YOLOV4结构参考文章:https://blog.csdn.net/weixin\_41560402/article/details/106119774  
开源代码及数据集网址:https://github.com/bubbliiiing/yolov4-pytorch  
  
主干特征提取网络:CSPDarkNet53  
特征金字塔:SPP PAN

激活函数:使用Mish激活函数  
  
**训练步骤:**VOC格式训练，标签格式是xml格式  
训练集VOC2007+2012训练集，一共5k多张，但其中有戴口罩的，要去掉。VOC格式的标签文件中name是0的就是没戴口罩的。  
具体步骤:1.训练前将标签文件放在VOCdevkit文件夹下的VOC2007文件夹下的Annotation中。

2.训练前将图片文件放在VOCdevkit文件夹下的VOC2007文件夹下的JPEGImages中。

3.在训练前利用voc2yolo4.py文件生成对应的txt  
voc2yolo4.py文件中训练集测试集划分比例为0.8  
运行后划分了test.txt与train.txt，按比例随机划分。只包含图片名字。

4.再运行根目录下的voc\_annotation.py，修改classes为‘0’(运行时跳过有口罩的人脸，训练还是放进去训练，只是会被当做没有目标  
此时会生成对应的2007\_train.txt，每一行对应其图片位置及其真实框的位置，还有图片类别

5.在训练前在model\_data下新建一个txt文档voc\_classes.txt，文档中输入需要分的类face，在train.py中将classes\_path指向该文件夹

6.将kmeans\_for\_anchor.py中的图片大小改为608，运行得出yolo\_anchors.txt中的数据

通俗地说：把数据集中所有的宽高聚类，聚成9类，人脸的宽高在这个范围内进行调整。

例如警察抓犯人，如果有犯人的体征，更容易实施抓捕。

Anochor Box的尺寸该怎么选择？使用K-means聚类来代替人工设计，通过对训练集的bounding box进行聚类，自动生成一组更加适合数据集的anchor，可以使网络的检测效果更好.

Anchor box参考文章：<https://blog.csdn.net/weixin_34693059/article/details/112179549>

7.train.py中的图片大小改为608×608。

train.py文件下:mosaic参数用于控制是否实现Mosaic数据增强，设置为true  
Cosine\_Ir用于控制是否使用余弦退火，设置为true。

Mosaic增强与余弦退火衰减（余弦学习率）参考文章自己找一下

val\_split设置为0.1，即从训练集中取出百分之十用来验证（在训练过程中），与测试不一样。

Batch\_size设置为4 参考文章：

https://blog.csdn.net/qq\_34886403/article/details/82558399?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522161865781916780261954464%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request\_id=161865781916780261954464&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~sobaiduend~default-1-82558399.first\_rank\_v2\_pc\_rank\_v29&utm\_term=batch-size

不会冻结特征提取层的网络进行训练（效果一般），选择整体训练，从0开始训练到100次

使用adam优化器，参考文章：

<https://blog.csdn.net/weixin_41036461/article/details/80426304?ops_request_misc=&request_id=&biz_id=102&utm_term=adam%E4%BC%98%E5%8C%96%E5%99%A8&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-8-80426304.first_rank_v2_pc_rank_v29>

运行，训练。

**预测步骤：**

**训练得出logs文件夹下面的权值文件。**

**在yolo.py文件里面，修改model\_path和classes\_path使其对应训练好的文件model\_path对应logs文件夹下面的权值文件，classes\_path是model\_path对应分的类。**

**Cuda设置为true,使用Gpu来预测。置信度阈值设置为0.5。**

**Iou：刚预测出来时，可能有很多个框叠加在目标上。Iou是指两个框重叠的比例，即交并比（NMS参考文章中提到了这个概念，不要漏掉）**

**Iou值设置为0.3**

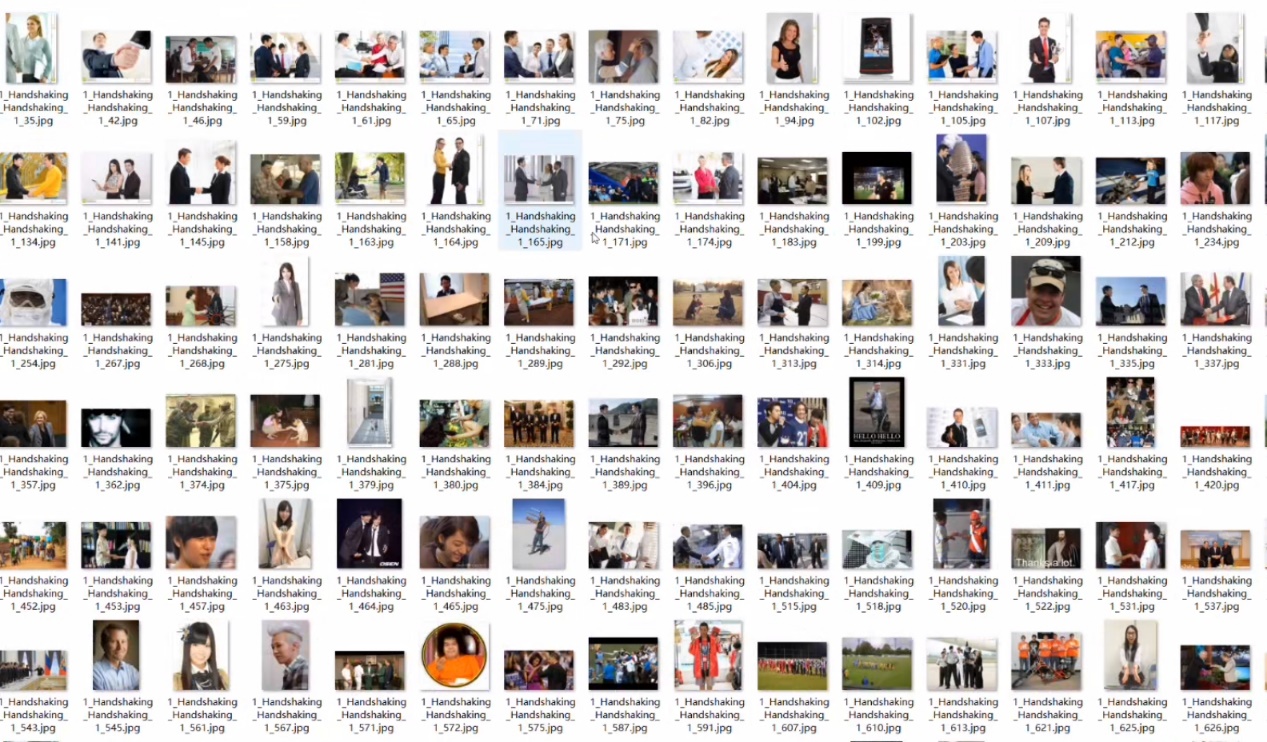
**通过NMS（非极大抑制）选择置信度最高的框，保证算法对每个对象只检测一次**

**iou参考文章：**<https://blog.csdn.net/IAMoldpan/article/details/78799857?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522161865982816780264081045%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=161865982816780264081045&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_v2~hot_rank-2-78799857.first_rank_v2_pc_rank_v29&utm_term=iou>

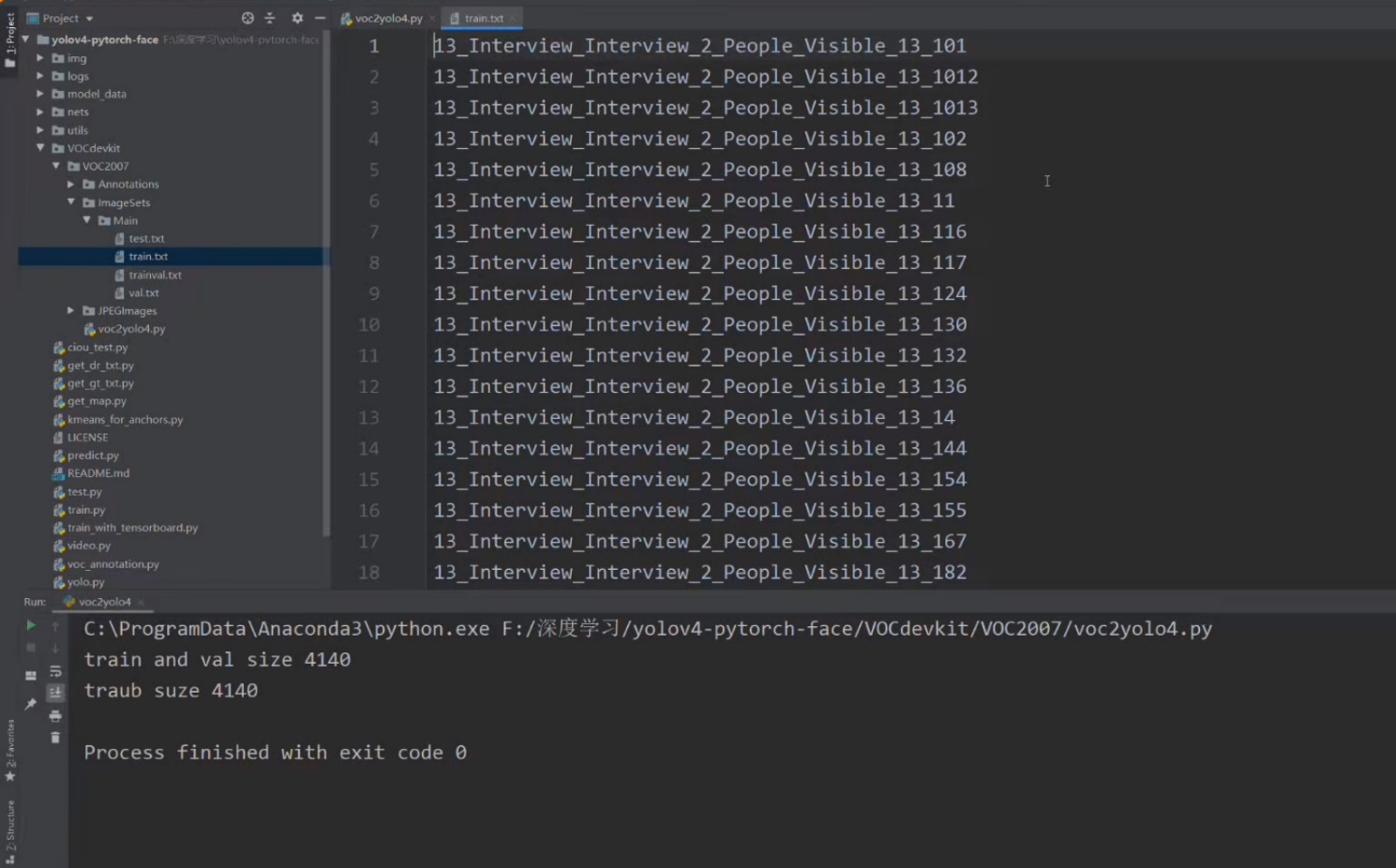
**NMS参考文章（2篇）**

<https://blog.csdn.net/qq_32618327/article/details/99173950?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522161866012816780274187418%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=161866012816780274187418&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_v2~hot_rank-3-99173950.first_rank_v2_pc_rank_v29&utm_term=%E9%9D%9E%E6%9E%81%E5%A4%A7%E5%80%BC%E6%8A%91%E5%88%B6>

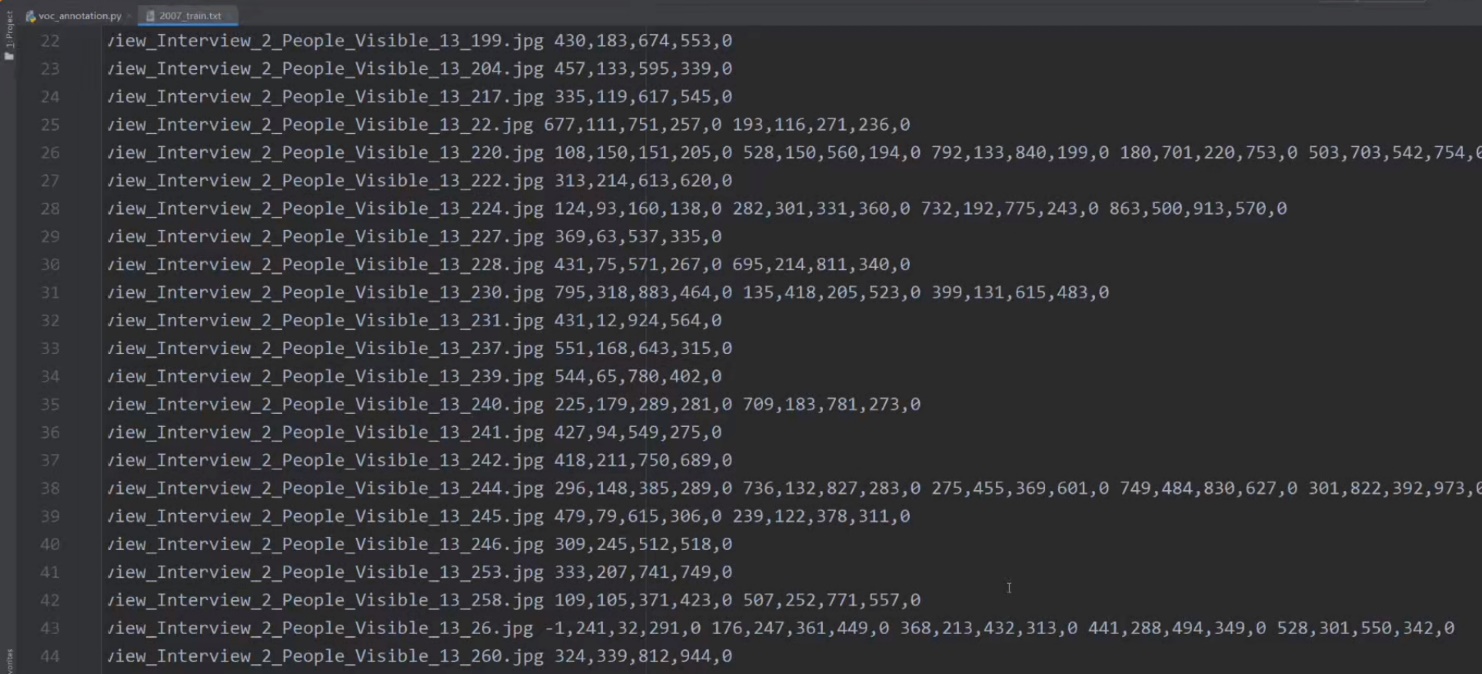
<https://blog.csdn.net/zchang81/article/details/70211851?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522161866032316780264015251%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=161866032316780264015251&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_v2~hot_rank-1-70211851.first_rank_v2_pc_rank_v29&utm_term=%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0%E2%80%94%E2%80%94NMS>



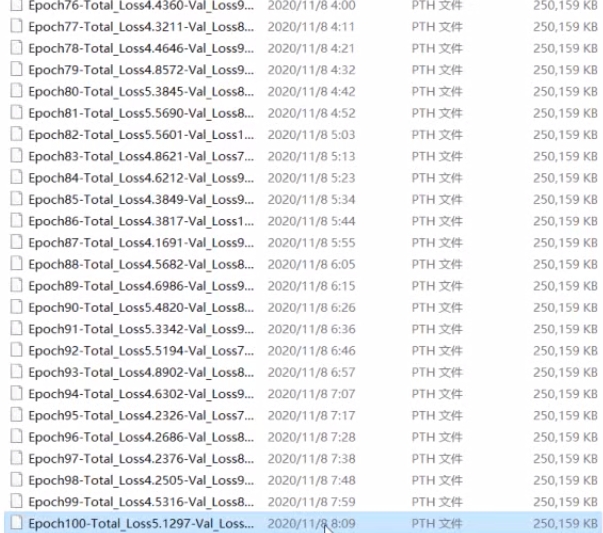
数据集



train.txt



2007\_train.txt



**训练得出logs文件夹下面的权值文件**



图片检测人脸结果



图片检测人脸结果



视频检测人脸结果