

一個簡單方法識別毛玻璃、高斯模糊

原創 晟沚 機器學習算法工程師 昨天

作者：晟沚

前言

本文主要推荐一種簡單的方法識別帶有毛玻璃、高斯模糊等效果的圖片。

01

毛玻璃效果

毛玻璃效果的原理，即遍歷每一個像素，隨機選取這個像素周圍的某一個像素，替換當前像素。可以使用opencv實現，代碼如下：

```
1 #coding:utf-8
2 import cv2
3 import numpy as np
4 import random
5
6 img = cv2.imread('test.jpg',1)
7 shape = img.shape
8 h = shape[0]
9 w = shape[1]
10 dst = np.zeros((h,w,3),np.uint8)
11 mm = 8
12 for m in range(h-mm):
13     for n in range(w-mm):
14         index = int(random.random()*8)
15         (b,g,r) = img[m+index,n+index]
16         dst[m,n] = (b,g,r)
17 cv2.imwrite("result.jpg", dst)
```

整個過程即選擇周圍像素的範圍距離最大為8像素，運行以上代碼前後如下圖：



02

識別毛玻璃等模糊效果

接下來介紹如何使用OpenCV、Python和Laplacian算子計算圖像中的模糊量，

如果有信號處理方面的背景，首先要考慮的方法是計算圖像的快速傅里葉變換，然後檢查低頻和高頻的分佈：如果圖像只有少量的高頻，那麼圖像就會被認為是模糊的。然而，定義什麼算低數量的高頻或者什麼是高數量的高頻是相當困難的。

本文介紹的方法可以計算一個單一的浮點值來表示一個給定圖像的模糊程度，只需採取一個圖像的單一通道(大概灰度)和卷積它與以下3 x 3的內核，然後取響應的方差(即標準差的平方)。如果方差低於預先定義的閾值，則認為圖像模糊；否則，圖像不會模糊。

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

方法很簡單，如下一行代碼就可

```
1 cv2.Laplacian(image, cv2.CV_64F).var()
```

这种方法有效的原因是由于Laplacian算子本身的定义，它用于测量图像的二阶导数。拉普拉斯算子突出显示图像中包含快速梯度变化的区域，很像Sobel和Scharr算子。和这些算子一样，Laplacian也经常用于边缘检测。这里的假设是，如果一幅图像的方差较高，那么就说明图像有广泛的响应，包括类边和非类边，这是一幅正常的聚焦图像的代表。但如果方差很低，那么就会有很小的响应扩散，这表明图像中几乎没有边缘。而图像越模糊，边缘就越少。所以可以用来检测是否模糊。

显然，这里的关键是设置正确的阈值，而阈值的设置与应用到的图像集相关。如果阈值过低，你就会错误地将原本不模糊的图像标记为模糊。如果阈值过高，那么实际上模糊的图像将不会被标记为模糊。这种方法只有在非常稳定的图像集（同一类型）中应用良好。

具体实现代码如下，定义variance_of_laplacian函数。这个方法将我们想要计算焦距的图像(假设为单个通道，例如灰度图像)作为参数，先读取输入图像图像，再将其转换为灰度，然后使用OpenCV应用模糊检测，通过自己设定的阈值即可区分出模糊图片和不模糊图片。

```
1 import cv2
2 def variance_of_laplacian(image):
3     return cv2.Laplacian(image, cv2.CV_64F).var()
4
5 imagePath = "test.jpg"
6 image = cv2.imread(imagePath)
7 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
8 fm = variance_of_laplacian(gray)
9 if fm < 100:
10     print("Blurry")
```



END

一个用心的公众号



长按，识别，加关注

进群，学习，得帮助

你的关注，我们的热度

我们一定给你学习最大的帮助

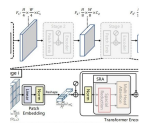


世上本没有好看，点的人多了，也就有了好看！

喜欢此内容的人还喜欢

PVT：可用于密集任务backbone的金字塔视觉transformer!

机器学习算法工程师



中国工程院院士：我这里博士生的待遇，不如能干的工人高！

人工智能与算法学习



機器學習博士自救指南（嚴肅者慎入）

圖靈人工智能

