giftsh: a DSL and shell for image transformations

Command list

`blur`	value > 0
`brightness`	value (-100, 100)
`colorbalance`	red green blue (-100, 500)
`colorize`	hue (0-360) saturation (0-100) value (0-100)
`colorspace`	l for linear->sRGB or s for sRGB->linear
`contrast`	value (-100, 100)
`crop`	x1 y1 x2 y2 (rectangle at (x1,y1) and (x2,y2)
`cropsize`	width height
`edge`	edge filter
`emboss`	emboss filter
`fliph`	flip horizontal
`flipv`	flip vertical
`gamma`	value (< 1 darken, > 1 lighten)
`gray`	grayscale image
`hue`	value (-180, 180)
`invert`	invert image
`max`	local maximum size (odd positive integer)
`mean`	local mean size (odd positive integer)
`median`	local median size (odd positive integer)
`min`	local minimum size (odd positive integer)
`opacity`	value (0-100)
`pixelate`	pixels
`resize`	width height
`resizefill`	width height
`resizefit`	width height
`rotate`	degrees counter-clockwise
`saturation`	value (-100, 500)
`sepia`	value (0-100)
`sigmoid`	midpoint (0,1) factor (-10,10)
`sobel`	sobel filter
`threshold`	color threshold percentage (0-100)
`transpose`	flip horizontally and rotate 90° counter-clockwise
`transverse`	flip vertically and rotate 90° counter-clockwise
`unsharp`	sigma (> 0) amount (0.5, 1.5) threshold (0, 0.05)

Transformations

original blur brightness colorbalance colorize colorspace-l colorspace-s contrast crop

cropsize edge emboss fliph flipv gamma gray hue invert

max mean median min opacity pixelate resizefill resizefit resize

rotate saturation sepia sigmoid sobel threshold transpose transverse unsharp

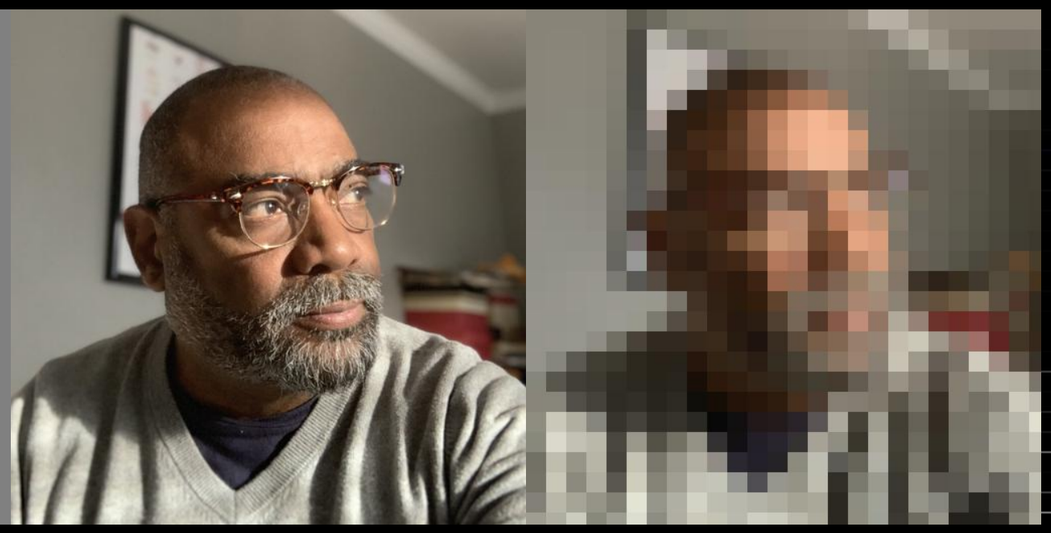
Install and run

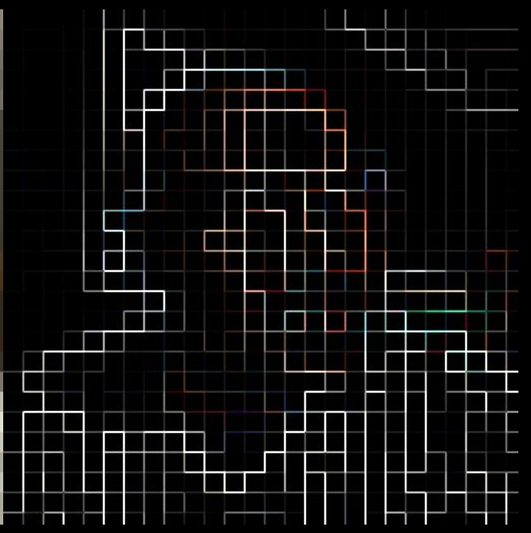
```
go install github.com/ajstarks/giftsh@latest
```

`giftsh`	commands from stdin, output to stdout
`giftsh < f.gsh > f.jpg`	commands from f.gsh, output to f.jpg
`giftsh -o f.jpg`	commands from stdin, output to f.jpg
`giftsh -c f.gsh`	commands from f.gsh, output to stdout
`giftsh -c f.gsh -o f.jpg`	commands from f.gsh, output to f.jpg
`giftsh -c f.gsh -w f.jpg`	commands from f.gsh, write after each command
`giftsh -h`	show help and command set

script → giftsh → result

```
read ajs.jpg
pixelate 10
sobel
```





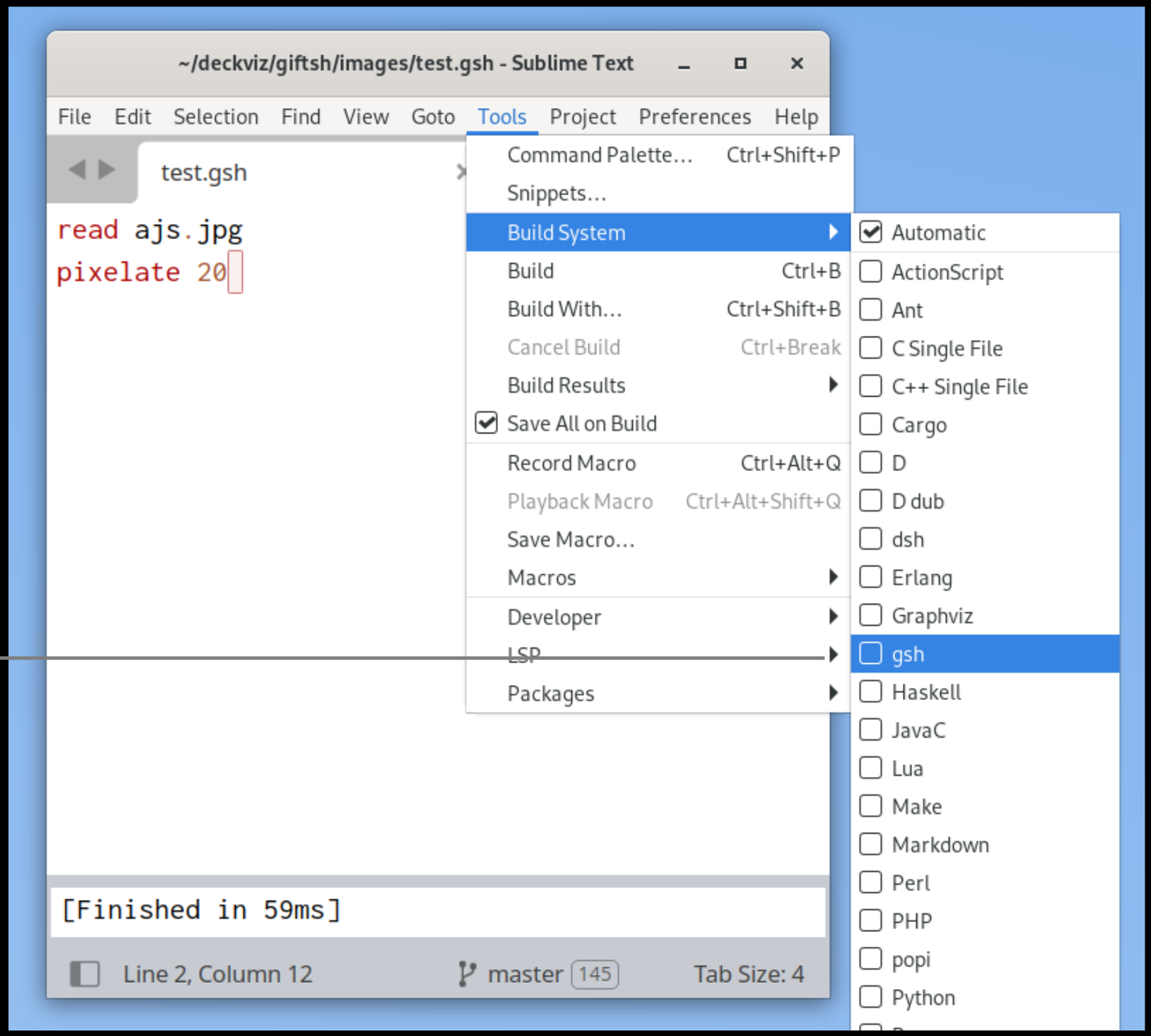
giftsh -c test.gsh -o test.jpg

Editor Setup (Sublime Text)

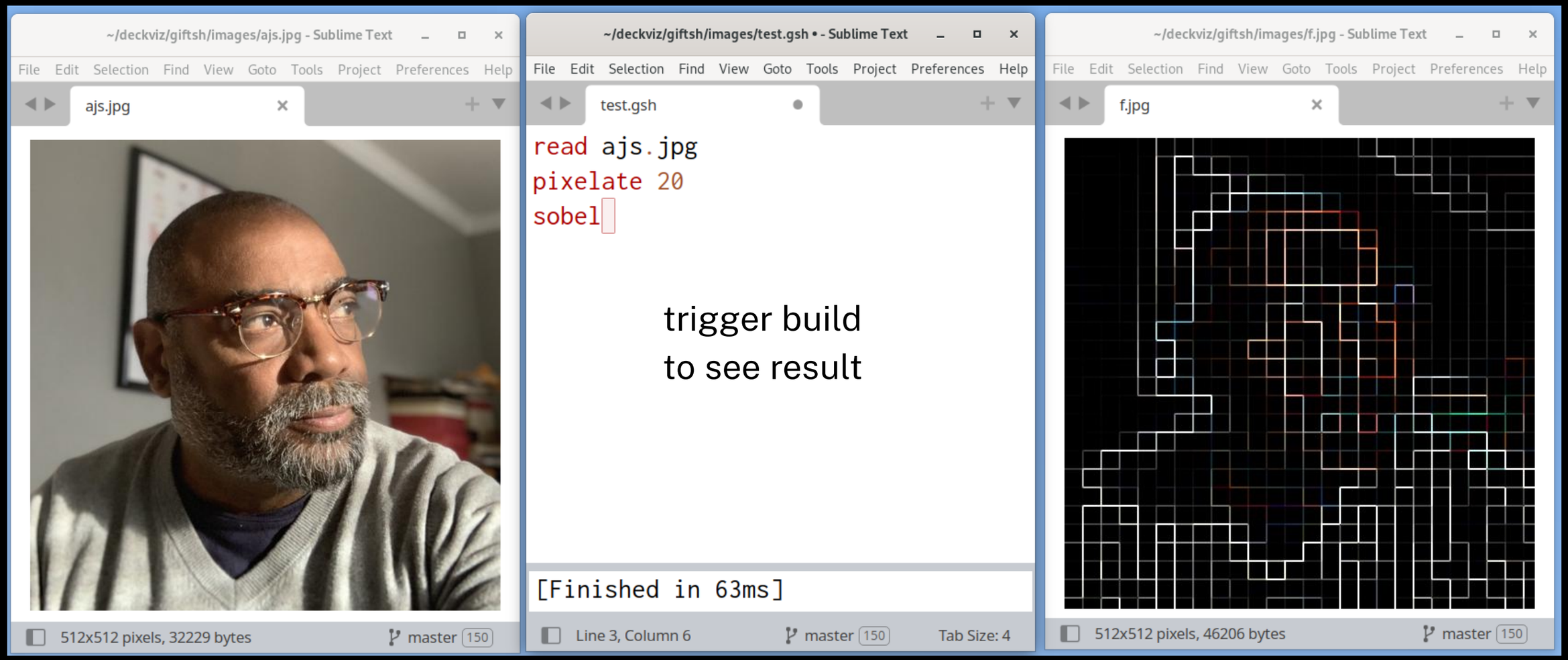
in <config-dir>/Packages/User/gsh.sublime-build:

```
{"shell_cmd": "giftsh < $file > f.jpg"}
```

Configure the build system



Running with the build system



original script result

Running with entr

~/deckviz/giftsh/images/ajs.jpg - Sublime Text

ajs.jpg

512x512 pixels, 32229 bytes master 2777

~/deckviz/giftsh/images/test.gsh - Sublime Text

test.gsh

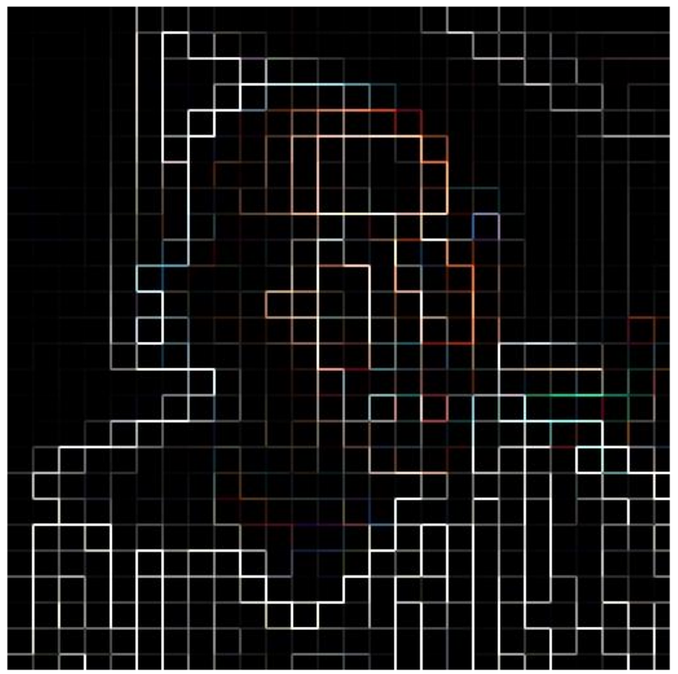
```
read ajs.jpg
pixelate 20
sobel
```

save file to
see result

Line 2, Column 11 master 2777 Tab Size: 4

~/deckviz/giftsh/images/f.jpg - Sublime Text

f.jpg



512x512 pixels, 46206 bytes master 2777

ajstarks@bcube:~/deckviz/giftsh/images

```
$ ls test.gsh | entr -s 'giftsh -c test.gsh -o f.jpg'
```

Running with entr and zed

test.gsh

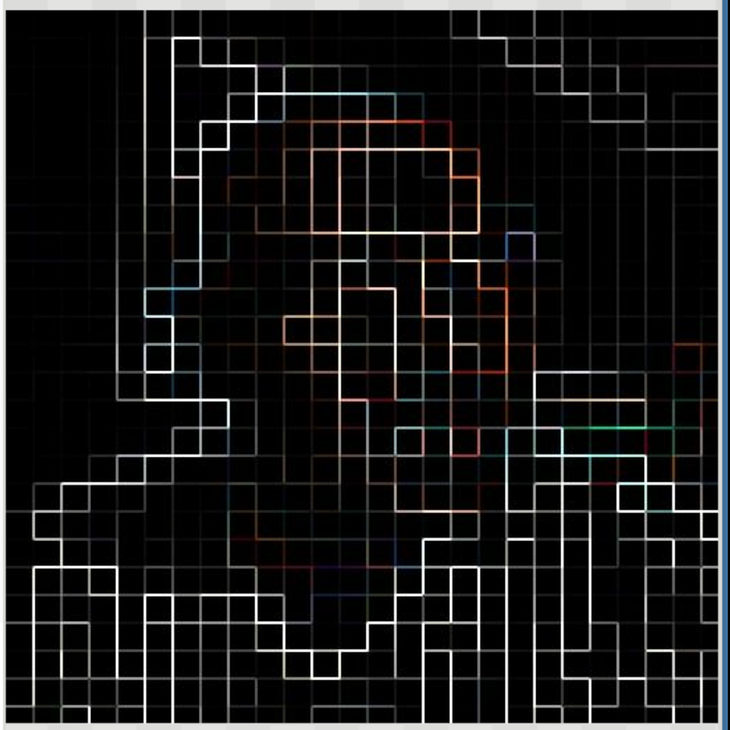
ajs.jpg

test.gsh

```
read ajs.jpg
pixelate 20
sobel
```

save file to
see result

f.jpg



images — bash

ajstarks@bcube:~/deckviz/giftsh/images

```
$ ls test.gsh | entr -s 'giftsh -c test.gsh -o f.jpg'
```

2:12 Unknown

Script to generate image transformations

```
#!/bin/sh
if test $# -ne 1
then
    echo "specify an image" 1>&2
    exit 1
fi
input="$1" ; type=`echo $input|awk -F. '{print $2}'`
(echo r $input; echo blur 10)                  | giftsh > blur.$type
(echo r $input; echo brightness 20)            | giftsh > brightness.$type
(echo r $input; echo colorbalance 200 0 0)     | giftsh > colorbalance.$type
(echo r $input; echo colorize 200 100 100 )    | giftsh > colorize.$type
(echo r $input; echo colorspace l)             | giftsh > colorspace-l.$type
(echo r $input; echo colorspace s)             | giftsh > colorspace-s.$type
(echo r $input; echo contrast 20)              | giftsh > contrast.$type
(echo r $input; echo crop 0 0 200 200)         | giftsh > crop.$type
(echo r $input; echo cropsize 100 100)         | giftsh > cropsize.$type
(echo r $input; echo edge)                     | giftsh > edge.$type
(echo r $input; echo emboss)                   | giftsh > emboss.$type
(echo r $input; echo fliph)                    | giftsh > fliph.$type
(echo r $input; echo flipv)                    | giftsh > flipv.$type
(echo r $input; echo gamma 2)                  | giftsh > gamma.$type
(echo r $input; echo gray)                     | giftsh > gray.$type
(echo r $input; echo hue 75)                   | giftsh > hue.$type
(echo r $input; echo invert)                   | giftsh > invert.$type
(echo r $input; echo max 3)                    | giftsh > max.$type
(echo r $input; echo mean 5)                   | giftsh > mean.$type
(echo r $input; echo median 5)                 | giftsh > median.$type
(echo r $input; echo min 5)                    | giftsh > min.$type
(echo r $input; echo opacity 60)               | giftsh > opacity.$type
(echo r $input; echo pixelate 50)              | giftsh > pixelate.$type
(echo r $input; echo resizefill 512 512)       | giftsh > resizefill.$type
(echo r $input; echo resizefit 512 512)        | giftsh > resizefit.$type
(echo r $input; echo resize 200 200)           | giftsh > resize.$type
(echo r $input; echo rotate 45)                | giftsh > rotate.$type
(echo r $input; echo saturation 200)           | giftsh > saturation.$type
(echo r $input; echo sepia 100)                | giftsh > sepia.$type
(echo r $input; echo sigmoid 0.5 0)            | giftsh > sigmoid.$type
(echo r $input; echo sobel)                    | giftsh > sobel.$type
(echo r $input; echo threshold 60)             | giftsh > threshold.$type
(echo r $input; echo transpose)                | giftsh > transpose.$type
(echo r $input; echo transverse)               | giftsh > transverse.$type
(echo r $input; echo unsharp 1 1 0.05)         | giftsh > unsharp.$type
```