1. **能不能用一些很简单的话来解释一下YOLO的逻辑和物体检测还有风格转移是怎样实现的，或者说能对完全不了解这方面的观众解释清楚**

答：YOLO有三种进化变体，最初的YOLO，后来的YOLOv2和YOLOv3.本次方法使用了YOLOv2.YOLO是单阶段的目标检测网络，所以他的作用就是做目标检测的，只是本文案例比较特殊，只用于检测人脸这一种属性，但是也可以看作是检测任务，所以可以用YOLOv2。YOLO是单阶段检测，单阶段是因为他在检测人脸并找出人脸位置时，这个过程是一个连续的过程，并且都是通过回归来实现的。风格迁移的话建议去看网上的几篇博客就懂了，一句话讲的话就是，给定内容图片和风格图片，内容属于高层语义特征，风格属于低层的纹理等细节特征。要合成一张最终图片的话，这个最终图片需要抽取并保留内容图片的高层内容特征，然后要抽取并保留风格图片的低层纹理特征，然后将这两种特征混合，就得到了最终合成图。那么这个过程怎么优化呢，我们知道像VGG16这种cnn网络，作用其实就是提取特征，并且网络的浅层，比如第1，2，3，4层那种卷积，提取的就是一些比较低级层次的纹理特征，而到了后面的比较深的卷积层，比如第5，6，7等卷积层，就能够提取比较高级语义的内容特征了。所以在使用VGG16时，我们将传入的内容图片和特征图片分别卷积，但是我们只将低层的卷积特征图作为风格特征，将高层的卷积特征图作为内容特征。然后这两种特征分别有各自的loss，分别是风格loss和内容loss，将这两个loss合并起来作为最终的总loss，迭代优化的目标就是去最小化这个最终的loss使其最小，这时即可得到合成后的既保留了风格图片的风格特征，也保留了内容图片的内容特征的合成图。

（强烈建议还不太懂原理的话去看几篇博客就能理解了，三言两语只能大概的讲到上述程度~）

1. yolov1、yolov2、yolov3详解：

<https://blog.csdn.net/zxyhhjs2017/article/details/83013297>

（2）风格迁移概述和实现：

<https://www.jianshu.com/p/ef1d148b2605>

1. 然后还有detection出来的人脸头像到放到风格转移中做了什么预处理吗

答：没做预处理。想做预处理可以去了解一下人脸对齐，将脸统一扭回正脸。但是脸可能会有些变形，所以其实在这个人脸检测任务中不需要人脸识别的话可以不必要做人脸对齐。

1. **能不能单独跑face detection和style transfer的部分**

答：可以单独跑两部分。其实这个pipeline在实现时本来就是两部分任务分开的，先做完人脸检测得到人脸图片，再将这个图片传到风格迁移网络去生成最终的风格人脸图片。

1. **现在做出来的这些东西哪些是自己实现的哪些是用的别人的code，或者说自己具体做了什么**

答：可以理解为只用了别人预训练好的3个模型，分别是人脸检测的YOLOv2模型，以及风格迁移时的VGG16和风格模型。但是其实YOLOv2也是可以自己重新训练一遍的，如果时间来得及完全可以自己train一遍。VGG16是官方默认推荐使用的预训练提取特征网络，因为是在ImageNet上train几个星期那种，所以这个一般自己没必要训练。最后就是为了能够实时的风格迁移，需要提前训练好几个风格网络模型。但是因为没找到相关的风格图片数据集可以用来train，所以用了别人提供的train好的5种风格模型。同理，只要自己有风格数据集，自己也能train这些风格模型，但是因为目前自己没有这方面数据集，并且train一个风格网络估计也要几十个小时，所以这个风格模型是用了别人train好的。但是如何去train这个风格模型也写了train.py，所以只要能够收集到风格图片，我们自己也是能够train的。最后，整个pipeline都是自己重组写的，虽然其中会参考一些别人的代码。

1. **这个pipeline我可以可以这样理解：**
   1. 第一步：我拿了一个pretrained过的在VGG-16上的YOLO v2用来做通用物体识别模型，然后拿这个重新train在新的数据集上用来做人脸的识别，如果是的话这样算不算转移学习，然后再拿这个训练好的模型来预测脸
   2. 第二部：我拿了一个VGG-16的模型做style transfer，然后每个style都是在VGG-16上面train了一定数量的迭代，然后拿新的content image算出content cost进来直接和本身训练过的cost结合作为新的generated image的cost，然后根绝这个cost来生成weight，也就是每个像素

答：确实pipeline是两个主体部分，但是其实你的理解不太正确。

1. 第一步：用darknet框架，在FDDB数据集上训练了YOLOv2人脸检测网络，注意这里和VGG16没有任何联系。将训练得到的darknet框架的人脸检测模型转换为keras模型，就可以直接拿来predict图片并检测图片中的人脸。其实这里没有用到迁移学习，这里是相当于直接在FDDB上从零训练YOLOv2网络。除非用了在例如coco等数据集上预训练好的YOLOv2模型，然后在FDDB上进行fine-tune，这种就是迁移学习。但是本次方法是从0开始train我们的YOLOv2模型的。
2. 第二步：这一步你的理解节本没错。首先用预训练好的VGG-16模型，来train出风格模型，这个就是迁移学习了，因为VGG-16以前是在分类任务中训练好的，现在用在style transfer任务上，是两个不同的任务，所以可以理解为迁移学习。后面你的理解基本是对的。具体的细节可以再看多几个网络的博客会清晰很多。
3. **能不能出一些可视化的东西，不管是dataset上的还是tensorboard来做**

答：因为自己没有时间去train过上面提到的三个网络，所以train的过程中的可视化东西没有怎么得到。不过我有自己train过100轮迭代的中间效果图，可以拿来和最终的迁移的效果图来做对比。另外我在网上也看到了一个很好的将中间结果图做成的一个gif动图，将中间训练的结果图片慢慢的展现出来，那个可视化效果不错，可以发给你作为pre的演示参考。

1. **头像检测中有没有训练集的evaluation作为参考**

答：头像检测只给了一个train时的loss下降曲线。目前没有去特别测试过evaluation效果。