

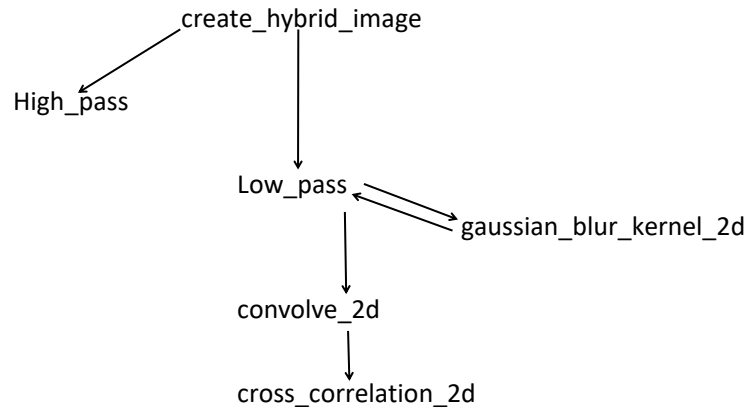
实验一：混合图像

一、实验目的

将两张图片通过高斯核进行滤波得到一幅图片的高通(left)和另一幅图片的低通部分(right)，然后按照混合比例将两张图片混合得到混合图像

二、实验内容

1. 首先明确需要实现的五个函数的调用关系：



其中 **high_pass** 通过原图减去 **low_pass** 部分得到，卷积通过将核两次逆时针反转 90° 然后与高斯核进行互相关得到。

2. 互相关函数：

- (1) 首先通过 **numpy** 中的函数将图片读成三位数组，发现 `img.shape = (361,410,3)`
- (2) 给图片加一圈半个核那么宽的白边，用于处理边界情况，防止矩阵相乘的时候发生越界。
- (3) 在原图中的坐标中循环，不断进行矩阵相乘，将得到的结果存入新开辟的数组中
- (4) 返回结果数组，得到滤波之后的图片
- (5) 注意要区分灰度图和 **RGB** 图的区别处理

3. 卷积：通过将核两次翻转，后与图片进行互相关得到

4. 卷积核：通过带入高斯核公式产生，其中按照测试例，需要将公式产生的核除以核中元素的和

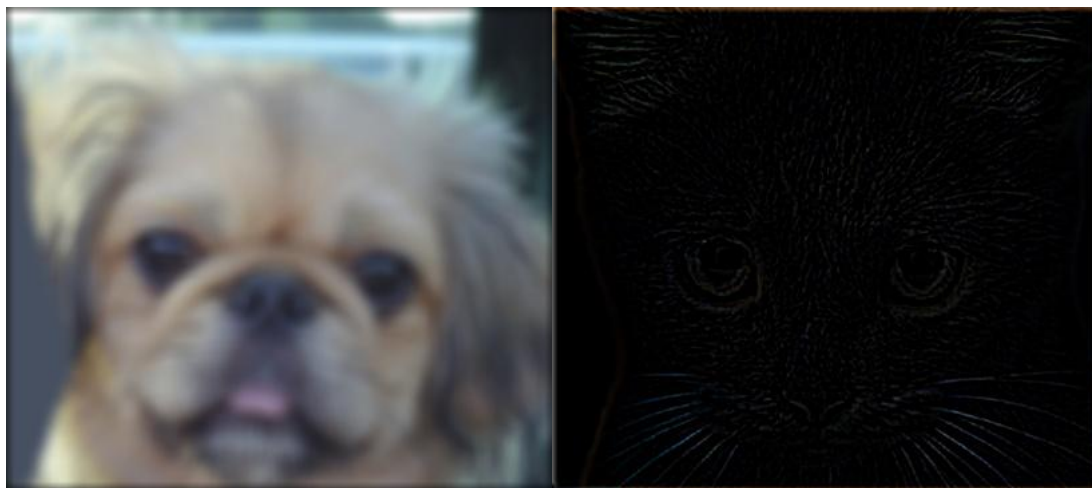
5. 低通滤波：将图片与产生的高斯核相卷积得到

6. 高通滤波：原图减去低通部分

三、实验说明

1. **left** 为低通，核大小 13，西格玛取 7.0
2. **right** 为高通，核大小为 8，西格玛取值 4.1
3. 混合比为 0.65
4. Python 3.7.4

四、实验结果



```

Unittests in test.py x
[✓] [⊘] [↓] [↓] [≡] [≡] [↑] [»] ✓ Tests passed: 19 of 19 tests – 261 ms

▼ ✓ Test Results 261 ms
  ▼ ✓ test 261 ms
    ▶ ✓ TestConvolve2D 124 ms
    ▼ ✓ TestCrossCorrelation 107 ms
      ✓ test_big_filter_grey 0 ms
      ✓ test_identity_filter_c 0 ms
      ✓ test_mean_filter_R 46 ms
      ✓ test_mean_filter_gre 0 ms
      ✓ test_mean_filter_re 15 ms
      ✓ test_rand_rect_filte 46 ms
    ▶ ✓ TestGaussianKernel2D 0 ms
    ▶ ✓ TestHighLowPass 30 ms
  
```

Testing started at 0:39 ...
 C:\Users\崔恩博\AppData\Local\Programs\Python\...
 Launching unittests with arguments python -m u...
 img : (10, 8)
 kernel : (11, 9)
 con : (10, 8)
 student
 [[8.82311922 10.19251813 12.44514836 14.1604...
 10.22966672 8.190708]
 [11.68025469 12.19769169 13.87652585 16.01613...
 11.6235984 10.11634883]

五、实验问题

1. 不熟悉 `python`，导致很多可以用 `np` 模块中的函数处理数组的部分，都用了类似 `c` 语言来写。

2. 最后一直有两个例子过不去，一个是核比图大的灰度图，一个是随机核的 `RGB` 图，一直想不通。期间将所有中间变量输出一一比对，没有发现问题，后来将我的程序得到的数组与 `np` 模块卷积出来的数组比对，发现在边角处我的函数得到的值比较低，但在每行(列)的中间处差距非常小，所以我以为是边界处理有问题，不能简单的填 `0`。

经过大量查阅资料，未果。

根据发现的规律，考虑能不能靠重复合适的像素点，将角落的值提起来，我在正常卷积的过程中，核能覆盖的最远位置处重复一次原图中的像素点，因为高斯核边缘处的权值比较低，想试试这样能不能在少影响中间部分值的情况下，将两边的值提到能通过的范围里。也不行，中间也会被提的很高

然后考虑能不能将边界像素和一维正态分布卷积，然后用原像素值减去这个结果到两变高中间低的一条像素点，将这一条围在图片四周再卷积。

最后发现是高斯核的反转用错了函数，因此而浪费了一整天，很快乐

3. 滤波之后图片会变暗，高斯核的大小如果远大于西格玛，图片会变得很蓝。

4. 最新发现是图片边缘发蓝是由于我使用 `Image` 的函数读取，使用从 `cv2` 的函数写入导致的，均改用 `cv2` 的函数后，问题解决

六、心得体会

通过本次实验，使我更加深刻的理解了高斯核的作用（按公式来说，不需要除以每个单元值的和，但 `test` 需要）。同时熟悉了卷积的操作，深刻的记住了卷积要先将核反转两次。

除了老师的素材，自己还做了一个，截图技术不好，但是确实激发了我的学习热情



原图



人比较黑所以表情变化不算很明显，但是倒数第三张可以较为清楚的感觉他笑了
相关参数

```
ratio = 0.65  
Img = create_hybrid_image(img1, img2, 7.1, 8, "low", 6.1, 8, "high", ratio)  
cv2.imwrite('hybrid.jpg', Img)
```