

Index

A

Atom editor

- add plug-ins, 16
- automatic updated image, 19–20
- description, 16
- IDE
 - autocompletion, 19
 - clean layout, 18
 - standard layout, 17–18
- installing, 16
- java language plug-in, 17
- submat function, 20

B

- Bilateral filter, 170–172
- Blur detection, 269–272
- Blurring images
 - bilateral filter, 170–172
 - gaussian blur, 169
 - main methods, 167
 - median blur, 168, 169, 172, 173
 - simple blur, 168–169

C, D, E

Canny

- bitwise_not function, 42
- black-and-white picture, 41

- cvtColor function, 40–41
 - description, 40
- drawContours, 47, 49–51
- findContours, 47–48
- function, 41
- gradient value, 42
- inverted tools, 43
- kitten pictures, 43–44
- mask, 44–46
- parameters, 41
- tools, 41, 42

Canvas effect

- applying, 224–226
- problem, 221
- solution, 221

Cartoons

- bilateral filter, 207–208
- grayed with update
 - channel, 209–211
- problem, 201
- solution
 - bilateral filter
 - function, 205–206
 - canny function, 201–203
 - Gaussian blur, 204

Circles, highlighting

- draw segment, 238–239
- finding circles, 236–237

INDEX

Circles, highlighting (*cont.*)

- hough-lines-p, 230–232
- pockets on pool table, 233–235

Colors, playing

- applying threshold, 177–186
- artful transformations, 198, 200, 201

channels by hand

- artful cat, 192
- blue-filtered cat, 191
- cyan cat, 190
- merge function, 187–188
- opencv split function, 187
- transformation, 189
- update-channel!
function, 189

problem, 176

solution, 177

transform, 193–197

F

Face recognition

- classifier, 293
- cool face, 296
- draw-rects!, 296
- features, 293
- find-faces!, 295
- new-cascadeclassifier, 294
- overview and close-up, 297
- quiet impressive morning
coffee face, 295
- rectangle objects, 294
- xml file, 294

Filtering methods

- bilateral, 170–172
- filter-2-d, 154–157
- highlight, 153
- luminosity, 148, 150, 152
- manual, 148–150
- multiply, 148, 151
- threshold, 158–161

G

Gaussian blur, 169

Gorilla

- block of code, 103
- in browser, 105
- clojure code, 103–104
- instant gratification, 105
- key shortcuts, 104
- notebook, 101–102
- project.clj file, 101
- REPL, 106
- text mode, 102–103

Grabcut

- description, 308
- live stream solution, 308
- still image
 - background mask, 313
 - cat picture, 309, 310
 - foreground mask, 311–312
 - foreground of feline kiss, 312
 - GC_INIT_WITH_MASK, 310
 - grabcut init param, 310
 - mask mat, 310
 - mat of background layer, 314

- opencv compare, 309
- rectangle, 309
- region of interest, 310
- submat mask, 310
- video stream
 - in-front function, 316
 - in-front-slow function, 314
 - lower-resolution mat, 315
 - simple-cam-window, 316
 - slow, slow, slow, 315
 - speed, 316, 317
 - u/simple-cam-window, 315

H

- Hue-Saturation-Value (HSV),
133–136

I, J

- Imaging techniques

- blur detection
 - problem, 269
 - solution, 269
 - working, 270–272

- canvas effect
 - applying, 224–226
 - problem, 221
 - solution, 221

- cartoons
 - bilateral filter
 - function, 205–208
 - canny function, 201–203
 - gaussian blur, 204

- grayed with update
 - channel, 209–211
 - problem, 201
 - solution, 201
- colors, playing
 - applying threshold, 177–186
 - artful transformations, 198,
200, 201
 - channels by hand, 187–192
 - problem, 176
 - solution, 177
 - transform, 193–197
- contours
 - approx-poly-dp function, 252
 - hand-drawn
 - shapes, 256–260
 - highlight, 252–256
 - problem, 252
 - solution, 252
- contours and bounding boxes
 - hot-air balloons, 247–251
 - problem, 239
 - solution, 240
 - Sony headphones, 240–247
- highlight lines and circles
 - draw segment, 238–239
 - finding circles, 236–238
 - hough-lines-p, 230–232
 - pockets on pool table,
233–235
 - problem, 226
 - solution, 226
 - tennis court with
 - hough-lines, 227–230

INDEX

Imaging techniques (*cont.*)

- pencil sketches
 - background, 213–215
 - foreground and result, 216–218
 - problem, 211
 - solution, 211
- photomosaics
 - problem, 272
 - solution, 272–273
 - working, 273–278
- shapes
 - problem, 260
 - solution, 261
 - working, 262–265
- trees
 - problem, 266
 - solution, 266
 - working, 266–269

K

Kitten image

- cropping, submat
 - autocompletion, 31
 - blue cat, 34
 - blue color, 34
 - blur effect, 32–33
 - imread, 31
 - Mat object, 30–32
 - parameters, 31
 - println, 31
 - ranges, 33
 - Rect object, 34
 - sub-cat, 32

loading

- black-and-white mode, 26
- google search, 25
- gray scale, 26
- image reading options, 27
- imread function, 24–25
- Mat object, 26
- reduced size, 27
- running kitten, 25

saving

- imwrite, 28
- JPEG format, 28–29
- parameters, 29
- PNG format, 29

Kotlin

color mapping

- applyColorMap
 - function, 68
- bone cat, 70
- hot cat, 70
- original cat image, 69
- winter cat, 70

command output, 68

First.kt file, 67

Mat object and dump, 67

project.clj file, 66

snippet, 67

UI (*see* User interface (UI))

L

Leiningen

- auto plug-in, 13–14
- compiled code, 2

- description, 2
- file changes, 14
- installing
 - Java 8, 3
 - macOS and Unix, 3–4
 - Windows, 4
- java program
 - HelloCv.java, 11–12
 - opencv and opencv-native
 - dependencies, 11
 - project template, 9–11
- java project
 - directory
 - structure, 5–7
 - Hello2.java file, 8
 - metadata, 6
 - project.clj, 5
- lein-auto plug-in, 14
- Mat object, 15
- template
 - file structure, 86
 - gorilla plug-in, 88
 - myfirstcljcv folder, 85
 - REPL, 93, 96
- Lines and circles, highlighting
 - draw segment, 238–239
 - finding circles, 236–238
 - hough-lines-p, 230–232
 - pockets on pool
 - table, 233–235
 - problem, 226
 - solution, 226
 - tennis court with
 - hough-lines, 227–230

M, N

Mats

- bright cat, 122
- create, 107–108
- create colored mat, 108–109
- create random gray, 113–114
- dark cat, 122
- default keys, 121
- filtering methods
 - filter-2-d, 154–157
 - highlight, 153
 - luminosity, 148, 150, 152
 - manual, 148–150
 - multiply, 148, 151
 - threshold, 158–161
- flipping, 140–141
- framed cat, 121
- gray gradient, 116
- grey-neko.png, 120
- loading from URL, 123–124
- not a cat, 118
- not a gray cat, 119
- piping process, 113–116
- rotate function, 139
- rotation
 - create matrix, 143–144
 - 45-degrees-rotated, 144
 - parameter, 141–142
 - pipeline, 145
 - rotate-by! function, 145
 - using hconcat!, 142
 - using vconcat!, 143
 - zoom parameter, 146

INDEX

Mats (*cont.*)

- setting pixel color, 111–113
- submat, 109–110
- warp-affine function, 139, 143–147

Median blur, 172–173

O

Open Source Computer Vision (OpenCV)

- Canny (*see* Canny)
- definition, 1
- Github's Atom editor (*see* Atom editor)
- Kotlin, 66
- leiningen (*see* Leiningen)
- Mat object, 1–2
 - channels per pixel, 22–23
 - colored submats, 35–37
 - float values, 24
 - picture submats, 38–40
 - setTo and copyTo, 35
 - static functions, 21–22
- Scala, 57
- threshold function, 179
- video streaming, 51

Origami, 176

- atom, REPL from, 96–100
- auto plug-in, 91–93
- clj-opencv template, 85
- color maps
 - apply-color-map function, 128
 - sepia cat, 131–132

standard, 129

- three-channel mode, 129–130

color space

- advantages, 134
- HSV, 133, 135, 136
- linear intensity, 134
- mask mat, 138
- natural red cat, 137
- red in RGB, 133
- switch, 132

DSL, 90

gorilla

- block of code, 103
- in browser, 105
- clojure code, 103–104
- instant gratification, 105
- key shortcuts, 104
- notebook, 101–102
- project.clj file, 101
- REPL, 106
- text mode, 102–103

grey-neko.jpg file, 88–89

leiningen template

- file structure, 86
- gorilla plug-in, 88
- myfirstcljcv folder, 85
- REPL, 93, 96

library, 83

mats (*see* Mats)

opencv3.ok namespace, 89

REPL, 93–100

RGB colors, 126–127

webcam, 90–91

P, Q

Pencil sketches

- background, 213–215
- foreground and result, 216–218
- problem, 211
- solution, 211

Photomosaics

- problem, 272
- solution, 272–273
- working, 273–278

Piping process, 113–116

Pixel color, 111–113

Portable Network Graphic (PNG)

- format, 29

R

Read-Eval-Print-Loop (REPL)

- from atom, 96–100
- cute cat, 95
- gray version, 95–96

Rose masking techniques

- bitwise-and, 163–164
- brightened version, 164–165
- convert from RGB to HSV, 162
- convert-to, 164
- copy-to, 165
- with imread, 161
- saturation and luminosity, 163

S

Scala

- blurring effect, 62–63
- canny effect, 63–65

code compilation, 60

Leiningen plug-in, 58

loadLibrary, 60

project.clj, 58

project directory
structure, 59

setup, 61

SimpleOpenCV class, 60

Shapes, moving

problem, 260

solution, 261

working, 262–265

T

Threshold filtering, 158–161

U

User interface (UI)

blurring application,
76, 78, 80, 81

Kotlin setup, 71

reactive buttons

action handler, 74

counter application, 76

SimpleIntegerProperty, 74

tornadofx application

classes, 72

graph, 72

graphical frame, 73

javafx boilerplate, 73

javafx widgets, 72

library, 71

V

Video streaming

background diffing

absdiff, 299

body soap, 299, 302

cam-window, 298

Clojure atom, 298

comp, 300

hearts and speaker, 299

highlight-new! function, 301

shape of body soap, 301

buffer, display loop, 283

camera_id, 52

camera.read()function, 52

colormap, 283

combining functions, 288–290

do_still_captures function, 52

face recognition (*see* Face
recognition)

finding movement

accumulate-weighted
function, 302

black and white, 303–306

contours, 306–308

finding orange

find-contours, 317, 322

hough-circles, 317–321

functions, 280

grabcut (*see* Grabcut)

image finding, body soap

background, 326

Carmen, 325

detector and extractor, 325

distance value, 328

draw-matches

method, 329–330

Eucalyptus scent, 324

feature extraction, 324

keypoints, 326–327

matching step, 328

OpenCV objects, 323

parameters, 329

Petit Marseillais, 324

picture, 331

where-is-my-body-soap!

function, 330

namespace, 281

Origami setup, 280

read function, 282

real time

BufferedImage, 54

canny picture, 57

do_still_captures

method, 55

Japanese room, 56

Mat objects, 55

MatPanel, 55

MatToBufferedImage, 54–55

release function, 282

release method, 52

soap, 282

Starbucks, 286

transformation function, 285–286

two/more frames,

same source, 287, 288

u/imshow, 281
u/simple-cam-window, 280,
283, 284
VideoCapture object, 51, 52,
280–281
video folder, 53
warping, 290–293

webcams, 281
window, 281

W, X, Y, Z

Warping video, 290–293
Webcam, 90–91