

# 实验：图像二值化

## 实验概要

### 二值图像

到目前为止，我们已经处理了单通道（灰度图像）和三个通道（彩色图像）的图像。我们还提到，图像中的像素值表示为 8 位无符号整数，因此它们的范围是 0 到 255。其实，图像还可以用浮点值表示，也可以用较少的位数表示。例如，使用 6 位无符号整数的图像的像素值范围为 0 到 63 (2 的 6 次方减 1)。

通常，我们只使用两种范围：0 到 255 的 8 位无符号整数和图像只有 0 和 1。第二类图像只使用两个像素值，这就是为什么它们被称为二值图像。二值图像只需要一个比特来表示一个像素值。这些图像通常用于选择或删除图像的特定区域的蒙版。这些图像通常使用按位运算。您能想到在现实生活中看到过二值图像的地方吗？

您可以在二维码中找到这种黑白图像。同时，二值图像被广泛用于文档分析，甚至用于工业机器视觉任务。这是一个二值图像示例：



现在，让我们看看如何将图像转换为二值图像。此技术属于阈值处理类，阈值处理是指将彩色图像转换为二进制图像的过程。可以使用多种阈值技术，但是在这里，由于我们正在处理二值图像，因此我们仅关注一种非常简单的阈值技术：图像二值化。

图像二值化背后的概念非常简单。您选择一个阈值，然后将低于和等于阈值的所有像素值都替换为 0，而将超过阈值的所有像素值都替换为指定值（通常为 1 或 255）。这样，您最终得到的图像只有两个唯一的像素值，这就是二值图像。我们可以使用以下代码将图像转换为二值图像：

```
# 设置阈值与最大值
thresh = 125
maxValue = 255

# 图像二值化
th, dst = cv2.threshold(img, thresh, maxValue, cv2.THRESH_BINARY)
```

在前面的代码中，我们首先将阈值指定为 125，然后指定最大值——该值将替换阈值以上的所有像素值。最后，我们使用 OpenCV 的 `cv2.threshold` 函数执行图像二值化处理。此函数包括以下参数：

- **img**：进行阈值处理的灰度图像。
- **thresh**：阈值。
- **maxValue**：最大值，它将替换阈值以上的所有像素值。
- **th, dst**：阈值标志。由于我们正在执行图像二值化设置，因此我们将使用 `cv2.THRESH_BINARY`。

下面，通过实验实现我们已经了解的关于图像二值化的知识。

## 实验目标

在本实验中，我们将使用图像二值化将彩色图像转换为二值图像。我们将处理以下斑马图像：



### 1. 导入依赖库

In [1]:

```
# 导入模块
import cv2                      # 导入OpenCV
import numpy as np              # 导入NumPy
import matplotlib.pyplot as plt # 导入matplotlib

# 魔法指令，使图像直接在Notebook中显示
%matplotlib inline
```

### 2. 加载图像

读取斑马的图像并将其转换为灰度。这是必要的，因为我们知道阈值化需要我们提供灰度图像作为参数。

您也可以上传自己的图像，需要注意的是确保加载图像路径有效，95% 以上的程序报错除了缺少安装依赖库以外，大部分就跟数据路径不正确有关。这里使用 `cv2.imread()` 加载图像的路径，往往使用的都是相对路径，应该确保指定了正确的图片文件所在的路径。

In [2]:

```
# 设置输入输出路径
import os
base_path = os.environ.get("BASE_PATH", '../data/')
data_path = os.path.join(base_path + "lab2/")
result_path = "result/"
os.makedirs(result_path, exist_ok=True)

img = cv2.imread("./data/zebra.jpg") # 读取图像文件
```

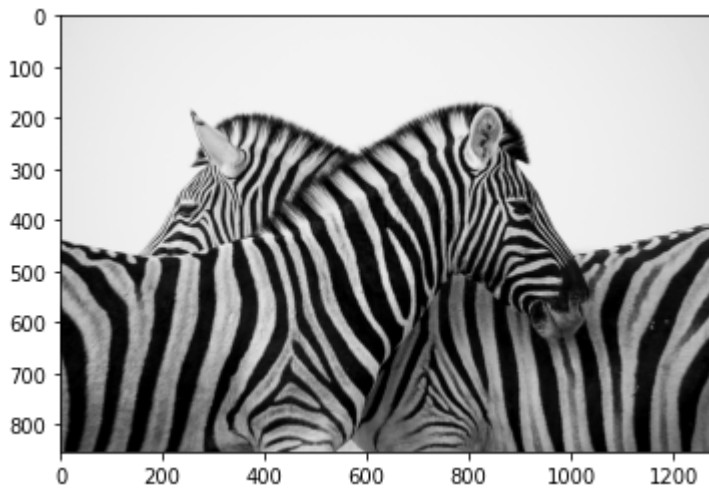
In [3]:

```
# 将图像文件转换为灰度图片，共后续图像二值化操作
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

使用 Matplotlib 显示图像，输出信息如下。X 轴和 Y 轴分别为图像的宽度和高度：

In [4]:

```
plt.imshow(img, cmap='gray') # 使用灰色“喷涂”图像输出显示
plt.show()                  # 显示图像
```



### 3. 设置阈值

使用 `cv2.thresholding` 函数并将阈值设置为 150，同时，您可以尝试使用阈值来获得不同的结果。

In [5]:

```
# 设置阈 (yu) 值和最大值
# 设置阈值，小于等于阈值的像素将被替换为0
thresh = 150
# 设置最大值，大于阈值的像素将被替换为这里设置的最大值
maxValue = 255
```

### 4. 图像二值化

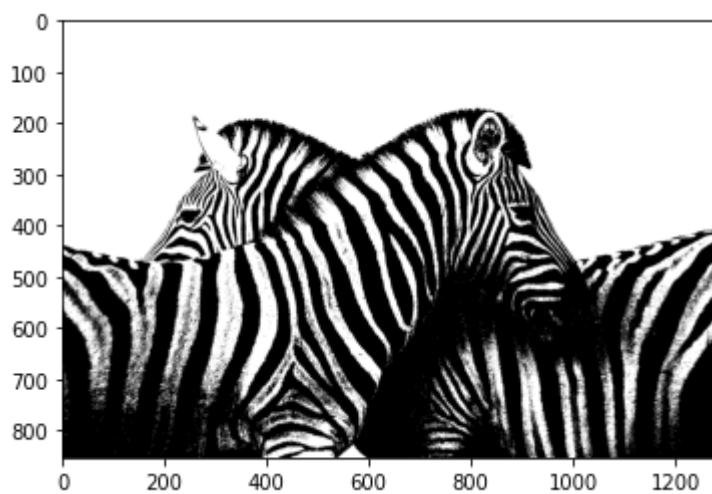
In [6]:

```
# 执行图像二值化
th, dst = cv2.threshold(img, thresh, maxValue, cv2.THRESH_BINARY)
```

使用 Matplotlib 显示图像，输出信息如下。X 轴和 Y 轴分别为图像的宽度和高度:

In [7]:

```
# 使用灰色“喷涂”图像输出显示  
plt.imshow(dst, cmap='gray')  
# 显示图像  
plt.show()
```



## 实验小结

在本实验中，我们看到了如何使用阈值获得二进制图像。