# 深度学习方法与实践作业一

张培　计算机学院　18023033

## 1、安装 opencv 库

1)安装 libopencv-dev 依赖包

- sudo apt install libopencv-dev

```
zpp@zpp-linux:~$ sudo apt install libopencv-dev
[sudo] zpp 的密码：
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了：
  linux-headers-4.15.0-29 linux-headers-4.15.0-29-generic linux-image-4.15.0-29-generic
  linux-modules-4.15.0-29-generic linux-modules-extra-4.15.0-29-generic
使用'sudo apt autoremove'来卸载它(它们)。
将会同时安装下列软件：
  autotools-dev debhelper dh-strip-nondeterminism gir1.2-gtk-2.0 i965-va-driver libaacs0
  libatk1.0-dev libavcodec-dev libavcodec-ffmpeg56 libavformat-dev libavformat-ffmpeg56
  libavutil-dev libavutil-ffmpeg54 libbdplus0 libbluray1 libcairo-script-interpreter2
  libcairo2-dev libcrystalhd3 libcv-dev libcvaux-dev libdc1394-22 libdc1394-22-dev
  libfile-stripnondeterminism-perl libfontconfig1-dev libfreetype6-dev libgdk-pixbuf2.0-dev
  libglib2.0-dev libgme0 libgsm1 libgtk2.0-dev libgtkglext1 libharfbuzz-dev libharfbuzz-gobject0
  libhighgui-dev libice-dev libilmbase-dev libjasper-dev libjbig-dev libjpeg-dev
  libjpeg-turbo8-dev libjpeg8-dev liblzma-dev libmail-sendmail-perl libmodplug1 libmp3lame0
  libopencv-calib3d-dev libopencv-calib3d2.4v5 libopencv-contrib-dev libopencv-contrib2.4v5
```

2)安装 opencv-python 库

- pip install opencv-python

```
zpp@zpp-linux:~$  pip install opencv-python
Collecting opencv-python
  Downloading https://files.pythonhosted.org/packages/7b/d2/a2dbf83d4553ca6b3701d91d75e42fe50aea97acdc00652
dca515749fb5d/opencv_python-4.1.0.25-cp36-cp36m-manylinux1_x86_64.whl (26.6MB)
    100% |████████████████████████████████| 26.6MB 13kB/s
Requirement already satisfied: numpy>=1.11.3 in ./anaconda3/lib/python3.6/site-packages (from opencv-python
)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.1.0.25
You are using pip version 9.0.1, however version 19.0.3 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

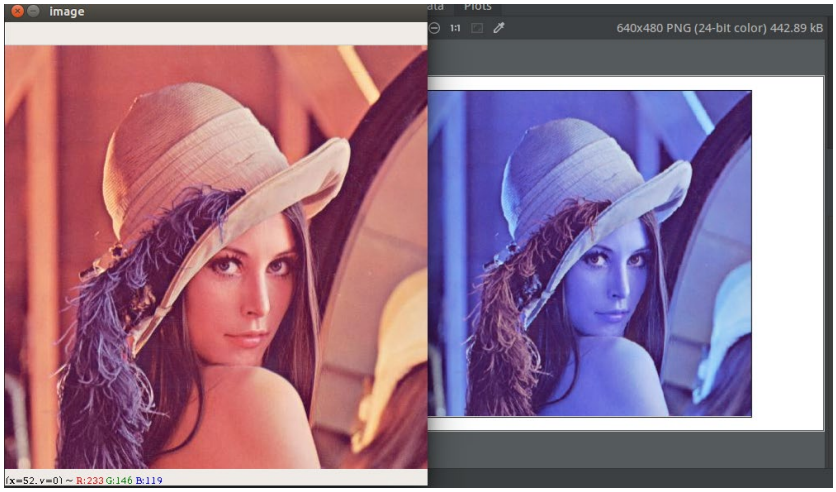
3)查看是否安装成功

- cv2.__version__

```
zpp@zpp-linux:~$ python
Python 3.6.4 |Anaconda, Inc.| (default, Jan 16 2018, 18:10:19)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.1.0'
>>>
```

## 2、测试

OpenCV 中图片通道按照 BGR 方式排列，而 Matpoltlib 中图片按照 RGB 方式排序。以下是同一张图片分别使用 cv2.imshow 显示和使用 matplotlib 显示




为了结果显示的更加明显，选择另外一张图片进行实验。

## 3、功能实现

(1) 将图片的三个通道顺序进行改变，由 RGB 变为 BRG，并用有关函数显示图片。

```
# 方法一：使用 cv2.split 函数获取通道
b, g, r = cv2.split(img)
img_BRG_bysplit = cv2.merge([b, r, g])
# 方法二：使用数组索引获取通道
b1 = img[:, :, 0]  # blue
g1 = img[:, :, 1]  # green
r1 = img[:, :, 2]  # red
img_BRG_bynp = cv2.merge([b1, r1, g1])
```

(2) 利用 Numpy 给改变通道顺序的图片中指定位置打上红框，其中红框左上角和右下角坐标定义方式为：学号为 18023033，则左上角坐标为(18,02),右下角坐标为(18+30,02+33)。(不使用 opencv 中自带的画框工具)

```
im_frame = img_BRG_bysplit.copy()
im_frame[18:49, 2] = (0, 0, 255)
im_frame[18:49, 35] = (0, 0, 255)
im_frame[18, 2:36] = (0, 0, 255)
im_frame[48, 2:36] = (0, 0, 255)
```

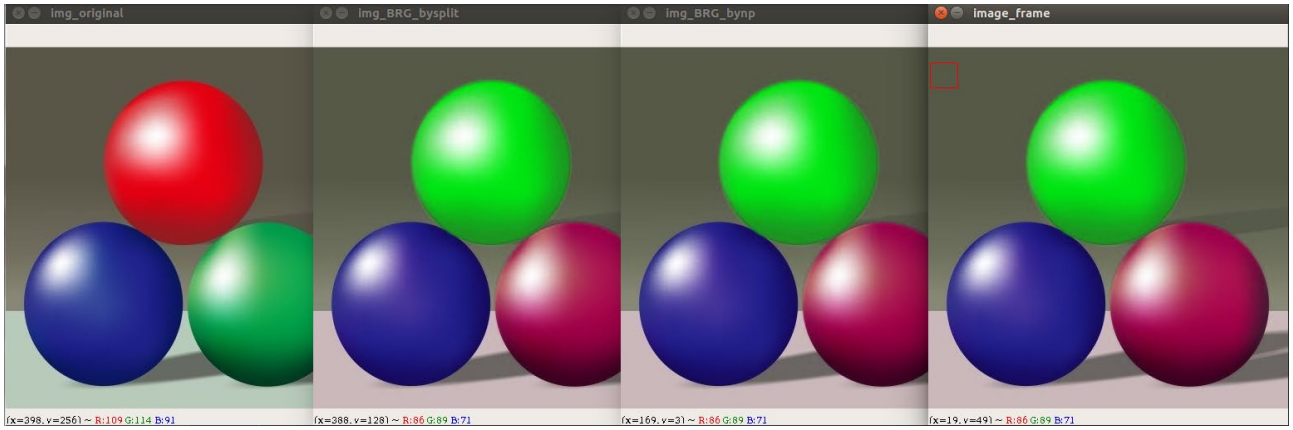
(3) 利用 cv2.imwrite()函数保存加上红框的图片。

```
# 保存加红框的照片
cv2.imwrite('im_frame.jpg', im_frame)
cv2.imshow('img_original', img)
cv2.imshow('img_BRG_bysplit', img_BRG_bysplit)
cv2.imshow('img_BRG_bynp', img_BRG_bynp)
cv2.imshow('image_frame', im_frame)
k = cv2.waitKey(0)
if k == 27:
    cv2.destroyAllWindows()
```

## 4、结果显示

第一幅为原图；第二幅和第三幅为用两种不同方式获取 BRG 的图像并用 cv2.imshow()输出；图四是使用学号作为坐标画框的结果。

由于我们将图像的通道变为 BRG 的顺序，而 cv2.imshow()读取图像的顺序是 BGR，所以预期得到的应该是蓝色保持不变，红色变成绿色，绿色变成红色。输出结果验证正确。




## 5、总结

1) cv2.imshow()和 plt.imshow()读取图像有差异，前者按 BGR 方式，后者按 RGB 方式读取。

2) opencv 提供 cvtColor()函数来实现图像灰度图、二值图、HSV、HSI 等之间的转换。

# 深度学习方法与实践作业一 进阶

张培　计算机学院　18023033

**实验目标：**

假设有函数 y = cos(ax + b),其中 a 为学号前两位，b 为学号最后两位。则我的函数为 y = cos(18x+33)。首先从此函数中以相同步长（点与点之间在 x 轴上距离相同），在 0<(ax+b)<2pi 范围内，采样出 2000 个点，然后利用采样的 2000 个点作为特征点进三次函数拟合。请提交拟合的三次函数以及对应的图样（包括采样点及函数曲线）。

## 1、功能实现

（1） 函数 y = cos(18x+33)

```
# 函数
Z = np.linspace(0, 2*np.pi, 2000, endpoint=True)
x = (Z-33)/18
C = np.cos(Z)
```

（2）在函数 y = cos(18x+33)上随机采样 2000 个点(datax,datay)。

```
'''----------------------采样 2000 个点----------------------'''
a = random.uniform(0, 2*np.pi)
zero_to_a = a
a_to_2pi = 2*np.pi - a
sample = 2000
sample_left = math.ceil(zero_to_a / (2*np.pi) * sample)
sample_right = sample - sample_left
data1 = np.linspace(0, a, sample_left, endpoint=False)
data2 = np.linspace(a, 2*np.pi, sample_right, endpoint=True)
dataz = np.concatenate((data1,data2),axis=0)
datay = np.cos(dataz)
datax = (dataz-33)/18
```

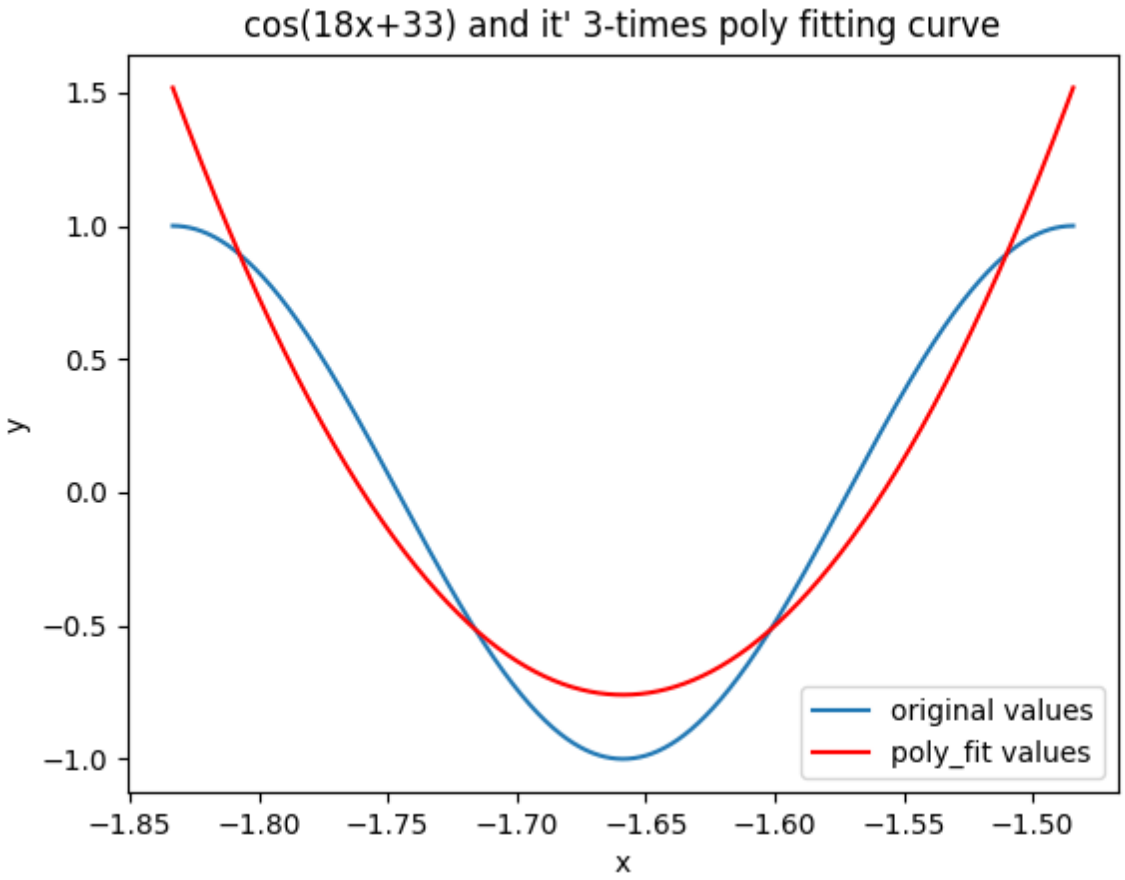
（3）使用 3 次多项式拟合随机采样的点，得到 yvals。

```
'''------------------------拟合------------------------'''
y_fit = np.polyfit(datax, datay, 3)#用 3 次多项式拟合
polynomial = np.poly1d(y_fit)#多项式表达式
yvals = np.polyval(y_fit, datax)
```

（4）绘制原始图像和多项式拟合图像

```
'''------------------------绘图------------------------'''
plot1 = plt.plot(datax, datay, '-', label='original values')
plot2 = plt.plot(datax, yvals, 'r', label='poly_fit values')
plt.xlabel('x')
plt.ylabel('y')
plt.legend(loc=4)#指定 legend 的位置
plt.title('cos(18x+33) and it\' 3-times poly fitting curve')
plt.savefig('poly_fitting.png')
plt.show()
```

2、实验结果


cos(18x+33) and it' 3-times poly fitting curve
1.5
1.0
0.5
0.0
−0.5
−1.0
y
−1.85
−1.80
−1.75
−1.70
−1.65
−1.60
−1.55
−1.50
x
original values
poly_fit values