[toc]

10.21-10.24(第一周)

任务1. 尝试跑一下示例图片，提取和理解图片每一层的特征信息

matlab 知识补充

matlab 函数

- randn() ：产生一个符合指定均值和方差的正态分布的矩阵。

  如 randn(10,10,3,2,'single') 指创建一个单精度值的

matconvnet 的 CNN wrappers

- 第一个封装器是SimpleNN，大部分是通过 MATLAB 函数 vl _ SimpleNN 实现，适用于具有线性拓扑结构的网络，即由计算块组成的链。
- The second wrapper is DagNN, which is implemented as the MATLAB class dagnn.DagNN

师姐获得每一层的feature map是依靠vl_SimpleNN封装后所提供的res。看了部分vl_SimpleNN文件，认为会限制上限，决定仍使用pytorch。

pytorch 部分

torch.squeeze() : 这个函数主要对数据的维度进行压缩，去掉维数为1的的维度，比如是一行或者一列这种，一个一行三列（1,3）的数去掉第一个维数为一的维度之后就变成（3）行。squeeze(a)就是将a中所有为1的维度删掉。不为1的维度没有影响。a.squeeze(N) 就是去掉a中指定的维数为一的维度。

torch.unsqueeze() : 这个函数主要是对数据维度进行扩充。给指定位置加上维数为一的维度，比如原本有个三行的数据（3），在0的位置加了一维就变成一行三列（1,3）。a.squeeze(N) 就是在a中指定位置N加上一个维数为1的维度。

使用tensorboard可视化tensor格式的图片

```
!pip install tf-nightly-gpu -i http://pypi.douban.com/simple --trusted-host
pypi.douban.com
!pip install future -i http://pypi.douban.com/simple --trusted-host
pypi.douban.com
```

```
from torch.utils.tensorboard import SummaryWriter
writer = SummaryWriter("./logs/fit")

%load_ext tensorboard
%tensorboard --logdir ./logs/fit

# 用 writer.add_image("名字",input_tensor) 添加
```

> !!! pytorch官网中下载的vgg-16（categories是imagenet_classes）并不是vgg-face，不要误会了

所以问题，出现了 ： pytorch 如何读取t7格式的模型？

过去有

```
from torch.utils.serialization import load_lua
x = load_lua('x.t7')
```

但pytorch在1.0之后删除了torch.utils.serialization，目前可以通过torchfile.load读取，但会报错：

```
TypeError: unhashable type: 'numpy.ndarray'
```

As of PyTorch 1.0 torch.utils.serialization is completely removed. Hence no one can import models from Lua Torch into PyTorch anymore. Instead, I would suggest installing PyTorch 0.4.1 through pip in a conda environment (so that you can remove it after this) and use this repo to convert your Lua Torch model to PyTorch model, not just the torch.nn.legacy model that you cannot use for training. Then use PyTorch 1.xx to do whatever with it. You can also train your converted Lua Torch models in PyTorch this way 😃 来源

但尝试失败，包括之后尝试的三个github的repo：PyVGGFace 、convert_torch_to_pytorch和 vgg-face.pytorch(issue中作者提到他仍能在linux中运行（今年8月)），希望转化成能用的形式，均因为相同的原因失败。

发现网页Samuel Albanie,能下载网络框架的py文件，但不能下载含有权重信息的.pth文件

尝试使用GitHub上的caffemodel2pytorch(这玩意获得proto是通过request的，本地的prototxt文件读不进去)和Caffe2Pytorch(易用，但是0.4.1以上版本没有torch.legacy，而使用anaconda激活的虚拟环境中，pytorch0.4.1报错No module named “caffe”)

困难的真正原因是，之前的torch是使用lua语言，之后在2017年根据python重构了代码变成pytorch，而vgg-face的作者提供的是torch模型，而不是pytorch的模型。VGGface2是支持的，还是因为vggface有些年份了。

在一个issues里向owner发表评论请求邮件模型

依着上面提到的vgg-face.pytorch的issue的思路，使用linux，但是工位电脑的root账号密码丢失，最终在自己电脑的Ubuntu子系统使用PyVGGFace，成功得到vggface.pth

正式开始任务（已经是10.27了）

如果直接使用torch.load导入：

```
model = torch.load(r"D:\Dataset\models\vggface.pth")
model.eval()
```

```
AttributeError: 'collections.OrderedDict' object has no attribute 'eval'
```

原因是这仅是字典形式的权重数据，没有模型的实体。想用来自Samuel Albanie的模型框架文件获得转化后的权重数据，报错如下，可以看出是字典的键值不匹配。

```
RuntimeError: Error(s) in loading state_dict for Vgg_face_dag:
    Missing key(s) in state_dict: "conv_1_1.weight", "conv_1_1.bias",
"conv1_2.weight", "conv1_2.bias", "conv2_1.weight", "conv2_1.bias",
"conv2_2.weight", "conv2_2.bias", "conv3_1.weight", "conv3_1.bias",
"conv3_2.weight", "conv3_2.bias", "conv3_3.weight", "conv3_3.bias",
"conv4_1.weight", "conv4_1.bias", "conv4_2.weight", "conv4_2.bias",
"conv4_3.weight", "conv4_3.bias", "conv5_1.weight", "conv5_1.bias",
"conv5_2.weight", "conv5_2.bias", "conv5_3.weight", "conv5_3.bias", "fc6.weight",
"fc6.bias", "fc7.weight", "fc7.bias", "fc8.weight", "fc8.bias".
    Unexpected key(s) in state_dict: "features.conv_1_1.weight",
"features.conv_1_1.bias", "features.conv_1_2.weight", "features.conv_1_2.bias",
"features.conv_2_1.weight", "features.conv_2_1.bias", "features.conv_2_2.weight",
"features.conv_2_2.bias", "features.conv_3_1.weight", "features.conv_3_1.bias",
"features.conv_3_2.weight", "features.conv_3_2.bias", "features.conv_3_3.weight",
"features.conv_3_3.bias", "features.conv_4_1.weight", "features.conv_4_1.bias",
"features.conv_4_2.weight", "features.conv_4_2.bias", "features.conv_4_3.weight",
"features.conv_4_3.bias", "features.conv_5_1.weight", "features.conv_5_1.bias",
"features.conv_5_2.weight", "features.conv_5_2.bias", "features.conv_5_3.weight",
"features.conv_5_3.bias", "fc.fc6.weight", "fc.fc6.bias", "fc.fc7.weight",
"fc.fc7.bias", "fc.fc8.weight", "fc.fc8.bias".
```

实测不能通过在定义层时在conv前添加features.xxx解决

```
self.add_module("features.conv_1_1",nn.Conv2d(3, 64, kernel_size=[3, 3], stride=
(1, 1), padding=(1, 1)))
```

```
KeyError: 'module name can\'t contain ".", got: features.conv_1_1'
```

写函数,得到新字典后成功获得完整模型实例

```
def transform_key(model):
    #列表来自报错内容，可复制进来更改，创建一个新字典，将key 从post_names -> names
    names = ["conv1_1.weight","conv1_1.bias", ......]
    post_names = ["features.conv_1_1.weight", "features.conv_1_1.bias", ......]

    from collections import OrderedDict
    new_state_dict = OrderedDict()
    for i in range(0,len(model)):
        name = names[i]
```

```
            new_state_dict[name] = model[post_names[i]]
        return new_state_dict
```

```
Vgg_face_dag(
  (conv1_1): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (relu1_1): ReLU(inplace=True)
  (conv1_2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (relu1_2): ReLU(inplace=True)
  (pool1): MaxPool2d(kernel_size=[2, 2], stride=[2, 2], padding=0, dilation=1,
ceil_mode=False)
  (conv2_1): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (relu2_1): ReLU(inplace=True)
  (conv2_2): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (relu2_2): ReLU(inplace=True)
  (pool2): MaxPool2d(kernel_size=[2, 2], stride=[2, 2], padding=0, dilation=1,
ceil_mode=False)
  (conv3_1): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (relu3_1): ReLU(inplace=True)
  (conv3_2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (relu3_2): ReLU(inplace=True)
  (conv3_3): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (relu3_3): ReLU(inplace=True)
  (pool3): MaxPool2d(kernel_size=[2, 2], stride=[2, 2], padding=0, dilation=1,
ceil_mode=False)
  (conv4_1): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (relu4_1): ReLU(inplace=True)
  (conv4_2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (relu4_2): ReLU(inplace=True)
  (conv4_3): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (relu4_3): ReLU(inplace=True)
  (pool4): MaxPool2d(kernel_size=[2, 2], stride=[2, 2], padding=0, dilation=1,
ceil_mode=False)
  (conv5_1): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (relu5_1): ReLU(inplace=True)
  (conv5_2): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (relu5_2): ReLU(inplace=True)
  (conv5_3): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (relu5_3): ReLU(inplace=True)
  (pool5): MaxPool2d(kernel_size=[2, 2], stride=[2, 2], padding=0, dilation=1,
ceil_mode=False)
  (fc6): Linear(in_features=25088, out_features=4096, bias=True)
  (relu6): ReLU(inplace=True)
  (dropout6): Dropout(p=0.5, inplace=False)
  (fc7): Linear(in_features=4096, out_features=4096, bias=True)
  (relu7): ReLU(inplace=True)
  (dropout7): Dropout(p=0.5, inplace=False)
  (fc8): Linear(in_features=4096, out_features=2622, bias=True)
)
```

接下来，如果直接想要运行vgg_face_dag(image) 是不行的：

- 首先是会

```
TypeError: __init__() takes 1 positional argument but 2 were given
```

原因是模型是作为类对象存在的，首先要先实例化类

```
vggface = Vgg_face_dag()
# 之后可以以 vggface() 调用
# 因为多def的函数会调用这个类，实际上我使用的是下面的语句
vggface = vgg_face(weights_path= r"D:\Dataset\models\vggface.pth")
```

- 其次是模型是通过nn.Module类封装的，需要输入一个tensor格式的四维张量，即使是单张图片也要制备成mini batch，如下

```
input_image = Image.open(r"D:\Dataset\VGG-1\dataset\Adelaide_Kane\3.jpg")
preprocess = transforms.Compose([
    transforms.Resize(256),
    transforms.CenterCrop(224),
    transforms.ToTensor(),  # 转化为tensor格式
    transforms.Normalize(mean=[129.186279296875, 104.76238250732422,
93.59396362304688], std=[1, 1, 1]),
])  # mean和std是使用的模型写在self.meta位置的数据，不确定是否正确
input_tensor = preprocess(input_image)
input_batch = input_tensor.unsqueeze(0) # create a mini-batch as expected by the
model
```

但我在测试该图片后发现，就分类结果来看，前五的probabilities都很低，猜测是数据传输出了一些问题，导致权重数据实际上没有被使用，跑的是一个随机初始化的网络，尝试debug, 学习并使用ipdb，暂时并没有发现有bug

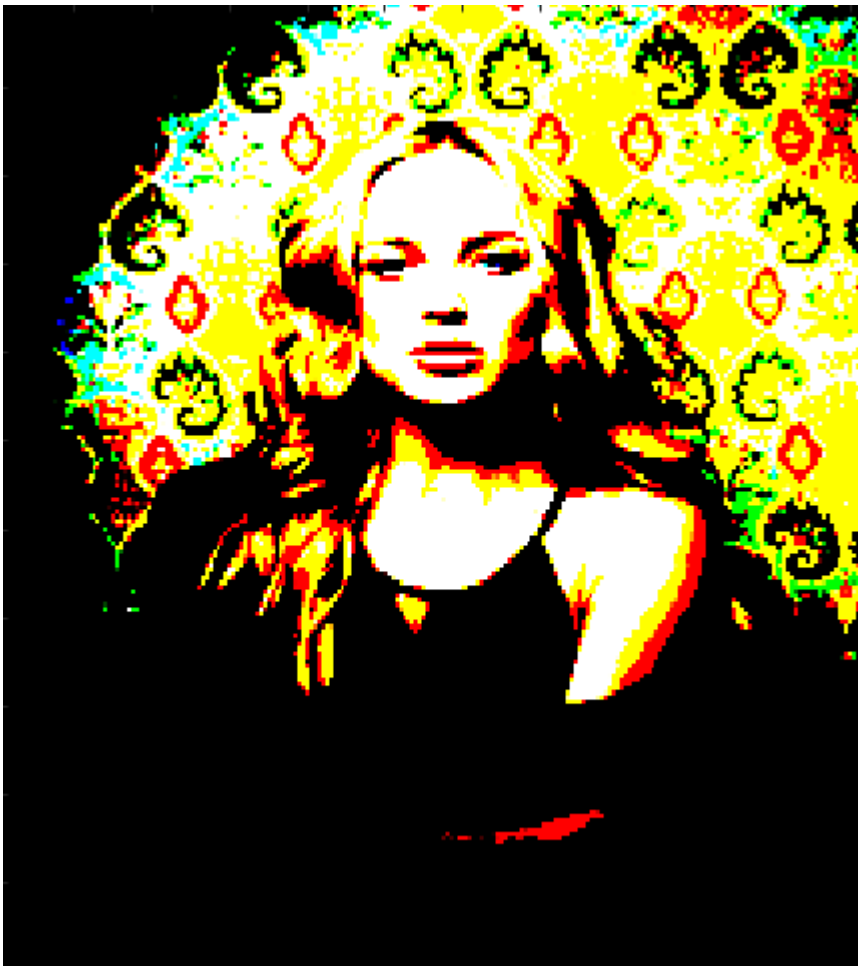
```
from IPython.core.debugger import set_trace
# 利用 set_trace() 在代码中插入断点，交互模式下调试
```

意识到预处理方式与文献不同，其中图片文件中含有面部框的 left top right bottom 的位置，不过提供的数据是含有小数的，不确定该数据的原点位置，表示单位应该是像素，推测小数是源于面部抓取的程序给出的计算结果。

就时间上来看，问题不一定能在短时间处理掉。为了在下次meeting前做出些结果，暂时转向matlab。阅读从师姐那要来的Q2releaseVggFaceO.m

我的预处理结果(上)，根据matconvernet这几句得到的结果（下），虽然预处理是对于RGB通道的，就结果对比来看，对神经网络的分类影响不小，训练时使用了，使用模型跑图片时还应使用相同的预处理方式。（之后学习到，这种更改RGB通道的预处理叫图片增强，和用于避免梯度饱和的标准化不相同，后来研究发现使用这种方法意义不大）

```
im_ = single(im) ;
im_ = imresize(im_,net.meta.normalization.imageSize(1:2)) ;
im_ = im_-net.meta.normalization.averageImage ;
```



学习裁剪方式（PIL包，img.crop）和归一化方法，matlab中按照它训练预处理方式就能够得到正确分类结果，正确率也高。

matlab 笔记 续

```
%%  # 代码分块执行

net.meta  是用来记录如何normalize的，执行时会被忽略
```

任务2、尝试找到VGG-face的训练集图片

VGG-face 论文

VGG Face Dataset 需要通过里面的url下载

文章有一个数据搜集的过程

1. 第一阶段是确定候选人名单，考虑到便于在网络上找到足够数量的不同图像，并避免下载图像时出现任何隐私问题，选择演员或政客。先从Internet Movie Data Base(IMDB)中按受欢迎排名提取了列表。（5K个名字的候选列表，我们有这些名字的属性信息，比如种族、年龄、亲属关系等(来自Freebase knowledge graph，全都有覆盖))

   过滤掉没有足够多清晰图片的候选人（多少足够？多清晰清晰？），并清除掉与标准基准库重复的图像。

之后每个候选者200张图像交由人类注释者，当90%以上都是该候选者时才保留，剩下3250位候选者。移去与LFW和YTF数据集重名的候选者，最终2,622。

2. 为每个候选者搜集更多图片，从Google、bing中搜索“人名”和“演员”+“人名”，每个获得500张，共计每人多出2000张。

3. 用支持向量机去掉所有错分类的面孔， （每个身份的top 50张图（基于谷歌搜索等级）用作正训练样本，所有其他身份的top 50作为负训练样本。训练的one-vs-rest linear SVM对该身份的2000张图片进行排名，取前1000张。

4. Near duplicates (e.g. images differing only in colour balance, or with text superimposed) are also removed.

   通过计算每个图像的 VLAD 描述符，使用一个非常紧密的阈值在每个标识的1,000个图像中对这些描述符进行聚类，并在每个集群中保留元素。

5. 为减轻注释任务的负担和开销，使用AlexNet训练multi-way CNN去用softmax给图片排序，200张一块，注释员注释后，如果纯度大于95%，为好。 The final number of good images is 982,803.

plus ：下载该数据库

编写脚本 giturl_vgg.ipynb,在工位电脑的jupyter lab上。 先尝试爬取第一个候选人的所有照片，发现如下问题：

- 部分图片（暂定为A类）连接超时，超时重连无效（重连还浪费时间），下载时将其跳过，计划再写个函数将跳过的收集起来。

  部分图片（暂定为B类）下载后显示图片错误，无法打开（已知原因：所里网络工作时间屏蔽）

  第一个候选者1000张图片中下载694张图像，即A类有306张，目测B类占一半，所以约有350张左右可用图片。

- 目前下载速度一般

- 肉眼可观察到作为人脸数据库，仍有一些简笔画或logo这样与人脸完全无关的内容；也有其他人的脸（包括性别和种族都不同）；同样也有一张图片中多个人脸的图片

- 候选者本人的图片大量的重复

- 有一个图片来源，ImageCollect , 其图片都带有大面积 、一致的水印，应当会对结果产生影响，推测如果使用爬虫的cookies模拟登录访问该网站，可能不会带有水印

- 如何进行反爬虫？

进一步的问题：

与老师交流过后，意识到作为2015年的数据集，这些图片url所指向的连接也许坏掉了。目前对face数据集有何种程度的了解？

任务3. 参考文献第一点，考虑对模型熟悉/不熟悉的图片进行模糊后用模型处理，计算特征相似度

10.25-10.31(第二周)

任务1、训练集图片下载

使用上周码的脚本，从B开头的人开始下载

plus ： 尝试提高爬虫性能

- 继续使用request但是添加 多线程
- 大部分时间开销在请求和相应
- 或许学下scrapy或者feapder（个人更相中feapder）

  feapder github位置、中文手册

封装的有些复杂，重新按照框架码代码比较麻烦，暂时尝试向之前码的脚本中添加多线程（#还未做，照顾到电脑两天停一天，目前共下载157个对象）。

任务2、Q1尝试用模型进行处理

阅读文献

- Masking in Visual Recognition: Effects of Two-Dimensional Filtered Noise
- L. D. Harmon,The recognition of faces
- A survey on heterogeneous face recognition: Sketch, infra-red, 3D and low-resolution
- 2014 NIPS How transferable are features in deep neural

任务3、尝试用matlab自带的toolbox进行模型的随机初始化

实际上已经能使用pytorch对vggface模型随机初始化了，再尝试使用一个图形界面交互的工具箱意义不大

阶段总结

针对于这个阶段的任务和我的了解，总结一些问题和想法：

- 我是意识到我自己对于编程语言还是有些执拗的，在深度学习方面，想要使用pytorch以至于有点抵制matlab，原因更多的是python在这个领域的广泛应用（高占比），对于构建模型更灵活，我认为至少tensorflow和pytorch是要掌握一个的。但在当下的任务上，如果说无需训练模型、只是用已有模型观察表征，特别是具体到vggface，matconvernet和pytorch之间对vggface易用性的支持差距是非常巨大的（但我不实际趟过也不知道），之后我尽量会使用更方便的方式，灵活一点。

  补充，matlab对GPU支持难搞

- 另一个问题是个人是有些技术主义偏向，之前meeting老师专门和我解释的我理解，之后的研究也是围绕人类认知研究深度神经网络的可解释性，我的想法自然是算力资源充足更好（有点火力不足恐惧症的意思），cpu肯定是无法胜任深度学习的，从师兄那了解到组里服务器显卡是10系的泰坦，在这方面我的实践经验没多少，只能说根据实际需求走一步看一步。

- 关于师姐目前做的研究（指10/18工作汇报ppt），具体到Q1(低分辨率)，个人认为如果只是对单一模型查看识别的正确率随模糊程度下降的趋势意义不大，我目前的观点是至少要比对多个预训练的用于面孔识别的模型的表现（1），特别是要包含目前性能最强面孔识别模型、用于在非约束（uncontrolled scenarios，关键词unconstrained face recognition，其中一项就是低分辨率）场景的面孔识别模型、专门用于识别低分辨率人脸的模型（2）。并且模型之间不应该是仅是数据库类型、规模大小的差别，更应该是结构上的区别，雨萌师兄说的神经动力学我了解甚少，我对自组织更感兴趣一些。
    - （1）师姐ppt也提到其他层以及其他模型如AlexNet就没有出现"相比于不熟悉的面孔，更能够在低分辨率下识别出熟悉的面孔"的结果，实际AlexNet模型训练是用于物体识别，这一点也许是与人类识别物体和识别面孔的特征不同相关，印象中人识别物体没有更容易认出熟悉物体的现象（记不清楚了，有待求证）
    - （2）目前我知道有专门的高分辨率和低分辨率结合的人脸数据库（如下图），注意到其中的低分辨率图像更多的是源于监控录像这样的应用场景。

TABLE 1. Summary of unconstrained datasets for very low resolution face recognition. Taken and complemented from [53]. Most of the available datasets come from real-world surveillance imagery and contain high resolution and low resolution pairs, except QMUL-surface and QMUL-Tinyface. However, these two datasets contain the largest number of images and identities, making them suitable training deep learning methods.

Database name	Source	Quality	Static image/video	# subjects	# images
Point and Shoot [4]	Manually Collected	HR + blur	static + video	558	12,178
SCface [29]	Surveillance	HR + LR	static	130	4,160
QMUL-Survface [16]	Surveillance	LR	static + video	15,573	463,507
QMUL-TinyFace [14]	Web	LR	static	5,139	169,403
UCCSface [84]	Surveillance	HR + blur	static	308	6,337
IJB-S [48]	Surveillance	HR + LR	static + video	202	3 million+

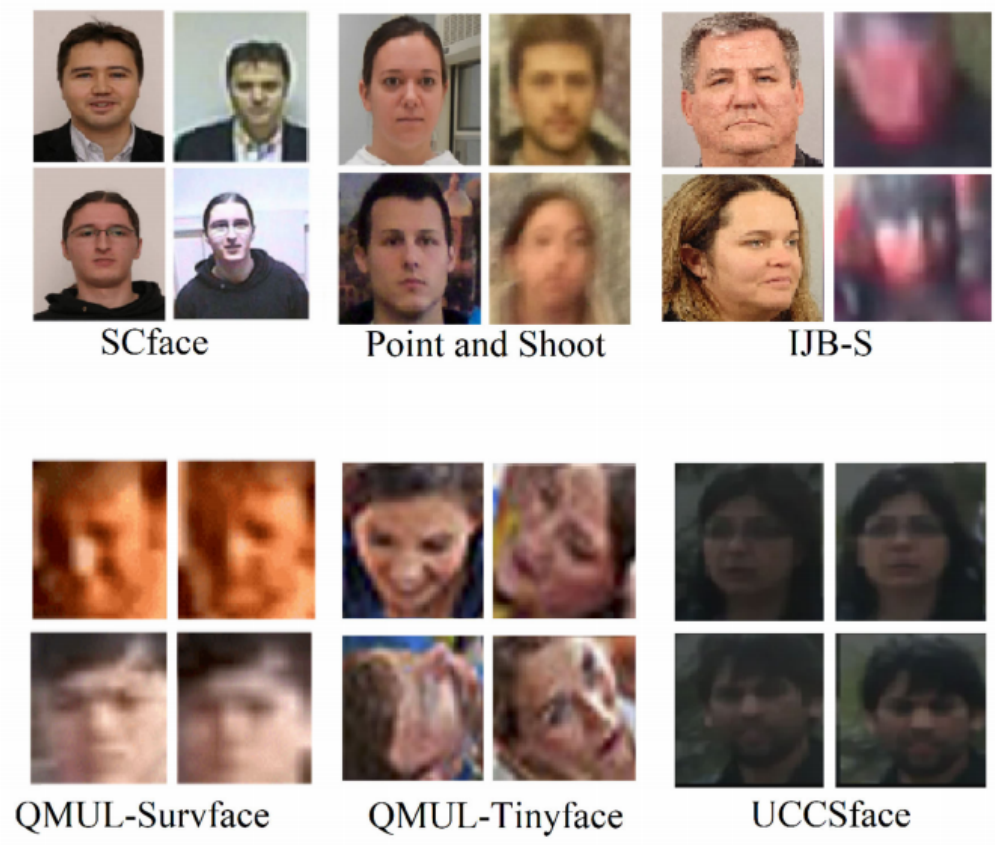
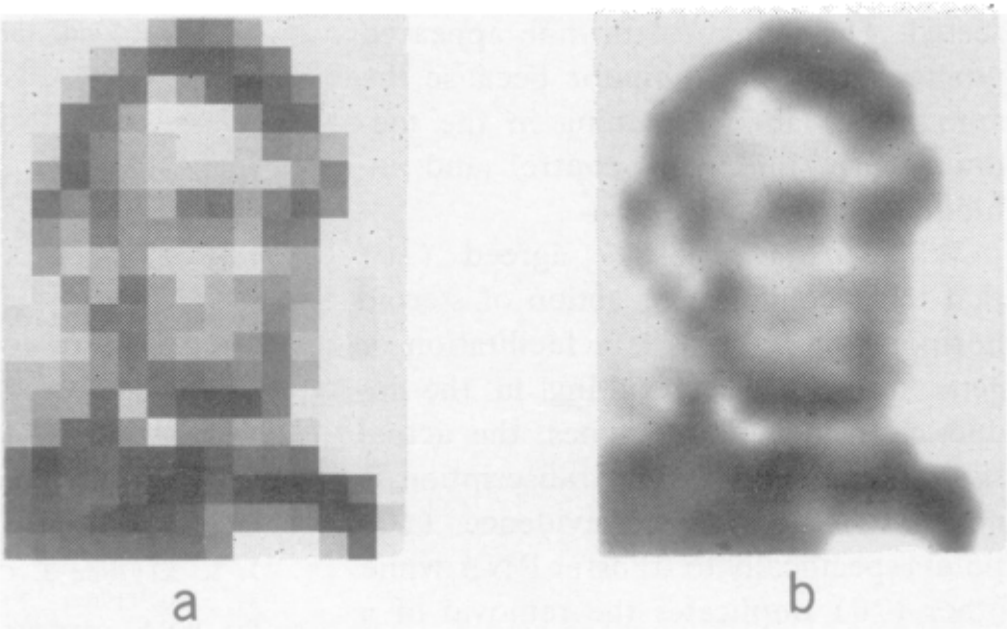


FIGURE 1. Example of subjects in the different datasets available for very low resolution face recognition, taken from their respective dataset papers [4], [16], [29], [48], [14], and [84]. The SCface [29], Point and Shoot [4], IJB-S [48], and UCCSface [84] are the most suitable for heterogeneous face recognition, because they supply the HR and VLR image pairs for each identity. The QMUL-Survface [16] and QMUL-Tinyface [14] are suitable for the homogenous face recognition problem, where only VLR images are available.

我暂时不清楚摄像机导致的图像模糊（采样率不足）、出于研究目的人为处理的图像模糊（比如1973年那两篇文献中带有很多历史色彩的像素化方式）和实际肉眼看到的模糊（近视、距离远、存在半透明阻挡物）有什么异同， 从目前我得知的，还是有区别的，文章 Masking in Visual Recognition:Effects of Two-Dimensional Filtered Noise 里提到将像素化的图片低通滤波之后（a到b），人的识别能力提升。从图片上来看，自然环境是很难出现a这种模糊的（想到的只有透着格子玻璃），它格子边界高频信息的变化对识别有干扰，结合已有的视觉上关于感受野的研究，a是属于将斜向激活感受野细胞能获得的信息去掉了，格子边界强化了原来在这个地方激活很弱或者没有激活的垂直和水平感受野细胞。 从这个理解出发，我认为图片信息量下降的方式即模糊方式是影响人对面孔的认知的。

而我对于摄像机导致的图像模糊（采样率不足）目前的理解，也是像素化的模糊，是否有量化非像素化模糊图片的方式？ 将图像像素化，之后再处理（比如高斯模糊），它的模糊程度应该可以用像素化这个过程量化（即16X16 像素点比32X32像素点模糊4倍）？



- 之前有阅读一篇 cosface 的文章，还没读完，是弄了个名为large margin cosine loss的损失函数。想法的来源是总结了一些当时使用新损失函数的论文，发现共同点是一个思想，即 maximizing inter-class variance and minimizing intra-class variance。这个思想也是对面孔空间模型的延伸，它表示距离还是基于L2范数（即欧几里得空间）。
    - 由此我想，我们的研究，其结果应当是什么样的，怎样才能实质上对神经生物学方面的神经机制和 人工智能方面的神经网络有启发？

接下来我想先了解下计算机是怎么捕获面部的，总之多读文献。如果可以的话，敲定几个模型，按照师姐的方式探索下Q1的问题。
