
MPCam's imutils

Release 0.2.0

Adrian Rosebrock

Apr 04, 2024

CONTENTS

1	Display	3
1.1	Display Stream	3
1.2	Screen Output	3
1.3	Web Output	3
2	Face Utils	5
2.1	Face Aligner	5
2.2	Helpers	5
3	Face Utils	7
3.1	Dense	7
3.2	Factories	7
3.3	Gfft	8
3.4	Harris	8
3.5	Helpers	8
3.6	Rootsift	8
4	IO	9
4.1	Temp file	9
5	Video	11
5.1	Count Frames	11
5.2	File Video Stream	11
5.3	FSP	11
5.4	Image Output	11
5.5	Pi Video Stream	11
5.6	Video Stream	11
5.7	Webcam Video Stream	11
6	Imutils	13
6.1	Contours	13
6.2	Convenience	13
6.3	Encodigns	14
6.4	Meta	14
6.5	Object Detection	14
6.6	Paths	15
6.7	Perspective	15
6.8	Text	15
	Python Module Index	17

A fork of [imutils](#) from PyImageSearch with new add-ons for the MPCam.

Contents:

DISPLAY

1.1 Display Stream

1.2 Screen Output

1.3 Web Output

FACE UTILS

2.1 Face Aligner

```
class imutils.face_utils.facealigner.FaceAligner(predictor, desiredLeftEye=(0.35, 0.35),
                                                  desiredFaceWidth=256, desiredFaceHeight=None)

    align(image, gray, rect)
```

2.2 Helpers

```
imutils.face_utils.helpers.rect_to_bb(rect)

imutils.face_utils.helpers.shape_to_np(shape, dtype='int')

imutils.face_utils.helpers.visualize_facial_landmarks(image, shape, colors=None, alpha=0.75)
```


3.1 Dense

```
class imutils.feature.dense.DENSE(step=6, radius=0.5)
```

```
    detect(img)
```

```
    setInt(var, val)
```

3.2 Factories

```
imutils.feature.factories.DescriptorExtractor_create(extractor, *args, **kw_args)
```

Parameters

- **extractor** – string of the type of descriptor extractor to return
- **args** – positional arguments for extractor
- **kw_args** – keyword arguments for extractor

Returns

the key extractor object

```
imutils.feature.factories.DescriptorMatcher_create(matcher)
```

Parameters

matcher – string of the type of descriptor matcher to return

Returns

the matcher int

```
imutils.feature.factories.FeatureDetector_create(detector, *args, **kw_args)
```

Parameters

- **detector** – string of the type of keypoint detector to return
- **args** – positional arguments for detector
- **kw_args** – keyword arguments for detector

Returns

the key point detector object

3.3 Gfft

```
class imutils.feature.gfft.GFFT(maxCorners=0, qualityLevel=0.01, minDistance=1, mask=None,  
                                blockSize=3, useHarrisDetector=False, k=0.04)
```

```
    detect(img)
```

3.4 Harris

```
class imutils.feature.harris.HARRIS(blockSize=2, apertureSize=3, k=0.1, T=0.02)
```

```
    detect(img)
```

3.5 Helpers

```
imutils.feature.helpers.corners_to_keypoints(corners)
```

function to take the corners from cv2.GoodFeaturesToTrack and return cv2.KeyPoints

3.6 Rootsift

```
class imutils.feature.rootsift.RootSIFT
```

```
    compute(image, kps, eps=1e-07)
```

4.1 Temp file

```
class imutils.io.tempfile.TempFile(basePath='./', ext='.jpg')
    cleanup()
```


5.1 Count Frames

5.2 File Video Stream

5.3 FSP

5.4 Image Output

5.5 Pi Video Stream

5.6 Video Stream

5.7 Webcam Video Stream

6.1 Contours

```
imutils.contours.label_contour(image, c, i, color=(0, 255, 0), thickness=2)
```

```
imutils.contours.sort_contours(cnts, method='left-to-right')
```

6.2 Convenience

```
imutils.convenience.adjust_brightness_contrast(image, brightness=0.0, contrast=0.0)
```

Adjust the brightness and/or contrast of an image

Parameters

- **image** – OpenCV BGR image
- **contrast** – Float, contrast adjustment with 0 meaning no change
- **brightness** – Float, brightness adjustment with 0 meaning no change

```
imutils.convenience.auto_canny(image, sigma=0.33)
```

```
imutils.convenience.build_montages(image_list, image_shape, montage_shape)
```

6.2.1 author: Kyle Hounslow

Converts a list of single images into a list of ‘montage’ images of specified rows and columns. A new montage image is started once rows and columns of montage image is filled. Empty space of incomplete montage images are filled with black pixels

_____ :param
image_list: python list of input images :param image_shape: tuple, size each image will be resized to for display (width, height) :param montage_shape: tuple, shape of image montage (width, height) :return: list of montage images in numpy array format _____

example usage:

```
# load single image img = cv2.imread('lena.jpg') # duplicate image 25 times num_imgs = 25 img_list = [] for i in xrange(num_imgs):
```

```
    img_list.append(img)
```

```
# convert image list into a montage of 256x256 images tiled in a 5x5 montage montages =  
make_montages_of_images(img_list, (256, 256), (5, 5)) # iterate through montages and display for montage  
in montages:
```

```
cv2.imshow('montage image', montage) cv2.waitKey(0)
```

```
imutils.convenience.check_opencv_version(major, lib=None)
imutils.convenience.get_opencv_major_version(lib=None)
imutils.convenience.grab_contours(cnts)
imutils.convenience.is_cv2(or_better=False)
imutils.convenience.is_cv3(or_better=False)
imutils.convenience.is_cv4(or_better=False)
imutils.convenience.opencv2matplotlib(image)
imutils.convenience.resize(image, width=None, height=None, inter=3)
imutils.convenience.rotate(image, angle, center=None, scale=1.0)
imutils.convenience.rotate_bound(image, angle)
imutils.convenience.skeletonize(image, size, structuring=0)
imutils.convenience.translate(image, x, y)
imutils.convenience.url_to_image(url, readFlag=1)
```

6.3 Encodigns

```
imutils.encodings.base64_decode_array(a, dtype)
imutils.encodings.base64_decode_image(a)
imutils.encodings.base64_encode_array(a)
imutils.encodings.base64_encode_image(a)
```

6.4 Meta

```
imutils.meta.find_function(name, pretty_print=True, module=None)
```

6.5 Object Detection

```
imutils.object_detection.non_max_suppression(boxes, probs=None, overlapThresh=0.3)
```

6.6 Paths

`imutils.paths.list_files(basePath, validExts=None, contains=None)`

`imutils.paths.list_images(basePath, contains=None)`

6.7 Perspective

6.8 Text

`imutils.text.put_centered_text(img, text, font_face, font_scale, color, thickness=1, line_type=8)`

Utility for drawing vertically & horizontally centered text with line breaks

Parameters

- **img** – Image.
- **text** – Text string to be drawn.
- **font_face** – Font type. One of FONT_HERSHEY_SIMPLEX, FONT_HERSHEY_PLAIN, FONT_HERSHEY_DUPLEX, FONT_HERSHEY_COMPLEX, FONT_HERSHEY_TRIPLEX, FONT_HERSHEY_COMPLEX_SMALL, FONT_HERSHEY_SCRIPT_SIMPLEX, or FONT_HERSHEY_SCRIPT_COMPLEX, where each of the font ID's can be combined with FONT_ITALIC to get the slanted letters.
- **font_scale** – Font scale factor that is multiplied by the font-specific base size.
- **color** – Text color.
- **thickness** – Thickness of the lines used to draw a text.
- **line_type** – Line type. See the line for details.

Returns

None; image is modified in place

`imutils.text.put_text(img, text, org, font_face, font_scale, color, thickness=1, line_type=8, bottom_left_origin=False)`

Utility for drawing text with line breaks

Parameters

- **img** – Image.
- **text** – Text string to be drawn.
- **org** – Bottom-left corner of the first line of the text string in the image.
- **font_face** – Font type. One of FONT_HERSHEY_SIMPLEX, FONT_HERSHEY_PLAIN, FONT_HERSHEY_DUPLEX, FONT_HERSHEY_COMPLEX, FONT_HERSHEY_TRIPLEX, FONT_HERSHEY_COMPLEX_SMALL, FONT_HERSHEY_SCRIPT_SIMPLEX, or FONT_HERSHEY_SCRIPT_COMPLEX, where each of the font ID's can be combined with FONT_ITALIC to get the slanted letters.
- **font_scale** – Font scale factor that is multiplied by the font-specific base size.
- **color** – Text color.

- **thickness** – Thickness of the lines used to draw a text.
- **line_type** – Line type. See the line for details.
- **bottom_left_origin** – When true, the image data origin is at the bottom-left corner. Otherwise, it is at the top-left corner.

Returns

None; image is modified in place

PYTHON MODULE INDEX

i

- `imutils.contours`, [13](#)
- `imutils.convenience`, [13](#)
- `imutils.encodings`, [14](#)
- `imutils.face_utils.facealigner`, [5](#)
- `imutils.face_utils.helpers`, [5](#)
- `imutils.feature.dense`, [7](#)
- `imutils.feature.factories`, [7](#)
- `imutils.feature.gftt`, [8](#)
- `imutils.feature.harris`, [8](#)
- `imutils.feature.helpers`, [8](#)
- `imutils.feature.rootsift`, [8](#)
- `imutils.io.tempfile`, [9](#)
- `imutils.meta`, [14](#)
- `imutils.object_detection`, [14](#)
- `imutils.paths`, [15](#)
- `imutils.text`, [15](#)

A

`adjust_brightness_contrast()` (in module *imutils.convenience*), 13
`align()` (*imutils.face_utils.facealigner.FaceAligner* method), 5
`auto_canny()` (in module *imutils.convenience*), 13

B

`base64_decode_array()` (in module *imutils.encodings*), 14
`base64_decode_image()` (in module *imutils.encodings*), 14
`base64_encode_array()` (in module *imutils.encodings*), 14
`base64_encode_image()` (in module *imutils.encodings*), 14
`build_montages()` (in module *imutils.convenience*), 13

C

`check_opencv_version()` (in module *imutils.convenience*), 14
`cleanup()` (*imutils.io.tempfile.TempFile* method), 9
`compute()` (*imutils.feature.rootsift.RootSIFT* method), 8
`corners_to_keypoints()` (in module *imutils.feature.helpers*), 8

D

DENSE (class in *imutils.feature.dense*), 7
`DescriptorExtractor_create()` (in module *imutils.feature.factories*), 7
`DescriptorMatcher_create()` (in module *imutils.feature.factories*), 7
`detect()` (*imutils.feature.dense.DENSE* method), 7
`detect()` (*imutils.feature.gftt.GFTT* method), 8
`detect()` (*imutils.feature.harris.HARRIS* method), 8

F

FaceAligner (class in *imutils.face_utils.facealigner*), 5
`FeatureDetector_create()` (in module *imutils.feature.factories*), 7
`find_function()` (in module *imutils.meta*), 14

G

`get_opencv_major_version()` (in module *imutils.convenience*), 14
GFTT (class in *imutils.feature.gftt*), 8
`grab_contours()` (in module *imutils.convenience*), 14

H

HARRIS (class in *imutils.feature.harris*), 8

I

imutils.contours
 module, 13
imutils.convenience
 module, 13
imutils.encodings
 module, 14
imutils.face_utils.facealigner
 module, 5
imutils.face_utils.helpers
 module, 5
imutils.feature.dense
 module, 7
imutils.feature.factories
 module, 7
imutils.feature.gftt
 module, 8
imutils.feature.harris
 module, 8
imutils.feature.helpers
 module, 8
imutils.feature.rootsift
 module, 8
imutils.io.tempfile
 module, 9
imutils.meta
 module, 14
imutils.object_detection
 module, 14
imutils.paths
 module, 15
imutils.text
 module, 15

`is_cv2()` (in module *imutils.convenience*), 14
`is_cv3()` (in module *imutils.convenience*), 14
`is_cv4()` (in module *imutils.convenience*), 14

L

`label_contour()` (in module *imutils.contours*), 13
`list_files()` (in module *imutils.paths*), 15
`list_images()` (in module *imutils.paths*), 15

M

module
 imutils.contours, 13
 imutils.convenience, 13
 imutils.encodings, 14
 imutils.face_utils.facealigner, 5
 imutils.face_utils.helpers, 5
 imutils.feature.dense, 7
 imutils.feature.factories, 7
 imutils.feature.gftt, 8
 imutils.feature.harris, 8
 imutils.feature.helpers, 8
 imutils.feature.rootsift, 8
 imutils.io.tempfile, 9
 imutils.meta, 14
 imutils.object_detection, 14
 imutils.paths, 15
 imutils.text, 15

N

`non_max_suppression()` (in module *imutils.object_detection*), 14

O

`opencv2matplotlib()` (in module *imutils.convenience*), 14

P

`put_centered_text()` (in module *imutils.text*), 15
`put_text()` (in module *imutils.text*), 15

R

`rect_to_bb()` (in module *imutils.face_utils.helpers*), 5
`resize()` (in module *imutils.convenience*), 14
RootSIFT (class in *imutils.feature.rootsift*), 8
`rotate()` (in module *imutils.convenience*), 14
`rotate_bound()` (in module *imutils.convenience*), 14

S

`setInt()` (*imutils.feature.dense.DENSE* method), 7
`shape_to_np()` (in module *imutils.face_utils.helpers*), 5
`skeletonize()` (in module *imutils.convenience*), 14
`sort_contours()` (in module *imutils.contours*), 13

T

TempFile (class in *imutils.io.tempfile*), 9
`translate()` (in module *imutils.convenience*), 14

U

`url_to_image()` (in module *imutils.convenience*), 14

V

`visualize_facial_landmarks()` (in module *imutils.face_utils.helpers*), 5