1. 课程：第一节．人脸探测

1. 理解机器学习基本概念,监督学习和非监督学习的区别．
2. 数据集,训练样本．（了解）
3. 了解人脸识别的实质(监督学习中的分类算法)．
4. Hog 的计算和 Python 实现．（了解）
5. 学习使用 skimage，matplotlib，numpy 等 Python 库．

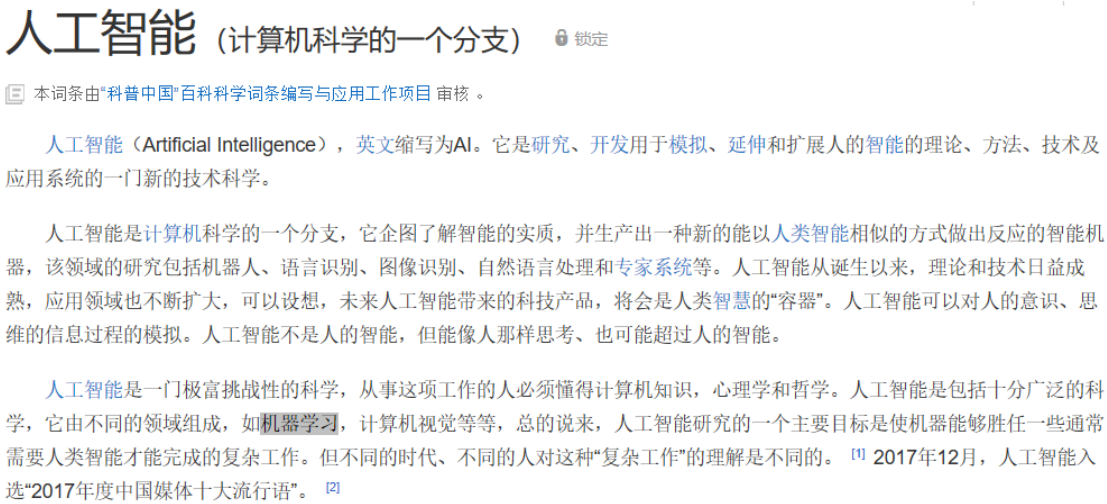
2. 基本概念

2.1. 机器学习基本概念

机器学习:

提起机器学习,人们往往想到的是机器人．其实,数十年前,在 20 世纪 60 年代,我们就有了机器学习的应用----**光学字符识别**(OCR); 90 年代，**垃圾邮件过滤**，作为机器学习的第一个应用，具有里程碑意义．

机器学习，是指 ”使计算机模拟或实现人类的**学习**行为，以从数据中获取**新的知识**或技能，重新组织已有的知识结构使之不断**改善自身**的性能” 这样一种过程．机器学习是数据科学的核心，是现代人工智能的本质.机器学习是人工智能之核心，是使计算机的行为更接近人类的根本途径; 机器学习的应用遍及人工智能的**各个领域**，它主要使用归纳、综合方法．类比，小孩学习任何一个新概念．


人工智能（计算机科学的一个分支） 锁定

本词条由“科普中国”百科科学词条编写与应用工作项目 审核 。

人工智能（Artificial Intelligence），英文缩写为AI。它是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。

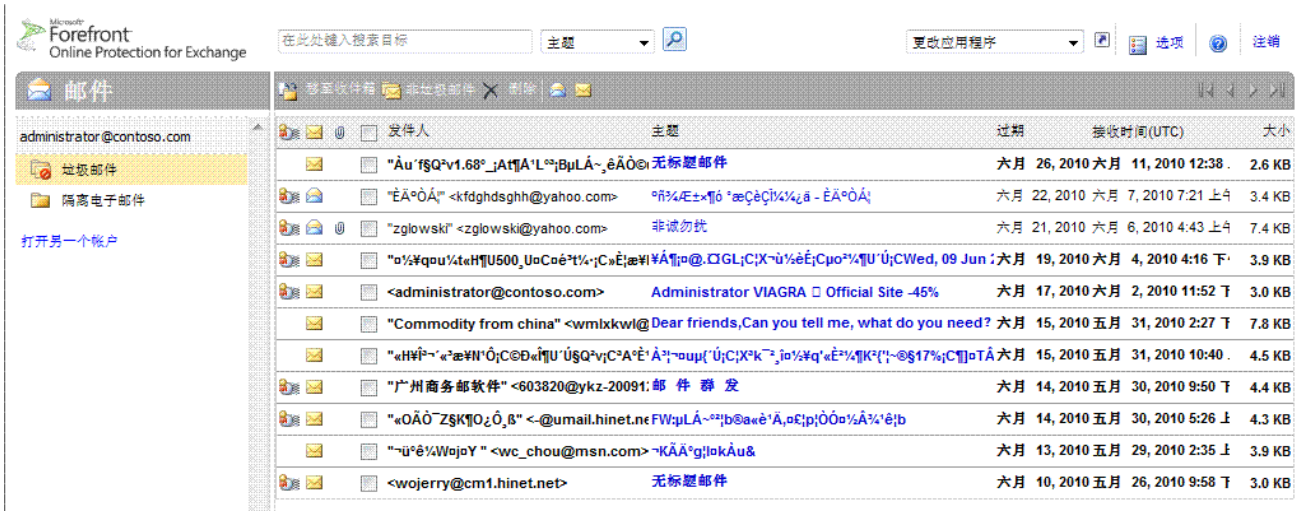
人工智能是计算机科学的一个分支，它企图了解智能的实质，并生产出一种新的能以人类智能相似的方式做出反应的智能机器，该领域的研究包括机器人、语言识别、图像识别、自然语言处理和专家系统等。人工智能从诞生以来，理论和技术日益成熟，应用领域也不断扩大，可以设想，未来人工智能带来的科技产品，将会是人类智慧的“容器”。人工智能可以对人的意识、思维的信息过程的模拟。人工智能不是人的智能，但能像人那样思考、也可能超过人的智能。

人工智能是一门极富挑战性的科学，从事这项工作的人必须懂得计算机知识，心理学和哲学。人工智能是包括十分广泛的科学，它由不同的领域组成，如机器学习，计算机视觉等等，总的说来，人工智能研究的一个主要目标是使机器能够胜任一些通常需要人类智能才能完成的复杂工作。但不同的时代、不同的人对这种“复杂工作”的理解是不同的。 [1] 2017年12月，人工智能入选“2017年度中国媒体十大流行语”。 [2]

1959 年,人工智能先驱 Arthur Samuel 给出了机器学习最早的定义：机器学习是”不用具体地编程,而让计算机具有学习能力的一个研究领域.”

从具体例子中到数据集概念:

垃圾邮件过滤器(spam filter)，就是一个能根据已知垃圾邮件和普通邮件的例子来标记垃圾邮件的机器学习程序．这些已知垃圾邮件和普通邮件的例子就称为**训练集**(training set)．每封邮件就是一个**样本**(或称记录，sample)．机器学习的任务就是对一封**新邮件**标记出它是否为垃圾邮件,也就是对邮件进行分类．新邮件组成的集合,就称为**测试集**(testing set)．机器学习做出的分类是否准确,可以用准确率(accuracy)来测量．这是一个分类问题．机器学习的一个主要任务是**分类**．**目标变量**是机器学习算法的预测结果（标签）．



训练集:用于训练机器学习算法的**数据样本集合**．

为了测试机器学习算法的效果,我们通常用到**两套**独立的样本集： 训练集和**测试集**． 机器学习先使用训练集作为算法的输入,训练完成之后输入测试集．

数据集的表示：

数据集：$D:\{(X^{(1)},y^{(1)}),\ldots,(X^{(N)},y^{(N)})\}$

X，用来表示特征． X=(“有无乱码”,”有无标题”,”有无敏感词汇”,) 第一封邮件： X=(1,0,1)

y，用来表示标签． y= “垃圾邮件”; y = 0．

机器学习的分类：

监督学习：机器学习两大类之一,训练样本中同时包含有**特征**和**标签**信息的机器学习算法．例如,邮件过滤器中,作为训练集的邮件自带”正常邮件”或”垃圾邮件”标签．

无监督学习： 另一种机器学习算法,其样本只含有特征,不包含标签信息． 最典型的算法是聚类算法(k 均值算法,期望最大化算法(EM)等)和降维算法(如主成分分析算法（PCA),线性判别式分析(LDA),小波分析,拉普拉斯特征分析,...）．

监督学习的分类：

我们这里先来了解机器学习中的监督学习．监督学习包含**分类**和**回归**两种算法．这类算法必须知道预测什么(这类算法必须知道目标变量的分类信息)才可进行．

监督学习的流程：

1.训练部分：(邮件，小孩学习新概念)

获取数据 → 特征提取 → 监督学习 → 评估

2．预测部分：

模型 → 预测

2.2. 分类算法和回归算法

分类算法的目标是从(已有的)数个可能的分类标签中预测一个样本所属的分类标签． 例如，前面介绍的标记垃圾邮件的算法，就是一种分类算法．又如，识别一个网站所用的语言．识别结果可以是英语，汉语，日语等语言中的一种．语言之间**没有连续性**．｛中文，日文，韩文，英文，德文，西班牙文，...｝ ， 好比一个一个的筐，一般物体不能同时放在两个筐中．)

回归算法的目标是预测一个**连续的数**(浮点型)． 例如，由一个人的年龄，教育程度，住址来预测其收入．由农作物的品种，种植的位置，降雨量等特征来预测该农作物的产量．这里的收入和农作物的产量的取值都可以是一定范围内的**任何值**．给计算机输入一套房的面积，位置，朝向，房间数目，计算机就可以自动给你算出它的价格（回归算法)；输入一个人的学历，住址，朋友数目，去过的地方，计算机也可以自动给你算出他／她的年收入(回归算法)； ．．．．．．

2.3 人脸识别的步骤概述

例 1.输入一种植物的花萼数目，花瓣数目，花瓣长宽比，雄蕊数目，雌蕊数目，颜色，香味等，计算机就可以告诉我们这种植物的名称(分类算法)． 这个识别花的问题都可以通过选择一个机器学习算法来解决：给它数据集，训练出模型，然后等待输出结果． 一种花就好比一张脸．识别一张人脸，就好比对识别一种花；对花儿进行分类，就好比人脸识别．

百合花　　梨花　　油菜花　　茉莉花　　海棠花

人脸识别(face recognition)也是可以通过机器学习算法来做的，它是由一系列机器学习算法构成的．下图显示同一个人的三张照片．

人脸识别的步骤如下．

1．在图上找出所有的人脸．

2． 对同一张脸，即使它旋转或者扭曲，让计算机要能知道这是同样一张脸．

3． 能找出这张脸独特的特征，以用来将这张脸与其他人的脸区分开（特征可以是：眼睛大小，鼻子长度等）．

4．将该脸的独特特征与所有已经标识的人脸进行对照，以确定该人脸对应的人的姓名．

注意：我们的大脑非常善于人脸识别！能瞬间完成上面所有的步骤！

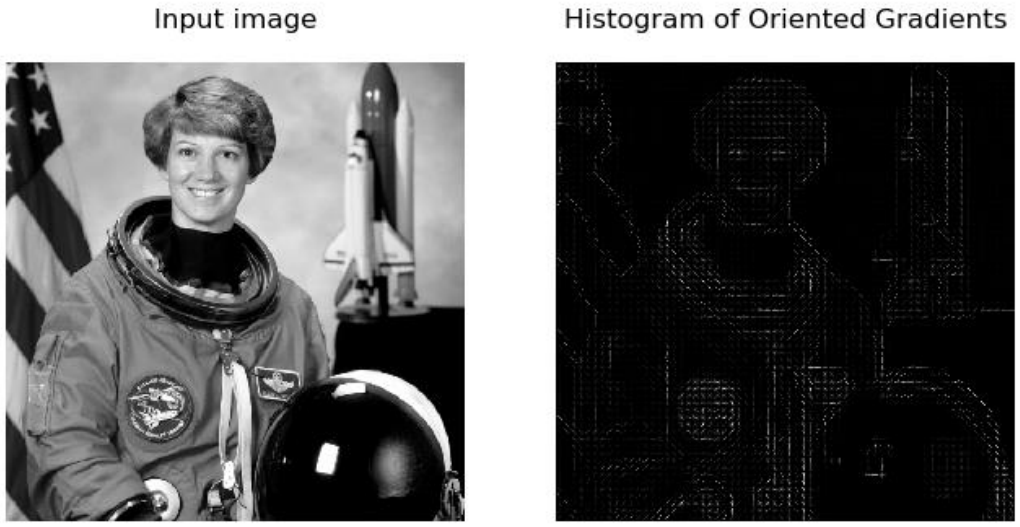
我们将分四个步骤来处理人脸识别． 对每一步， 我们将学习一个不同的机器学习算法．我们将会学习每一个算法背后的基本思想.以后你可以用Python去建立自己的人脸识别系统.可能用到的库有OpenFace和dlib， scikit-image,os,matplotlib,scikit-learn 等.

3. 人脸探测

第一步是人脸探测． 区分人脸之前，首先要找出人脸在照片中的位置！ 近几年来的相机都有人脸探测的功能，它能找出人脸，以便可以对每张人脸对焦，从而得出更清晰的照片.我们这里用它来做识别人脸，而不是用来得到人脸的更清晰的照片.

本世纪初,Paul Viola 和 Michael Jones 发明了能应用在一般照相机上的快速探测人脸的方法，人脸探测在 2000 到 2010 年间就成为主流技术. 现在，人们有了更好的方法用于快速探测人脸.

我们将用的方法----**方向梯度直方图**(Histogram of Oriented Gradients, HOG, 2005).

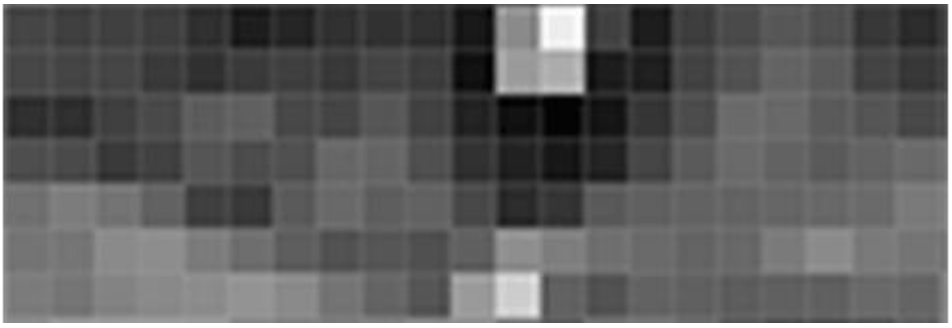


首先，因为色彩的信息对我们做人脸识别时不需要,所以我们先将照片转成黑白照片.

然后，我们依次关注照片中的每一个像素点. 先看一个例子.

如果我们直接分析像素， 同一个人的非常暗的照片和非常亮的照片将具有**完全不同**的像素值(如图). 一个很好的办法是: **考虑像素亮度的变化**! 当考虑亮度改变的方向时，暗照片和两照片将有几乎一样的表示! 所以我们不直接处理像素值信息，而是处理像素明暗的变化信息.

对每个像素点， 我们想看看那些直接包围着它的像素点. 我们的目标：计算出当前像素相对于其周围的像素有多暗. 然后我们可以**画出**像素由明到暗**变化最快的方向**:



<img src="https://github.com/hg08/tututu/blob/master/oriented_gradients.gif?raw=true">

对照片中的每一个像素点重复上述操作，最终每一个像素都被一个箭头取代了．这些箭头称为**梯度**，它们显示整张照片由明到暗变化最快的方向．

[Dalal, N and Triggs, B, *Histograms of Oriented Gradients for Human Detection,* IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2005 San Diego, CA, USA, https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf, DOI:10.1109/CVPR.2005.177]

在计算 HOG 时，如果将**方向数目**这个参数设置为 9,那么圆周(360°)就被分成九个区间．它们分别是:
[0, 40],[40, 80],[80,120],[120, 160],[160, 200],[200, 240],[240, 280],[280, 320],[320,360].
单位为度(°).

进一步简化：我们用梯度来看一些基本模式，而不是所有的细节！为了达到这个目的，我们将照片分成小方格，每个方格的大小为16×16像素(或其他分割方法,如8×8等)．同时设置主要方向的总数目(可取4,8等)．在每一个小方格中,我们将计算在每个主要方向有多少个梯度点（有多少个指向上方，有多少个指向下方，有多少个指向右上方，等等)．然后我们以最强的箭头方向去取代该**小方格**．这样，我们将原图片转化成了非常简单的表示．它只以简单的方式描述人脸的基本结构.

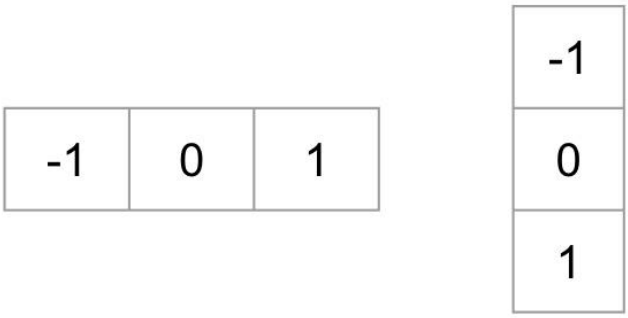
怎么从 HOG 图中找到人脸? 为了在这张 HOG 图片中找到人脸，我们只需找到我们的图片中最像**已知 HOG 模式**的那部分．已知 HOG 模式由大量其他照片训练并提取出来，如图：

<src="https://github.com/hg08/tututu/blob/master/HOG_face_pattern_standard.png?raw=true">

具体组成部分：运用 Dlib 库中的 face detection algorithm 来看是否图片中有人脸存在.

3.1. 计算梯度图像

为了计算一个 HOG 描述器，我们首先需要计算水平梯度和竖直梯度;然后可求得方向梯度直方图. 这可以由如下的核过滤该图片而得:



<src='https://www.learnopencv.com/wp-content/uploads/2016/11/gradient-kernels.jpg' width='350'>

我们也可用 OpenCV 中的 Sobel 算符(with kernel size 1)来达到同样的目的.

```
# Python gradient calculation
# Read image
im = cv2.imread('E:\open_courses\eg.png')
im = np.float32(im) / 255.0

# Calculate gradient
gx = cv2.Sobel(im, cv2.CV_32F, 1, 0, ksize=1)
gy = cv2.Sobel(im, cv2.CV_32F, 0, 1, ksize=1)
```

(附: Sobel 算子的原理)

然后，我们可以用g_x, g_y来计算梯度的大小和方向：

$$g^2 = (g_x{}^2 + g_y{}^2)$$

$$\text{angle} = \arctan(\frac{g_y}{g_x})$$

如果你用 OpenCV，可用 cartToPolar 函数来做计算，如下.

```
# Python Calculate gradient magnitude and direction ( in degrees )
mag, angle = cv2.cartToPolar(gx, gy, angleInDegrees=True)
```

3.2. 计算 HOG：用 scikit-image 显示 HOG 图

python 代码示例

我们可以利用 io.imread()导入图片，使之成为 numpy.ndarray 型的数据. 然后也可对图片进行同样的操作，得到 HOG 图. 代码如下：

```
import matplotlib.pyplot as plt
from skimage import color ,io
from skimage.feature import hog
from skimage import data, exposure

image = io.imread('./li_00.jpg')
image = color.rgb2gray(image)

print("type of image is:{}".format(type(image)))
print("image is:{}".format(image))
print("no. of columns of image is:{}".format(len(image[0])))
print("no. of rows of image is:{}".format(len(image)))

array, hog_image = hog(image, orientations=9, pixels_per_cell=(8, 8),
            cells_per_block=(1, 1), visualise=True)

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(8, 4))

ax1.imshow(image, cmap=plt.cm.gray)
ax1.set_title('Input image')

# Rescale histogram for better display
hog_image_rescaled = exposure.rescale_intensity(hog_image, in_range=(0, 10))
ax2.imshow(hog_image_rescaled, cmap=plt.cm.gray)
ax2.set_title('Histogram of Oriented Gradients')
plt.show()
```

4. 课程总结

4.1. 重点

1. 监督学习和非监督学习的区别,监督学习的流程,训练集,测试集,分类和回归的区别
2. HOG 的原理
3. Python 库的使用:skimage, OpenCV
4. 了解 Dlib 库

4.2. 难点

1. HOG 的原理和计算

4.3. 扩展

1. Dalal, N. and Triggs, B., "Histograms of Oriented Gradients for Human Detection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, San Diego, CA, USA.

2. David G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.

3. Adam Geitgey, Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning