

原创

FelixFuu

于 2019-05-17 15:48:33 发布

2559

收藏 6

版权

分类专栏：

C/C++

Image Processing

Data structures and Algorithms



C/C++

同时被 3 个专栏收录

因为要毕业和实习的一些缘故，很久没有写博客了，其实有很多东西需要整理，包括近期一些detection的方法，以及时序序列分析方面的算法和人脸识别等的模型及损失函数等。

这里先记录一下similarityTransform的C++实现。这个主要是用在一些需要做校正的场景下，比如face align等。在python里面有很全的包可以直接使用，比如

```
1 from skimage import transform as trans
2
3 tform = trans.SimilarityTransform()
4 tform.estimate(dst, src)
5 M = tform.params
```

得到矩阵M，而后做仿射变换

```
1 arped = cv2.warpAffine(img,M,(w,h))
```

其C++实现如下：

```
1 namespace FacePreprocess {
2
3     cv::Mat meanAxis0(const cv::Mat &src)
4     {
5         int num = src.rows;
6         int dim = src.cols;
7
8         // x1 y1
9         // x2 y2
10
11         cv::Mat output(1,dim,CV_32F);
12         for(int i = 0 ; i < dim; i ++){
13             {
14                 float sum = 0 ;
15                 for(int j = 0 ; j < num ; j++){
16                     {
17                         sum+=src.at<float>(j,i);
18                     }
19                     output.at<float>(0,i) = sum/num;
20                 }
21
22                 return output;
23             }
24
25         cv::Mat elementwiseMinus(const cv::Mat &A,const cv::Mat &B)
26         {
27             cv::Mat output(A.rows,A.cols,A.type());
28
29             assert(B.cols == A.cols);
30             if(B.cols == A.cols)
31             {
32                 for(int i = 0 ; i < A.rows; i ++){
33                     {
34                         for(int j = 0 ; j < B.cols; j++){
35                             {
36                                 output.at<float>(i,j) = A.at<float>(i,j) - B.at<float>(0,j);
```

```






37     }
38     }
39     }
40     return output;
41 }
42
43
44 cv::Mat varAxis0(const cv::Mat &src)
45 {
46     cv::Mat temp_ = elementwiseMinus(src, meanAxis0(src));
47     cv::multiply(temp_, temp_, temp_);
48     return meanAxis0(temp_);
49 }
50
51
52
53
54 int MatrixRank(cv::Mat M)
55 {
56     cv::Mat w, u, vt;
57     cv::SVD::compute(M, w, u, vt);
58     cv::Mat1b nonZeroSingularValues = w > 0.0001;
59     int rank = countNonZero(nonZeroSingularValues);
60     return rank;
61 }
62
63
64 // References
65 // -----
66 // .. [1] "Least-squares estimation of transformation parameters between two
67 // point patterns", Shinji Umeyama, PAMI 1991, DOI: 10.1109/34.88573
68 //
69 // ""
70 //
71 // Author: Jack Yu
72 cv::Mat similarTransform(cv::Mat src, cv::Mat dst) {
73     int num = src.rows;
74     int dim = src.cols;
75     cv::Mat src_mean = meanAxis0(src);
76     cv::Mat dst_mean = meanAxis0(dst);
77     cv::Mat src_demean = elementwiseMinus(src, src_mean);
78     cv::Mat dst_demean = elementwiseMinus(dst, dst_mean);
79     cv::Mat A = (dst_demean.t() * src_demean) / static_cast<float>(num);
80     cv::Mat d(dim, 1, CV_32F);
81     d.setTo(1.0f);
82     if (cv::determinant(A) < 0) {
83         d.at<float>(dim - 1, 0) = -1;
84     }
85 }
86 cv::Mat T = cv::Mat::eye(dim + 1, dim + 1, CV_32F);
87 cv::Mat U, S, V;
88 cv::SVD::compute(A, S, U, V);
89
90 // the SVD function in opencv differ from scipy .
91
92
93 int rank = MatrixRank(A);
94 if (rank == 0) {
95     assert(rank == 0);
96 }
97 else if (rank == dim - 1) {
98     if (cv::determinant(U) * cv::determinant(V) > 0) {
99         T.rowRange(0, dim).colRange(0, dim) = U * V;
100     } else {
101         // s = d[dim - 1]
102         // d[dim - 1] = -1
103         // T[:dim, :dim] = np.dot(U, np.dot(np.diag(d), V))

```

```
104 // d[dim - 1] = s
105     int s = d.at<float>(dim - 1, 0) = -1;
106     d.at<float>(dim - 1, 0) = -1;
107
108     T.rowRange(0, dim).colRange(0, dim) = U * V;
109     cv::Mat diag_ = cv::Mat::diag(d);
110     cv::Mat twp = diag_*V; //np.dot(np.diag(d), V.T)
111     c v::Mat B = cv::Mat::zeros(3, 3, CV_8UC1);
112     c v::Mat C = B.diag(0);
113     T.rowRange(0, dim).colRange(0, dim) = U* twp;
114     d.at<float>(dim - 1, 0) = s;
115 }
116 }
117 else{
118     cv::Mat diag_ = cv::Mat::diag(d);
119     cv::Mat twp = diag_*V.t(); //np.dot(np.diag(d), V.T)
120     cv::Mat res = U* twp; // U
121     T.rowRange(0, dim).colRange(0, dim) = -U.t()* twp;
122 }
123 cv::Mat var_ = varAxis0(src_demean);
124 float val = cv::sum(var_).val[0];
125 cv::Mat res;
126 cv::multiply(d,S,res);
127 float scale = 1.0/val*cv::sum(res).val[0];
128 T.rowRange(0, dim).colRange(0, dim) = - T.rowRange(0, dim).colRange(0, dim).t();
129 cv::Mat temp1 = T.rowRange(0, dim).colRange(0, dim); // T[:dim, :dim]
130 cv::Mat temp2 = src_mean.t(); //src_mean.T
131 cv::Mat temp3 = temp1*temp2; // np.dot(T[:dim, :dim], src_mean.T)
132 cv::Mat temp4 = scale*temp3;
133 T.rowRange(0, dim).colRange(dim, dim+1)= -(temp4 - dst_mean.t());
134 T.rowRange(0, dim).colRange(0, dim) *= scale;
135 return T;
136 }
137 }
```

使用说明：

```
1 float default[5][2] = {
2     {30.2946f+8.0f, 51.6963f},
3     {65.5318f+8.0f, 51.5014f},
4     {48.0252f+8.0f, 71.7366f},
5     {33.5493f+8.0f, 92.3655f},
6     {62.7299f+8.0f, 92.2041f}
7 }; // +8.0f for 112*112
8 float detect[5][2] = {
9     {30.2946f, 51.6963f},
10    {65.5318f, 51.5014f},
11    {48.0252f, 71.7366f},
12    {33.5493f, 92.3655f},
13    {62.7299f, 92.2041f}
14 }; //
15 cv::Mat src(5,2,CV_32FC1, default);
16 memcpy(src.data, default, 2 * 5 * sizeof(float));
17 cv::Mat dst(5,2,CV_32FC1, detect);
18 memcpy(dst.data, detect, 2 * 5 * sizeof(float));
19
20 cv::Mat M = FacePreprocess::similarTransform(dst, src); // skimage.transform.SimilarityTransform
21 cv::Mat warpImg;
22 //cv::warpAffine(input, warpImg, M, cv::Size(112, 112));
23 cv::warpPerspective(inpu, croppedMat, M, cv::Size(112, 112));
```

周排名	总排名	访问	等级
109	73	101	183
粉丝	获赞	评论	收藏
 FelixFuu	关注	 4  2  6 	专栏目录

FelixFuu

码龄8年 暂无认证

28

原创

1829

积分



原文链接：<https://blog.csdn.net/u011956147/article/details/90294249>

私信

关注

搜博文文章

热门文章

Faster RCNN代码理解（Python）

26729

TP、TN、FP、FN、Recall、Miss Rate、MCC、F1 Score 等指标计算

22528

openpose训练代码（一）

16726

五大常用算法：分治、动态规划、贪心、回溯和分支界定

11263

SSD：Single Shot MultiBox Detector（三）

10116

最新评论

skimage-SimilarityTransform的C++实现紫橙子: 可以自己写一个C++的动态链接库在C中调用

skimage-SimilarityTransform的C++实现qq_21336847: 博主你好，我想问一下可以在c代码中调用opencv吗？c可以调用c++...

openpose训练代码（一）new__hope: 您好，请问您实现了使用openpose 训练动物在台模型吗？

误用rm -rf *文件修复及修改rm指令为mvTisfy: 楼主绝对是具备广阔胸怀和完整知识体系的人

openpose训练代码（一）FelixFuu 回复 SyliaJason: 可以的，你 有数据就可以了，按照顺序准备好点位就行了...

您愿意向朋友推荐“博客详情页”吗？

😞

😐

😐

😄

😄

强烈不推荐

不推荐

一般般

推荐

强烈推荐

最新文章

相关开源协议

模型压缩

Merge k Sorted Lists

2020年 1篇

2019年 2篇

2018年 16篇

2017年 10篇

2016年 5篇

图像相似度比较

08-22

在vc中用c++实现图片相似度比较_方法及源码

用opencv函数替代skimage函数生成相似矩阵

yichxi的博客

240

人脸识别之前通常需要做校准工作，使得标准人脸和待识别人脸均位于“统一规范化”的条件下做对比；这里校准工作用的是相似变换SimilarityTransform【...

评论 2

您还未登录，请先 登录 后发表或查看评论

人脸图像预处理:仿射变换(对齐),相似变换(截取)_Jo厨的...

2-7

tfrom=SimilarityTransform() tfrom.estimate(dst,src) M=tfrom.params[0:2,:] align_image=cv2.warpAffine(image,M,(112,112),borderValue=0.0) 112*112*3 ...

skimage库的transform.SimilarityTransform()用法

小C的博客

6655

【时间】2019.09.04 【题目】skimage库的transform.SimilarityTransform()用法 一、SimilarityTransform（图像变换的一种：相似变换）相似变换：等距...

skimage transform SimilarityTransform 使用记录

最新发布

weixin_41977938的博客

56

SimilarityTransform 相似变换 源码及中文注释: class SimilarityTransform(EuclideanTransform): ""2d的相似变换 具有以下形式 X = a0 * x - b0 * y + a1 = ...

相似变换 SimilarityTransform 以及skimage源码

weixin_34138056的博客

1239

http://scikit-image.org/docs/dev/api/skimage.transform.html#skimage.transform.SimilarityTransform https://github.com/scikit-image/scikit-image/blob/m...

skimage(scikit-image)库学习03

C++

516

1. 图像数据类型及颜色空间转换 1.1 图像数据类型 skimage中，一张图像就是一个简单的numpy数组，数组中数据类型有很多种，同时彼此之间是可以互...

python cv::getPerspectiveTransform以及skimage的SimilarityTransform的问题总结

ANONYMOUSLYCN的专栏

1398

python cv::getPerspectiveTransform的问题总结 在python里面可以使用如下代码demo来实现： import cv2 import numpy as np img = cv2.imread('mao.jp...

OpenCV 的AffineTransform（传说中的仿射变换）

看见别人的专栏文采飞扬忍不住自己养一个专栏

6293

为了让数据集能够有旋转不变形，希望在caffe训练处好结果，我对采集的数据集进行了一个仿射变换。 利用opencv可以比较方便的实现这个事情。 我的...

python的Image和skimage处理图片

热门推荐

野孩子的专栏

2万+

一、import Image img = Image.open(path)#打开图片 img.getpixel((height, width))#得到(height, width)处的像素值（可能是一个list，3通道）img.convert(...

人脸对齐(Face Alignment)

LeeWanzhi的博客

5215

有时候，我们希望数据集中的人脸全是正的，而不是歪着头的，并且希望所有的人脸都处于图片中固定的位置，于是我们需要对图像进行旋转，将头摆正...

OpenCV 中的仿射变换和透射变换(获取鸟瞰图)

weixin_40279184的博客

1155

1.相似变换（Similarity Transform）相似变换， 旋转+平移+尺度变换； opencv中 M=cv2.getRotationMatrix2D(img.shape[1]/2,img.shape[0]/2,30,0.5)#参...

python数字图像处理（7）：图像的形变与缩放

weixin_34025051的博客

2474

图像的形变与缩放，使用的是skimage的transform模块，函数比较多，功能齐全。 1、改变图片尺寸resize 函数格式为： skimage.transform.resize(image,...

人脸对齐 matlab,人脸对齐中的相似性变换

weixin_36305686的博客

94

人脸对齐是大多数人脸分析算法中的一个关键模块，在人脸识别、表情识别、人脸建模等领域有着广泛的应用。人脸对齐获取图像中人脸的几何结构，基...

transform.SimilarityTransform()=> 图像的相似变换=等距变换（平移+旋转变换）+均匀尺度缩放

马鹏森的博客

1万+

SimilarityTransform（图像变换的一种：相似变换）相似变换：等距变换+均匀尺度缩放，所谓等距变换就是平移+旋转变换。 变换矩阵：变换效应：角度...

批量裁剪（crop）&重置图像分辨率（resize）【Python】==> skimage

写东西仪式感极强（其实拖延症晚期晚期）的大龄小白

1487

from skimage import io,transform import numpy as np #from resizeImage import resizeImage def imgfunc(f): rgb=io.imread(f) #依次读取rgb图片 ROI=rg...

opencv仿射变换：平移，缩放和旋转

DayDayUp

1099

仿射变换原理 仿射变换（Affine Transformation或 Affine Map），又称仿射映射，是指在几何中，一个向量空间进行一次线性变换并接上一个平移，变换...

C++人脸对齐，部分

m0_37192554的博客

571

python 和 c++ https://github.com/deepinsight/insightface/issues/481 代码 https://github.com/deepinsight/insightface/blob/master/cpp-align/FacePrepro...

©2022 CSDN 皮肤主题：大白 设计师：CSDN官方博客 返回首页

关于我们 招贤纳士 商务合作 寻求报道 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 ©1999-2022北京创新乐知网络技术有限公司 版权与免责声明 版权申诉 出版物