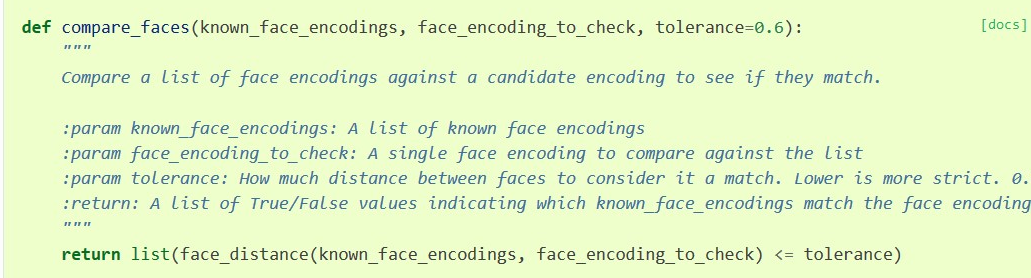
**人脸算法**

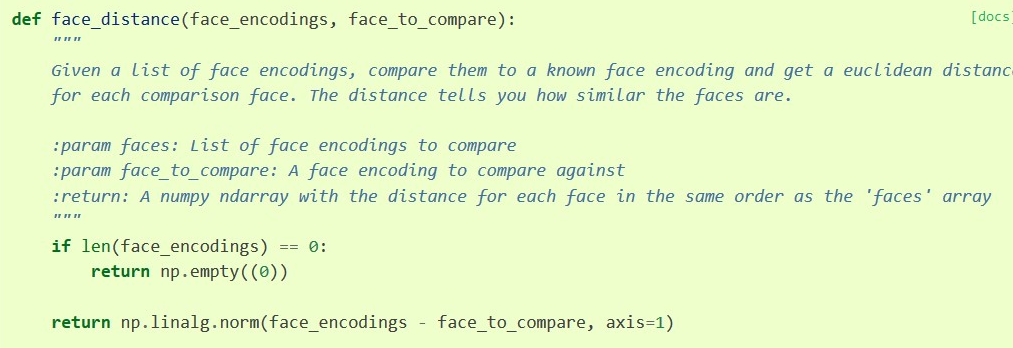
人脸算法效果决定了人脸识别的上限，也是人脸比对最关键的要素。随着深度学习技术的普及，各大公司的人脸算法效果差距也越来越小。

目前主流的人脸识别算法可以分为以下四类：

1. 基于人脸特征点的识别算法
2. 基于整幅人脸图像的识别算法
3. 基于模板的识别算法
4. 利用神经网络进行识别的算法

Face\_recognition

欧式距离比较，tolerance是可以调整参数



np.linalg.norm(求范数)



这里是1范数 ord=1：列和的最大值

Demo code

results = face\_recognition.compare\_faces(known\_faces, unknown\_face\_encoding)

print("Is the unknown face a picture of Biden? {}".format(results[0]))

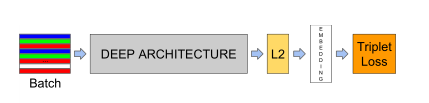
print("Is the unknown face a picture of Obama? {}".format(results[1]))

FaceNet

[FaceNet: A Unified Embedding for Face Recognition and Clustering](https://arxiv.org/abs/1503.03832)

<https://github.com/davidsandberg/facenet>

通过 CNN 将人脸映射到欧式空间的特征向量上，计算不同图片人脸特征的距离，通过**相同个体的人脸的距离，总是小于不同个体的人脸**这一先验知识训练网络。   
测试时只需要计算人脸特征，然后计算距离使用阈值即可判定两张人脸照片是否属于相同的个体。



前面就是一个传统的卷积神经网络，然后在求L2范数之前进行归一化，就建立了这个嵌入空间，最后损失函数。

输入是一个三元组 <a, p, n>

* a： anchor
* p： positive, 与 a 是同一类别的样本
* n： negative, 与 a 是不同类别的样本

公式是： *L*=*max*(*d*(*a*,*p*)−*d*(*a*,*n*)+*margin*,0)

所以最终的优化目标是拉近 a, p 的距离， 拉远 a, n 的距离

提取特征直接计算距离，比之前的使用 PCA + SVM 更加简单，训练的损失函数直接针对实际误差，end-to-end 方式训练都能提高精度

直接学习图像到欧式空间上点的映射，两张图像所对应的特征的欧氏空间上的点的距离直接对应着两个图像是否相似

ArcFace / InsightFace 论文解读

InsightFace 是 DeepInsight 实验室对其论文 ArcFace: Additive Angular Margin Loss for Deep Face Recognition 的开源实现。本文工作将 MegaFace 的精度提升到 98%，超过俄罗斯 Vocord 公司保持的 91% 的纪录。

ceter loss提出了可视化在某种损失函数下样本的分布，在此基础上sphere face，norm face，cos face的出现，都揭示了一个现象，最后一个全连接层的权重W的每一列，都代表了每一类的样本中心向量

作者认为他们提出的损失函数有四点优势

Engaging（有吸引力?）:arcFaca 直接优化能测量出的距离（角度距离），这种优化是通过规范化的超球体的确切的联系实现的。

Effective：因为state of the art ，而且在10个人脸识别benchmarks上测试效果都很好。

Easy：容易实现

Efficient：仅仅需要少量的多余计算

损失函数

基于欧式距离的损失函数

网络的输出结果为人脸的特征向量，再根据不同人脸特征向量之间的距离判断两者是否是一个人

以这种思路为基础，一些损失函数(Centre loss, Range loss, Marginal loss)额外添加惩罚来使类内距离变小,类外距离变大，以提升准确率。利用这几个损失函数的算法都在网络的最后使用了softmax，但是由于人脸分类结果的庞大（百万级），这样会是的分类层消耗大量GPU资源

contrastive loss和Triplet loss利用了结对训练的策略(pair training strategy)。contrastive loss函数包括positive pairs和negative pairs。positive pairs的损失函数倾向于小，negative pairs的损失函数倾向于大。Triplet loss损失函数最小化一个anchor和positive样例，最大化一个anchor和negative样例（如果两个样例图为同一个人，即positive；如果两个样例图为两个人，即negative）。

Angular and cosine margin based loss

出现了large margin Softmax (L- Softmax)，除了Euclidean distance上的差异，引入了angular margin。使得类内间距变小，类外间距变大。

欧式距离和angular and cosine margin based loss比较

关于马氏距离与余弦相似度的区别，个人以为这是一种长度与方向的度量所造成的不同。

马氏距离度量的是长度，它只有远近之分。而余弦相似度是度量方向的，余弦相似度为1，你不可以说这两个向量一样，只能说他们是相似的，因为他们是同方向的，就好象（3，3）与（5，5）。但是马氏距离度量的是长度，长度为0就可以认为他们是一样的。余弦相似度只在［0，1］之间，有准则，而马氏距离在［0，无穷）之间，无判别准则．余弦相似度为０（即直交）就可以说他们很不相似，就算他们之间距离小，但是他们方向完全不一致。而马氏距离就需要另找判别准则来定义怎么算大，怎么算小。于是相对应的应用也就出来，如果你所想应用的是对应方向上，则应该用余弦相似度。但是若要用来区分价值大小，高低价值用户聚类，则应该用马氏距离

[insightface](https://link.zhihu.com/?target=https%3A//github.com/deepinsight/insightface)和[facenet](https://link.zhihu.com/?target=https%3A//github.com/davidsandberg/facenet)，包括算法效果与性能，facenet使用的是较早的softmax,Python3环境，基于tensorflow实现；insightface使用的是18年出的arcface，Python2环境，基于mxnet实现。

Seetaface 人脸检测算法简析

整个算法框架呈漏斗型，即先用计算量小的特征，分类方式快速过滤大量非人类窗口，即在图像上划窗，然后用复杂特征，分类方式逐层筛选人脸。

第一层LAB特大分类模型

1. 一共有左侧脸、正脸、右侧脸三个LAB boosted 模型，即有三份不同特征、权重、阈值参数。
2. 每个LAB boosted模型中，包括150个弱分类器。可以理解为40\*40的划窗中，不考虑边界情况1600个点，选取150个作为LAB特征的计算特点，每个LAB特征为9\*9像素。3\*3cell，cell宽高是3\*3像素，计算输出为0~255；每个计算输出值对应一个累计权重值，用于分类器判断。
3. LAB的计算与opencv的稍有差异，在于lbp的9个顺序不一样、判断符号为大于等于号。
4. 弱分类器工作时，以10个为一组进行权重值累计，如果小于对应的累加权重值，则为非人脸窗口，继续下一个滑窗判断；如果15次累加结束(即150个弱分类器)，大于对应的权重值，则判断为人脸窗口。

2.第二~四层SURF MLP分类模型

1）一共有三个SURF MLP分类模型，每个模型在SURF窗口数、MLP神经网络层数、权重/偏置参数有不同，体现在模型文件中。

2）SURF特征，均是2x2的cells，但特征的宽高比有5种(1:1 1:2 2:1 2:3 3:2)。比如(0,0,16,16)为1:1、(0,0,32,16)为2:1。每个cells计算得到8维数据[dx+、|dx|+、dx-、|dx|-、dy+、|dy|+、dy-、|dy|-]，2x2cells即为2x2x8=32维数据，因此一个SURF特征为32维数据。三个模型在每个40x40滑窗中分别需计算4、8、16个SURF特征，即对应MLP的输入位数4x32=128、8x32=256、16x32=512。

3）MLP的使用激活函数均为Relu，最后一层激活函数为Sigmoid。

4）MLP输出结果为4维数据，0：score， 1：x回归， 2：y回归， 3：w/h回归。 所有MLP模型score > 0.5，且经过NMS后score累加大于默认阈值2，才判断为人脸。(可以理解为NMS前，这个位置至少要有3个窗口叠加，单个人脸窗口score一般为0.9多)

3.NMS非极大值抑制

1）nms主要根据设置iou值(重叠区域/合并区域)来判断两个区域是否需合并。

2）首先，先按score对人脸位置队列排序，高分放在最前面。并设置一个标记位mask\_merged来判断对应区域是否已被合并。

3）遍历人脸位置队列，分别与后面mask\_merged为0的区域进行判断，如果重叠比例大于设定的iou值，则累加score，并将被合并的区域标记位置1。

4）nms实际是根据iou值删除需被合并的人脸位置。