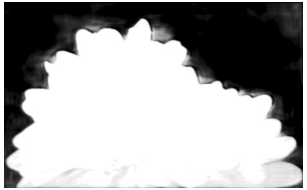
Bayesian + U-Net:

- Any image input, no need to manually annotate Tri-map
- More time consumption(< 2s) and accuracy loss (PSNR = 14.31230)

Bayesian + Tri-map:

- Tri-map needed
- Less time consumption and accuracy advantage (PSNR = 16.63610)

Tri-map

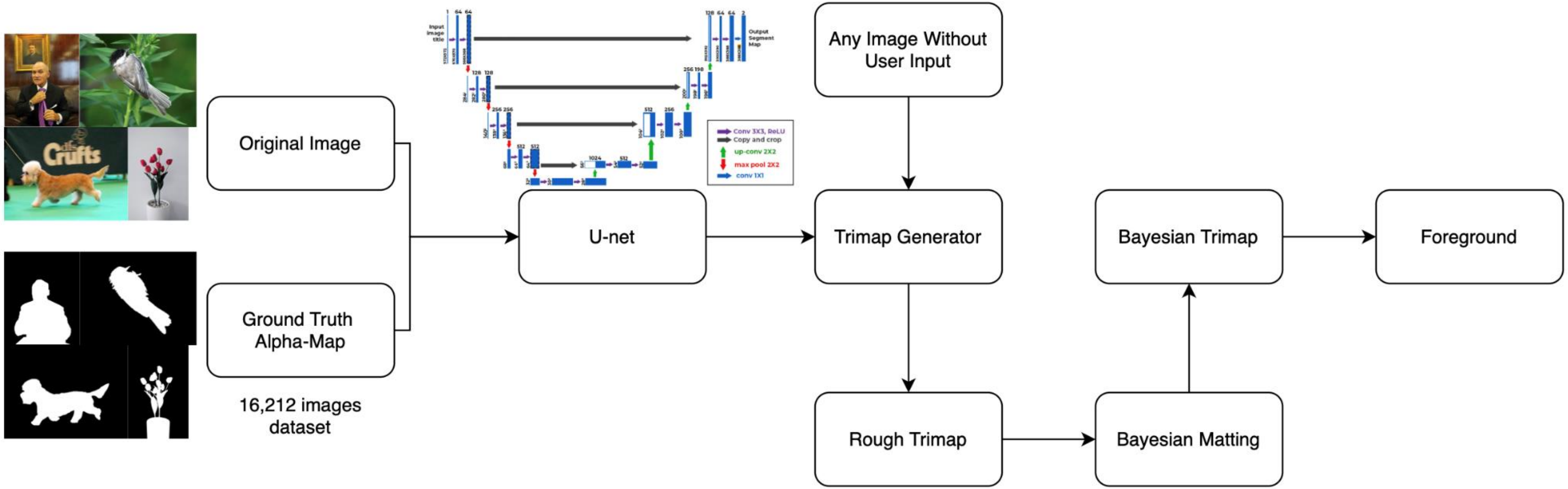


Tri-map by U-Net

The above performance evaluation is based on GT01 low-resolution images using Ori=8, iteration=2 with python scripts.

Implementatons

Trimap Generator

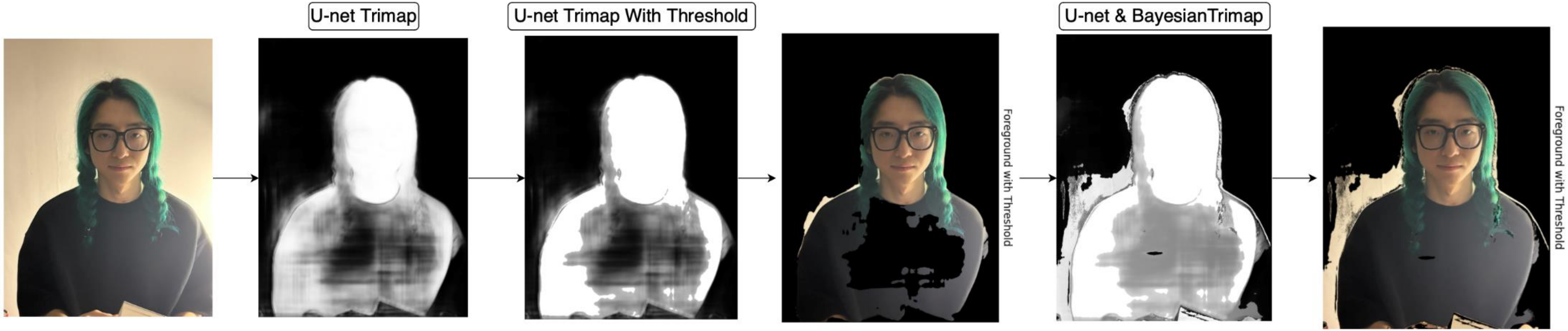


Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Implementatons

Advantage

No User Inputs Needed & Can Take in Any Size Image

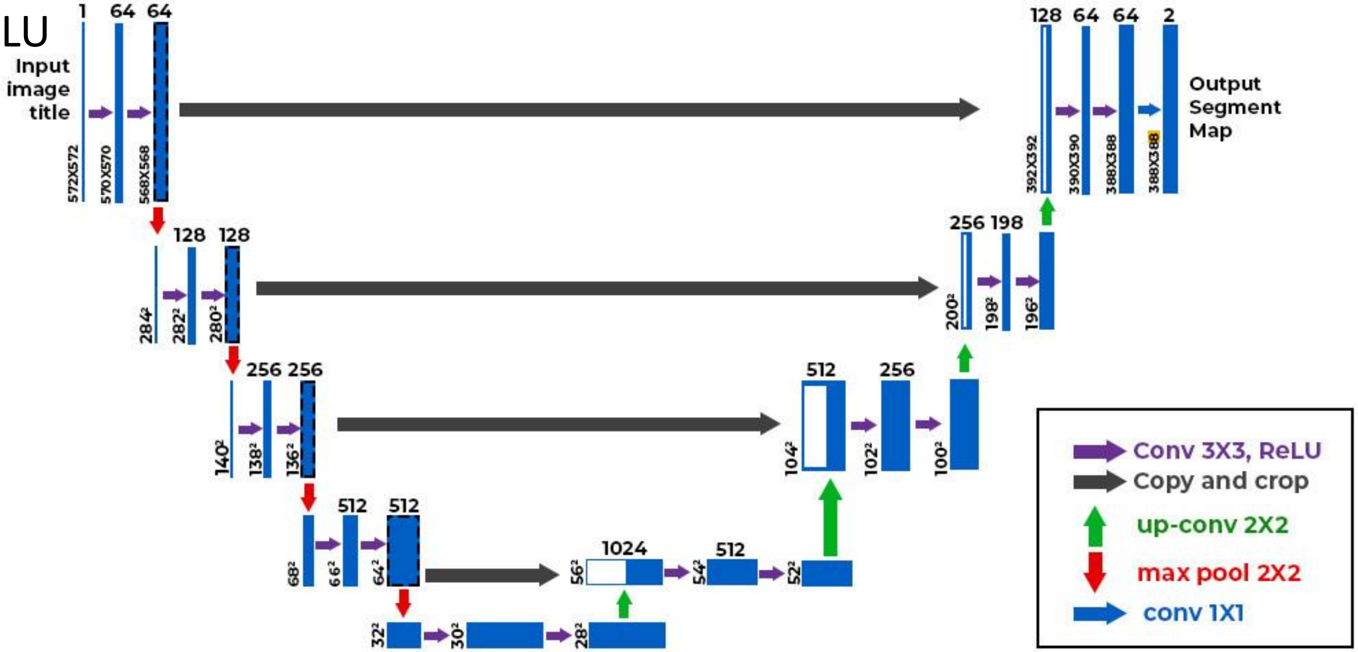


Model

Structure

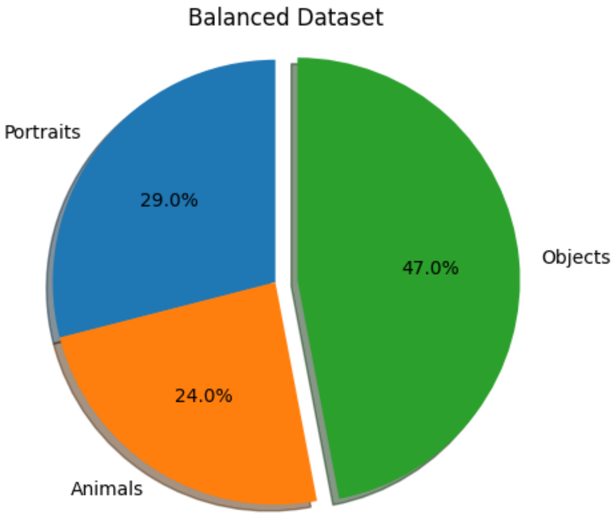
Loss Function: Dice Loss

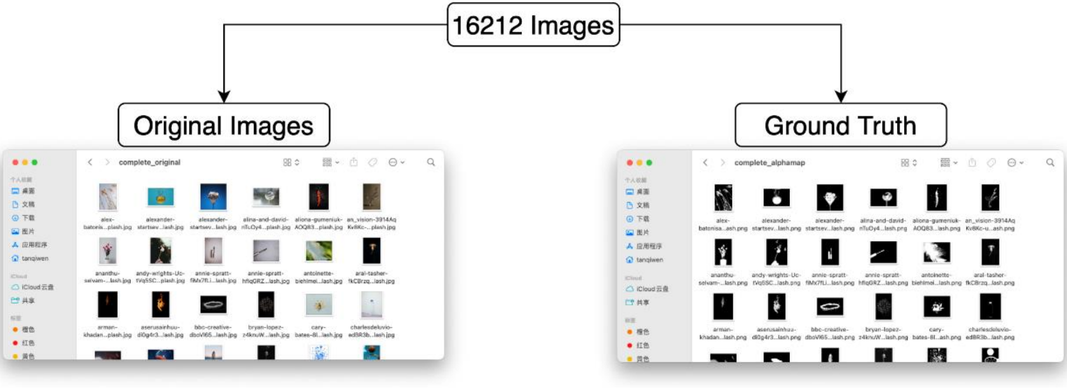
Activation Function: ReLU



Model

Dataset





Model

Smooth Alpha-Map

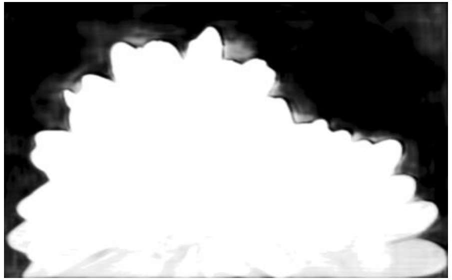
Sigmoid as the activation function for the output layer.
Ensuring the smoothness of the trimap.

Optimization

Dynamic Foreground / Background Segmentation

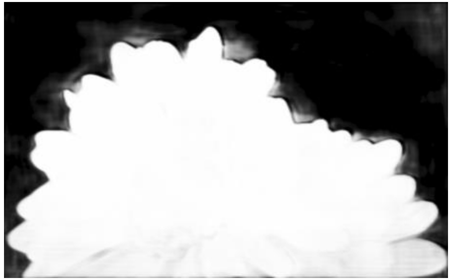
Originally

Foreground: >255×0.95
Background: <255×0.05



Optimized

Foreground: 25% whitest pixels
Background: 25% darkest pixels



Optimization

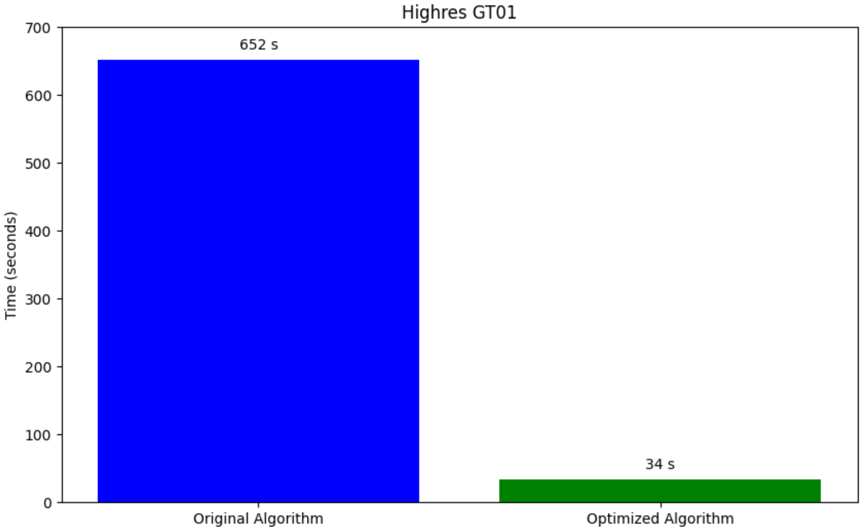
Operations of For Loop > Matrix

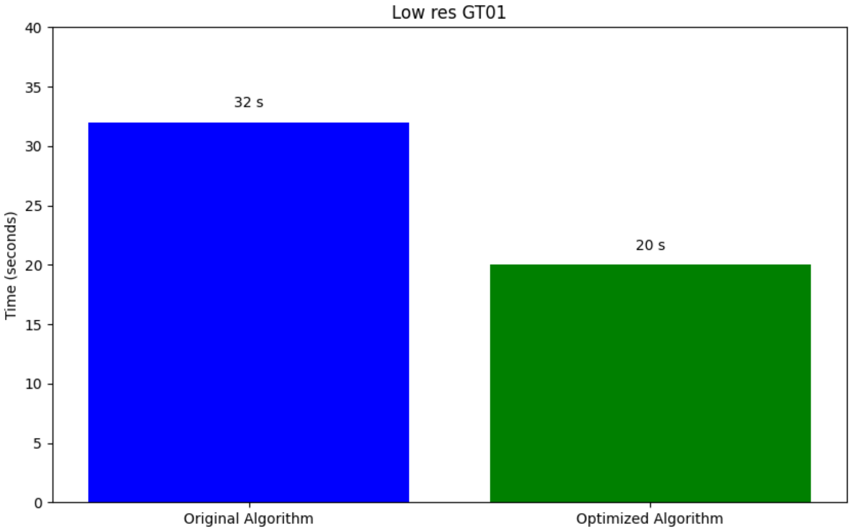
For Loop > Matrix Operation

Process multiple pixels for foreground and background

&

update multiple alpha values





Optimization

Dynamic Iteration

Calculate image complexity through three indices

&

Establish the relationship between image complexity and iteration.

$$Iteration = \frac{\sqrt[4]{Pixel\ count\ \times\ Entropy\ \times Grayscale\ Levels\ \times\ 100}}{50}$$

Pixel count = Width × Height

Entropy = The average amount of information in the image.

Grayscale Levels = The vividness or uniformity of colors within the image.

Optimization

Dynamic Iteration

Pixel count = 12,192,768
Entropy = 7.69
Grayscale Levels = 256
Iteration = 25

Pixel Count: 6193265
Entropy: 7.04
Grayscale Levels: 223
Iteration: 20

Highres GT01

Pixel Count = 397600
Entropy = 7.03
Grayscale Levels = 217
Iteration = 10

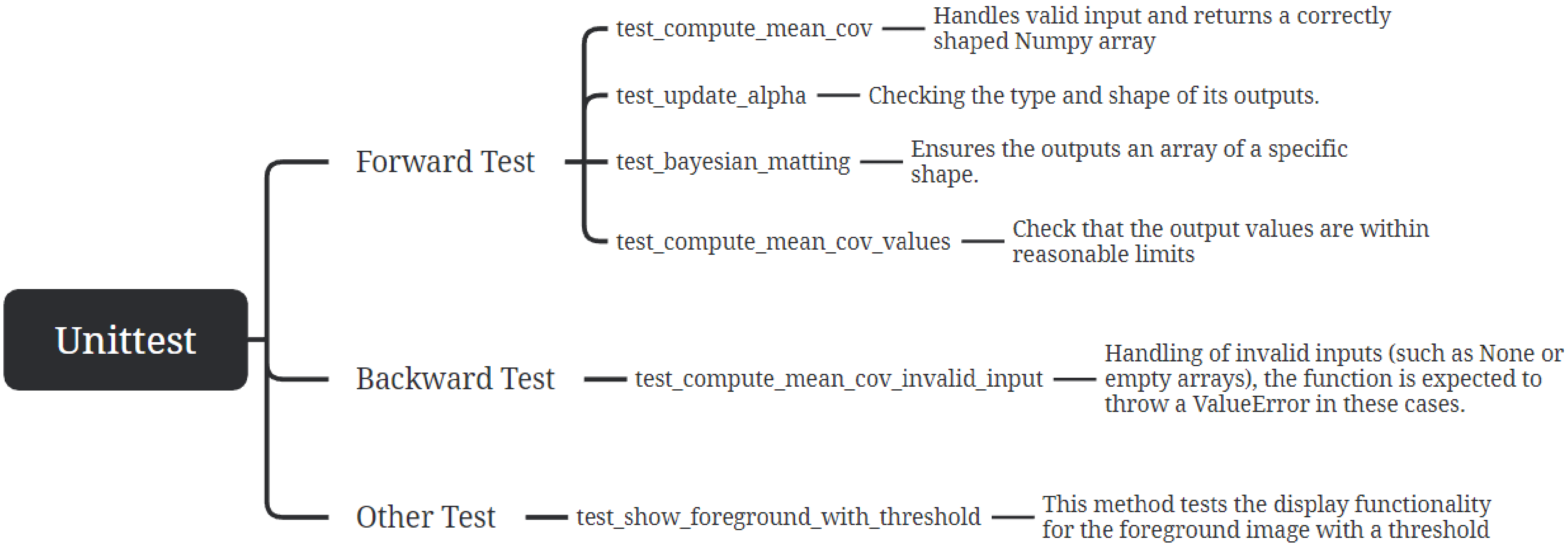
Lowres GT01

Pixel Count: 7576800
Entropy: 7.04
Grayscale Levels: 245
Iteration: 22

Highres GT04

Pixel Count: 450400
Entropy: 7.01
Grayscale Levels: 238
Iteration: 11

Lowres GT04


Unittest
Forward Test
test_compute_mean_cov
Handles valid input and returns a correctly shaped Numpy array
test_update_alpha
Checking the type and shape of its outputs.
test_bayesian_matting
Ensures the outputs an array of a specific shape.
test_compute_mean_cov_values
Check that the output values are within reasonable limits
Backward Test
test_compute_mean_cov_invalid_input
Handling of invalid inputs (such as None or empty arrays), the function is expected to throw a ValueError in these cases.
Other Test
test_show_foreground_with_threshold
This method tests the display functionality for the foreground image with a threshold

E2E Test

- Environment configuration (setUp method).
- Test execution (test_bayesian_matting_on_dataset method).
  - The test verifies that the output of the bayesian_matting function is as expected
- Resource cleanup (tearDown method).
- Script Execution Entry.

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Thank You