Name: Aryavrat Gupta
JHED ID: agupt110
Compiler / Environment / Platform Used: GCC, macOS, Visual Studio Code (VS Code)
Number of late days used: 1

Fully Implemented:

1. addRandomNoise
2. brighten
3. luminance
4. contrast
5. saturate
6. crop
7. quantize
8. randomDither
9. orderedDither2x2
10. floydSteinbergDither
11. blur3x3
12. edgeDetect3x3
13. nearestSample
14. bilinearSample
15. gaussianSample
16. scaleNearest
17. scaleBilinear
18. scaleGaussian
19. rotateNearest
20. rotateBilinear
21. rotateGaussian
22. funFilter

Partially Implemented:

1. Beier-Neely Morphing

Left Un-implemented: (N/A)

Source and Destination Images:

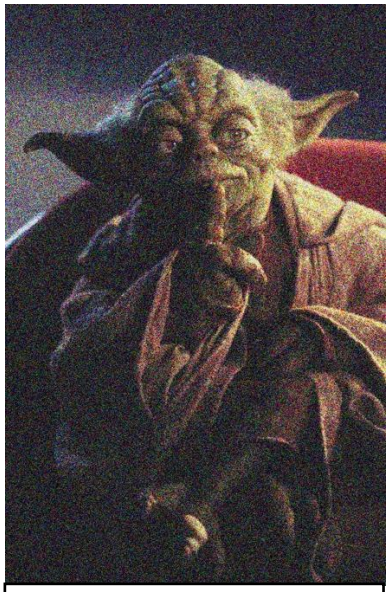
Source Images are named in the form <operation>.src.bmp. E.g. noise.src.bmp.
Destination Images are names in the form <operation>.dest_<parameter(s)>.bmp.
All the source and destination images are linked in agupt110_HTML directory.

1. addRandomNoise

Noise source img

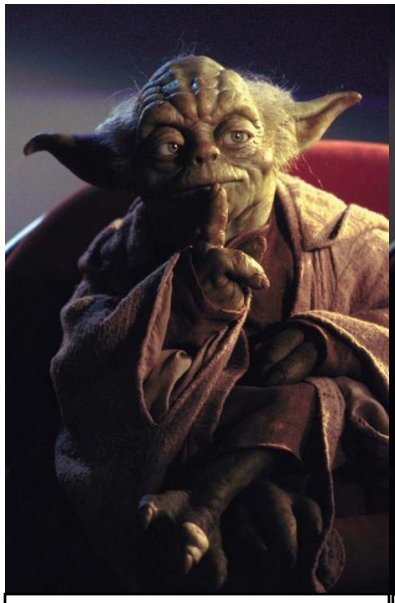
--noisify 0

--noisify 0.1



--noisify 0.5

2. brighten



Brighten source img

--brighten 0.5

--brighten 1

--brighten 2

3. luminance

gray source img

--gray

4. contrast

Contrast source img

--contrast 0.5

--contrast 1

--contrast 2

5. saturate

Saturate source img	--saturate 0.5

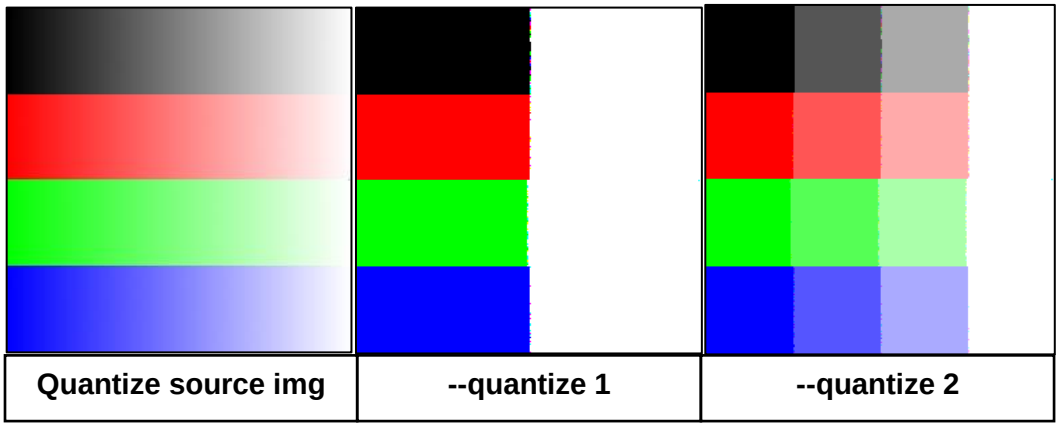
--saturate 1	--saturate 2

6. crop

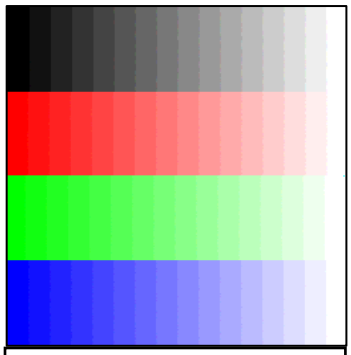
crop source img	--crop 50 50 600 500

7. quantize

* Images are outlined black for clarity

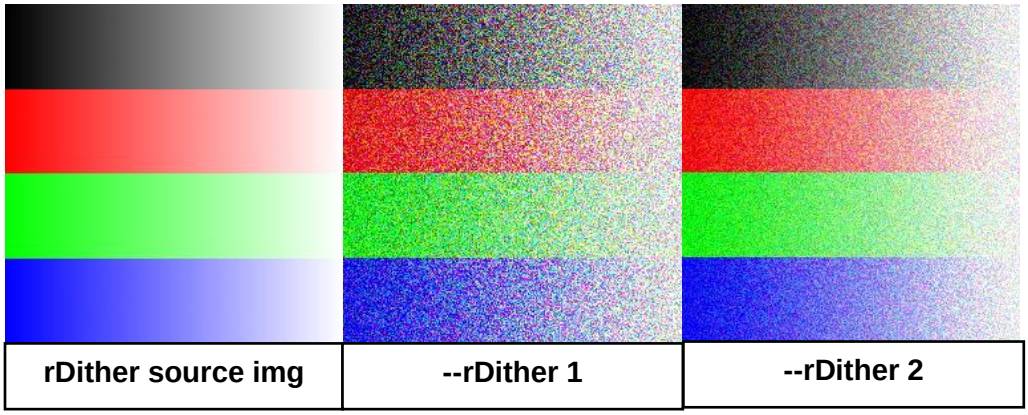


Quantize source img	--quantize 1	--quantize 2

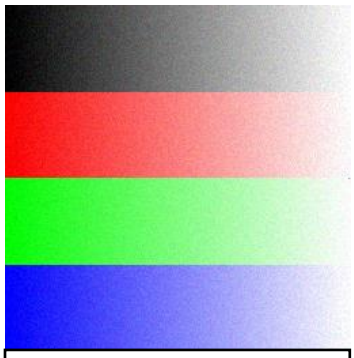


--quantize 2

8. randomDither

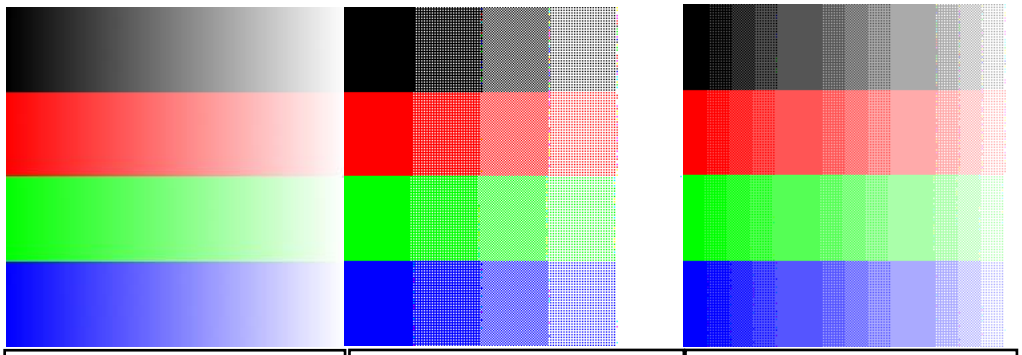


rDither source img	--rDither 1	--rDither 2

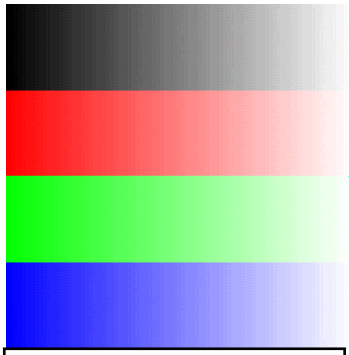


--rDither 4

9. orderedDither2x2

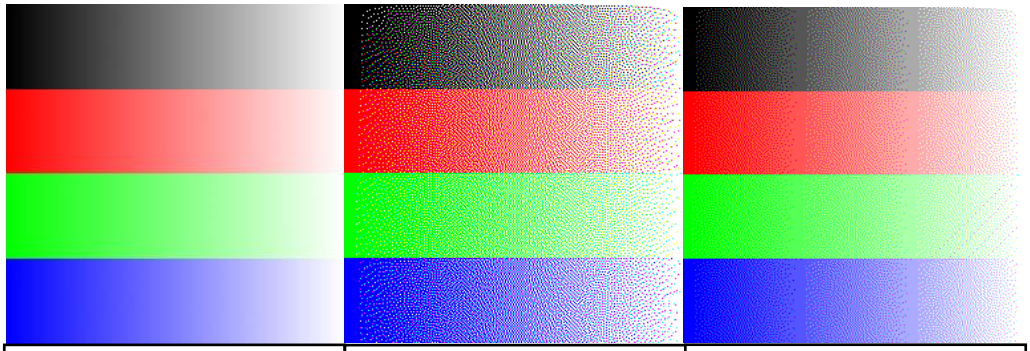


oDither source img | **--oDither2x2 1** | **--oDither2x2 2**

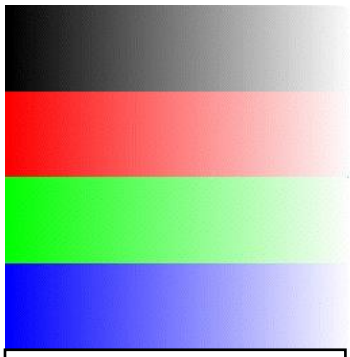


--oDither2x2 4

10. floydSteinbergDither



fsDither source img	--fsDither 1	--fsDither 2



--fsDither 4

11. blur3x3

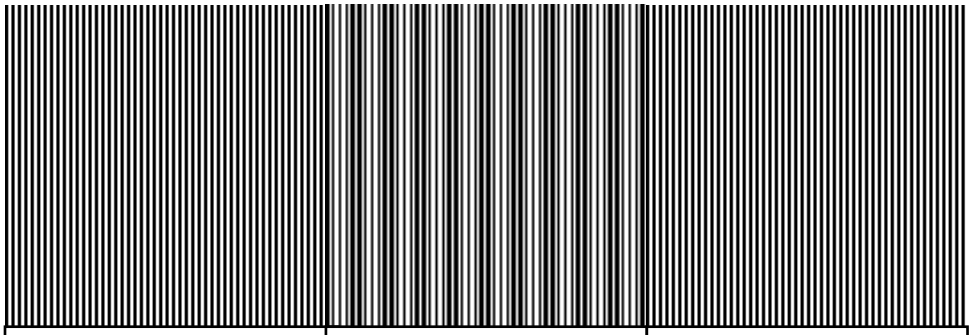
Blur source img	--blur3x3

12. edgeDetect3x3

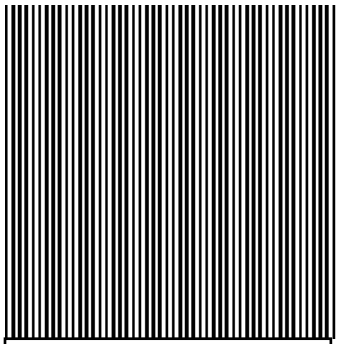
Edges source img

--edges3x3

13. scaleNearest

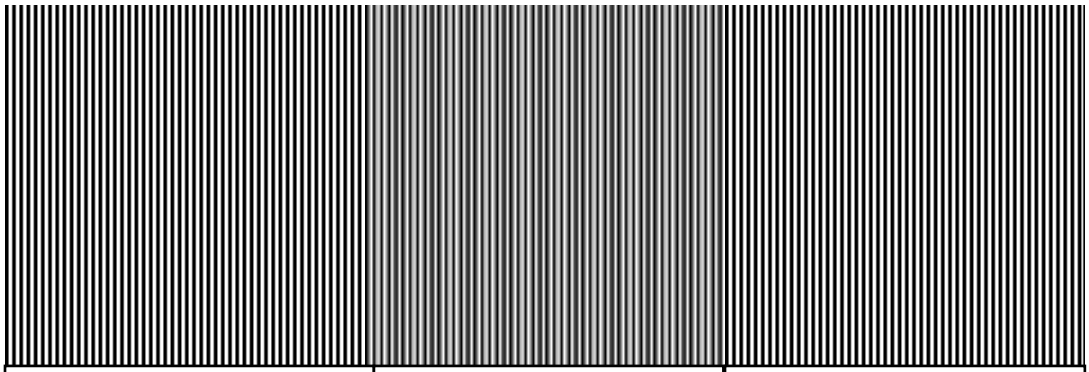


scaleNearest source img | --scaleNearest 0.7 | --scaleNearest 1

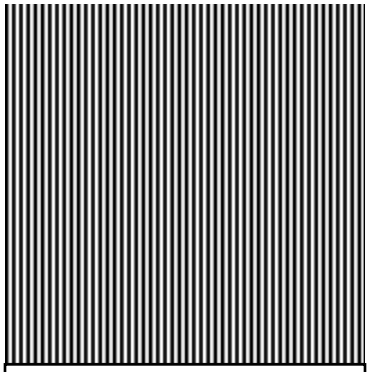


--scaleNearest 1.3

14. scaleBilinear



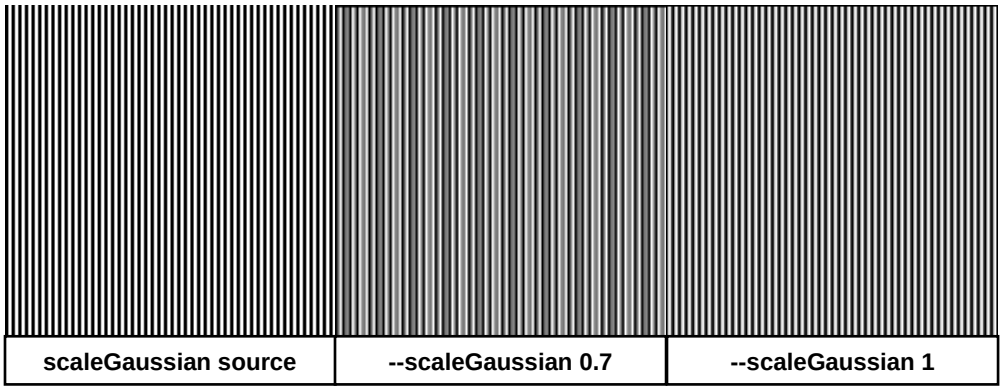
scaleBilinear source img | --scaleBilinear 0.7 | --scaleBilinear 1



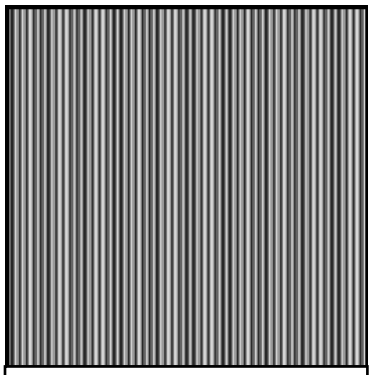
--scaleBilinear 1.3

15. scaleGaussian

* The images look different than sample results because radius and variance were 2 and 1 respectively. To my knowledge, gaussianSample was implemented correctly.

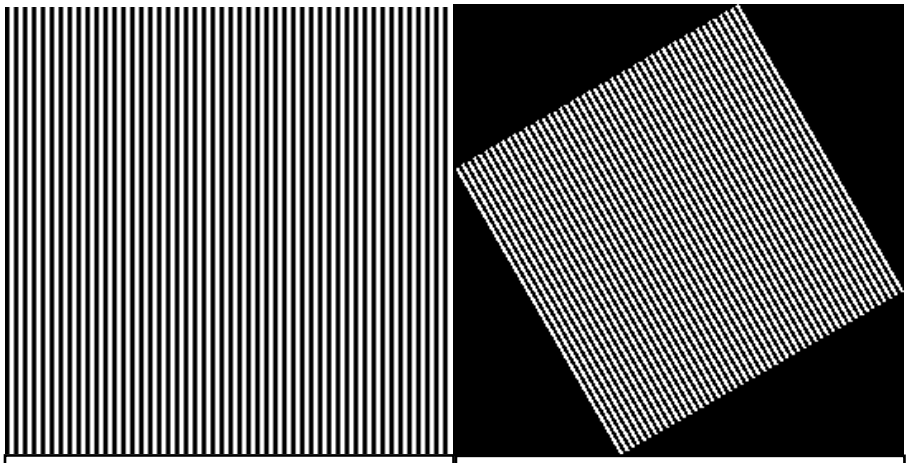


scaleGaussian source | --scaleGaussian 0.7 | --scaleGaussian 1



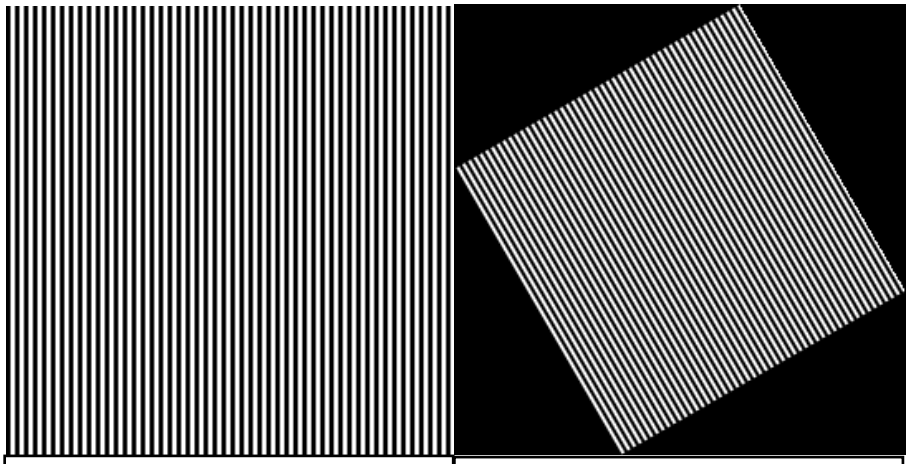
--scaleGaussian 1.3

16. rotateNearest



rotateNearest source img | **--rotateNearest 30**

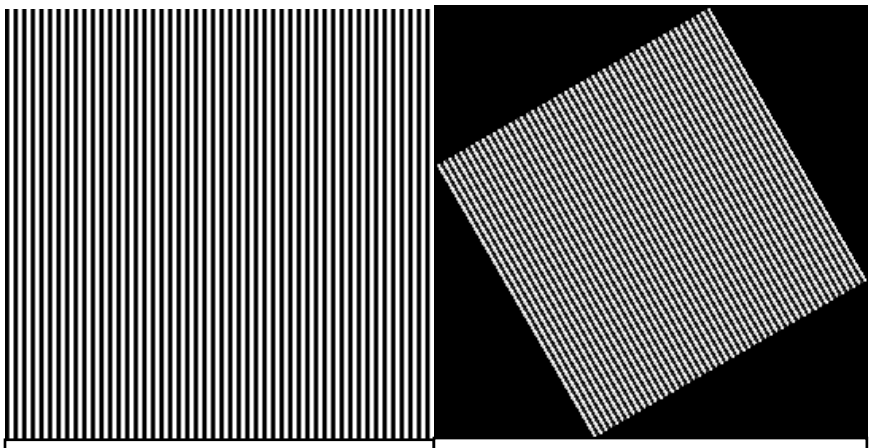
17. rotateBilinear



rotateBiliear source img | **--rotateBilinear 30**

18. rotateGaussian

* The images look different than sample results because radius and variance were 2 and 1 respectively. To my knowledge, gaussianSample was implemented correctly.

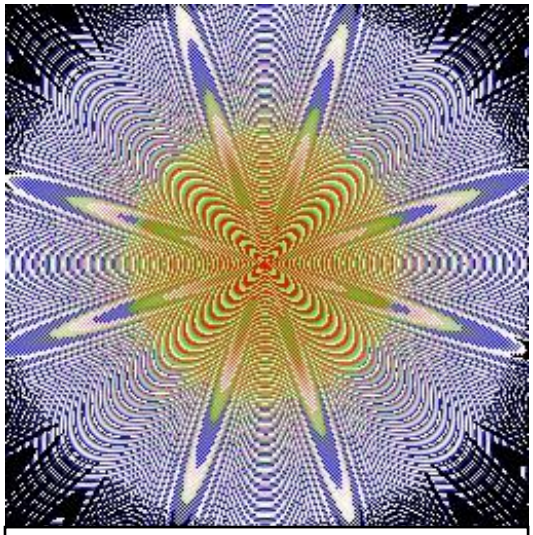


rotateGaussian source img	--rotateGaussian 30

19. funFilter

Fun source img	--fun (scale = 50)

Art Image and Movie:



Name: The Creation of the flat Earths

Commands used to generate the image:

./Assignment1 --fun --in Images/ramp.bmp --out Images/art.bmp (scale = 0.1 in funFilter, default scale = 50)
./Assignment1 –saturate 4 --in Images/ramp.bmp --out Images/art.bmp

Commands used to generate the GIF:

./Assignment1 --fun --in Images/yoda.bmp --out img<i>.jpeg

(with scale = 0.1, 50, 100, 150, 200, … , 1000 in funFilter. Here i is the image number)
* All jpeg images were combined and the gif was created using https://ezgif.com/jpg-to-gif.