# [利用Python PIL、cPickle读取和保存图像数据库](http://blog.csdn.net/u012162613/article/details/43226127)

计算机视觉、机器学习任务中，经常跟图像打交道，在C++上有成熟的OpenCV可以使用，在Python中也有一个图像处理库PIL（Python Image Library），当然PIL没有OpenCV那么多功能（比如一些人脸检测的算法），不过在Python上，我们用PIL进行一些基本的图像读取与保存工作就行了，因为算法方面，Python有很多强大的算法库（机器学习库sklearn、深度学习库theano）。

本文以一个人脸图像数据库Olivetti Faces为例，展示如何使用PIL模块、cPickle模块将这个图像数据库读取并保存为pkl文件。

关于cPickle模块的使用，我在这篇文章里有讲到： [DeepLearning tutorial（2）机器学习算法在训练过程中保存参数](http://blog.csdn.net/u012162613/article/details/43169019" \t "_blank)。下文就不重复了。

**一、人脸图像库Olivetti Faces介绍**

[Olivetti Faces](http://www.cs.nyu.edu/~roweis/data/olivettifaces.gif)是纽约大学的一个比较小的人脸库，由40个人的400张图片构成，即每个人的人脸图片为10张。每张图片的灰度级为8位，每个像素的灰度大小位于0-255之间，每张图片大小为64×64。如下图，这个图片大小是1190\*942，一共有20\*20张人脸，故每张人脸大小是（1190/20）\*（942/20）即57\*47=2679：



**二、利用Python PIL、cPickle读取和保存 Olivetti Faces**

首先使用PIL.Image打开获取这张图片，为了便于数值计算，将其转化为numpy.array类型，然后每张图片摊成一个一维向量1\*2679，因为有400张，所以得到了400\*2679的numpy.array，接着使用cPickle模块，将转化得到的numpy.array保存为pkl文件。注意这是不带label的数据，我们可以人工地给它们类别0～39，每个类别有10个样本，故新建一个400\*1的label，作为每张图片对应的类别。

代码如下：

**[python]** [view plaincopy](http://blog.csdn.net/u012162613/article/details/43226127)

1. #读取人脸库olivettifaces，并存储为pkl文件
2. **import** numpy
3. **from** PIL **import** Image
4. **import** cPickle
6. #读取原始图片并转化为numpy.ndarray，将灰度值由0～256转换到0～1
7. img = Image.open('/home/wepon/olivettifaces.gif')
8. img\_ndarray = numpy.asarray(img, dtype='float64')/256

11. #图片大小时1190\*942，一共20\*20个人脸图，故每张人脸图大小为（1190/20）\*（942/20）即57\*47=2679
12. #将全部400个样本存储为一个400\*2679的数组，每一行即代表一个人脸图，并且第0～9、10～19、20～29...行分别属于同个人脸
13. #另外，用olivettifaces\_label表示每一个样本的类别，它是400维的向量，有0～39共40类，代表40个不同的人。
14. olivettifaces=numpy.empty((400,2679))
15. **for** row **in** range(20):
16. **for** column **in** range(20):
17. olivettifaces[row\*20+column]=numpy.ndarray.flatten(img\_ndarray [row\*57:(row+1)\*57,column\*47:(column+1)\*47])
19. #建一个<span style="font-family: SimSun;">olivettifaces\_label</span>
20. olivettifaces\_label=numpy.empty(400)
21. **for** label **in** range(40):
22. olivettifaces\_label[label\*10:label\*10+10]=label
23. olivettifaces\_label=olivettifaces\_label.astype(numpy.int)

26. #保存olivettifaces以及olivettifaces\_label到olivettifaces.pkl文件
27. write\_file=open('/home/wepon/olivettifaces.pkl','wb')
28. cPickle.dump(olivettifaces,write\_file,-1)
29. cPickle.dump(olivettifaces\_label,write\_file,-1)
30. write\_file.close()

这样，在目录/home/wepon/下就会得到一个olivettifaces.pkl文件。这个文件就存储了一个400\*2679的向量和一个400\*1的向量，代表样本及样本类别。

从olivettifaces.pkl中读取显示单张图片：

如果要查看单张图片，必须先将代表图片的2679维的向量reshape，如：faces[1].reshape(57,47)。调用pylab显示图片。

**[python]** [view plaincopy](http://blog.csdn.net/u012162613/article/details/43226127)

1. **import** cPickle
2. **import** pylab
3. read\_file=open('/home/wepon/olivettifaces.pkl','rb')
4. faces=cPickle.load(read\_file)
5. read\_file.close()
6. img1=faces[1].reshape(57,47)
7. pylab.imshow(img)
8. pylab.gray()
9. pylab.show()

## 机器学习算法中如何使用olivettifaces.pkl？

在机器学习算法中，我们一般都会将样本分拆为训练样本、验证样本、测试样本，以及对应的label。该如何拆分？代码如下：

**[python]** [view plaincopy](http://blog.csdn.net/u012162613/article/details/43226127)

1. 读取olivettifaces.pkl文件，分为训练集（40\*8个样本），验证集（40\*1个样本），测试集（40\*1个样本）
2. **import** cPickle
3. read\_file=open('/home/wepon/olivettifaces.pkl','rb')
4. faces=cPickle.load(read\_file)
5. label=cPickle.load(read\_file)
6. read\_file.close()
8. train\_data=numpy.empty((320,2679))
9. train\_label=numpy.empty(320)
10. valid\_data=numpy.empty((40,2679))
11. valid\_label=numpy.empty(40)
12. test\_data=numpy.empty((40,2679))
13. test\_label=numpy.empty(40)
15. **for** i **in** range(40):
16. train\_data[i\*8:i\*8+8]=faces[i\*10:i\*10+8]
17. train\_label[i\*8:i\*8+8]=label[i\*10:i\*10+8]
18. valid\_data[i]=faces[i\*10+8]
19. valid\_label[i]=label[i\*10+8]
20. test\_data[i]=faces[i\*10+9]
21. test\_label[i]=label[i\*10+9]