周报　2022.5.13

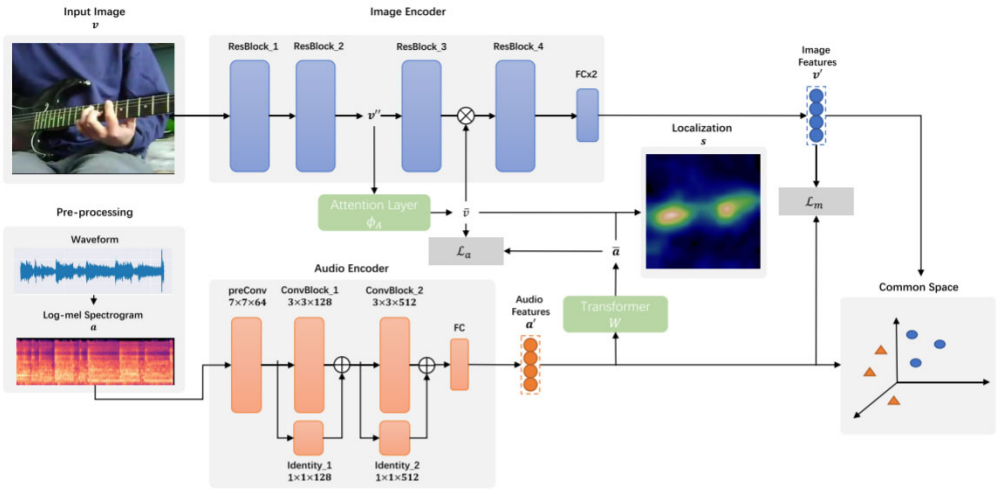
一、 完成工作

这周主要看了两篇论文，一篇是 Look，Listen and Infer（本组贾瑞坚学长发表在 2020MM 上的跨模态文章（不光有跨模态检索，还可以完成零样本图片识别以及零样本声音定位的任务））。还有一篇是 Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval，这篇是 ICCV2021 的文章主要是将图片视频一起使用进行跨模态文本检索。

1. LLINET

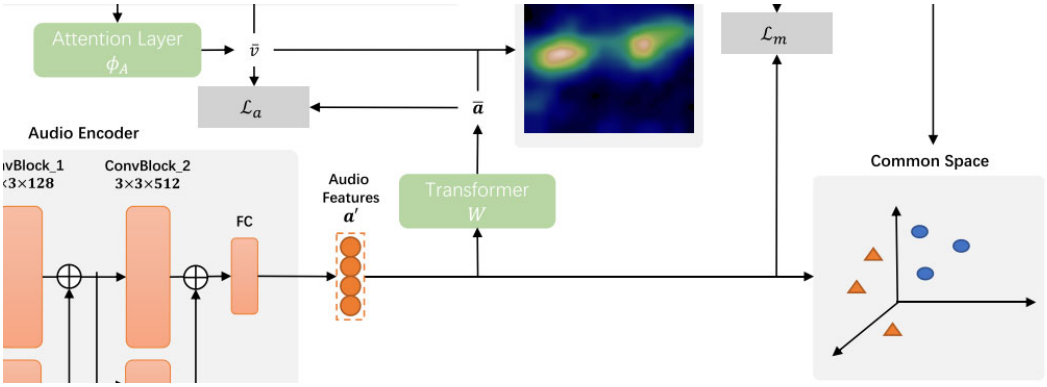
   这是一篇关于零样本跨膜态检索以及声音定位的文章。贾学长这篇文章的贡献还是比较多的，首先论文做了一个数据集 instrument-32CLASS，这个数据集里的音频是 AudioSet 里的一个子集，将数据集 10 秒的数据截断到 3 秒，而图像是从相应的视频中截取的图像帧作为代表，生成了图像-音频对，由于该数据集用来完成零样本学习，所以训练集和测试集不能有交集，所以分为了 24 个训练类和 8 个测试类。除此以外还提供了 segmentation，用于声音定位的评估。

第二个贡献就是学长提出的 LLINET 模型，该模型有三个模块，分别是音频编码器，视觉编码器和注意力模块。



其中，音频编码模块是比较有特点的，首先，将表示音频的波形图转换为梅尔频谱图，这是比较好的处理方法，之后，我记得以前的音频有关的网络都是将音频视为时序信息，然后使用 RNN 来处理，但是这里学长将转换之后的频谱图直接看作单通道图像作为音频编码器的输入。编码器的结构这里就不详细说了，主要还是应用了 ResBlock 的结构，最后输出了 1024-d 的向量。视觉编码器也是用了 ResNet-101 的结构。

另一个部分是注意力模块，注意力模块由两部分组成:



可以看到，注意力模块主要有两部分，AttentionLayer 和 Transformer，attention layer 是和图片编码器第三个残差块结构一样的，但是没有冻结，可训练，而 Transformer 是一个 1024 单元的全连接层。可以看到，作者将图片编码器第二个残差块的特征提出来，然后和 audio 特征投影到一个公共空间，这样就可以计算图片每个局部位置和整体音频特征的相似度，专业昂就可以得到一个注意力图，也就是模型图上的 s，这样就可以实现声音定位。

然后就到了论文的损失函数部分：整体损失包括两个，分别是匹配损失和注意力损失。首先是匹配损失，这个损失主要用于将图像和音频表示在公共空间中对齐图像和音频表示。总体来说，就是让匹配的对在公共空间中具有更大的相似度，不匹配的较小。

相似度定义为：

$$P_{i,j} = \begin{cases} 0, & y_i = y_j \& i \neq j \\ \frac{v_i'^T \cdot a_j'}{\|v_i'\| \|a_j'\|}, & \text{otherwise} \end{cases}$$

就是余弦相似度。

然后：

ed as:

$$\mathcal{L}_{IA} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} Q_{i,j} log P'_{i,j},$$

$'$ is formulated by normalizing P with the softmax fu
s an identity matrix of size $N \times N$ that labels the
tion between image-audio pairs. Specifically, P' an
ed as follows:

$$P'_{i,j} = \frac{\exp(P_{i,j})}{\sum_{j=1}^{N} \exp(P_{i,j})}$$

$$Q_{i,j} = \begin{cases} 1, & i = j \\ 0, & \text{otherwise} \end{cases}$$

整体还是一个交叉熵损失函数的格式，我们知道交叉熵是描述 Q 和 P 的相似性，这样该损失函数就可以描述图片到音频的损失，反过来也可求得音频到图像的损失，。

$$\mathcal{L}_m = \mathcal{L}_{IA} + \mathcal{L}_{AI}$$

而 Attention Loss 主要用于将图像区域和音频关联起来，通过 attention 可以得到 16*16*1024 也就是 256 个局部特征，然后将局部特征与经过 transformer 的音频特征相乘，就得到了注意力图。

$$s_i = \bar{v}_i{}^T \bar{a}.$$

然后，我们就可以实现注意力模式，也就是 QKV，其中权重就是我们通过注意力图来得到的，然后 element wise 一下就可以得到权重后的 v。

$$\hat{v} = (s' + 1) \odot \bar{v},$$

然后，损失就可以顺理成章了，也就是可以通过交叉熵获取得到损失函数。

$$\mathcal{L}_{IA}^{(a)} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} Q_{i,j} log P_{i,j}'^{(a)},$$

整体损失和匹配损失相似。然后和匹配损失相加就得到了整体损失。

然后方法就结束了。

我觉得学长的方法还是很强的，主要就是音频编码器和注意力模块这里，让我得到了该如何设计注意力模块来得到好用的注意力模块。

效果提升也很大。

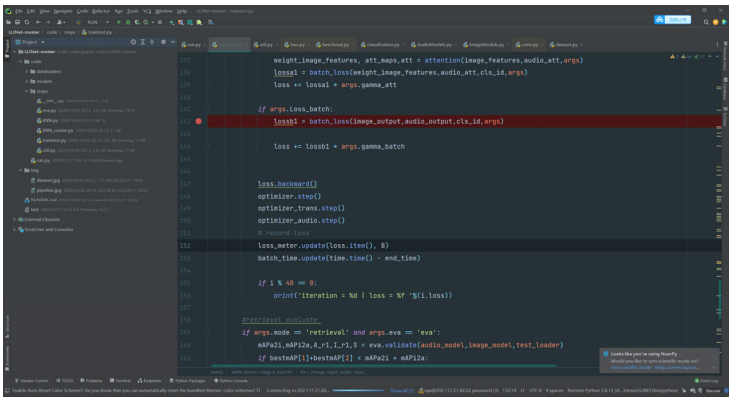
Table 1: Performance comparison (in %) with state-of-the-art cross-modal retrieval approaches on the INSTRUMENT-32CLASS dataset.

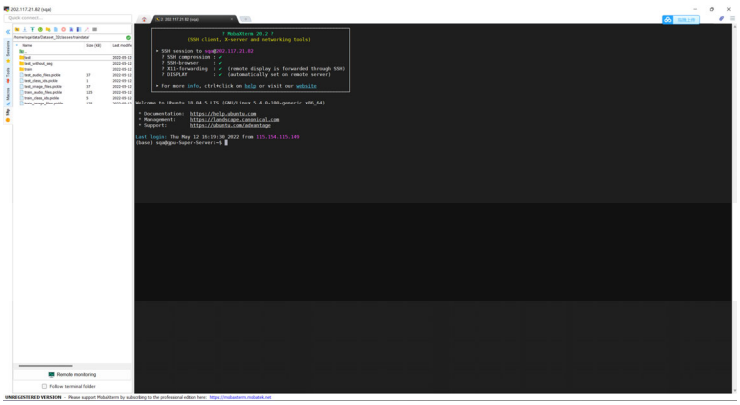
Model	Image to Audio		Audio to Image	
	mAP	R@1	mAP	R@1
DAR [4]	13.3	13.4	13.6	15.3
ULSLVC [12]	13.4	8.2	20.1	21.0
SIR [26]	21.2	20.9	15.4	12.8
SCAN [21]	16.5	16.7	18.2	15.0
DSCMR [45]	14.8	20.5	18.1	24.6
TIMAM [31]	39.1	28.5	25.8	25.0
CMPM [42]	39.3	28.9	26.2	25.8
CME [35]	30.1	35.7	27.4	30.8
LLINet	**49.3**	**41.1**	**31.2**	**38.3**

学长的方法主要有特点的就是音频编码器和匹配损失，细节前面已经说了。

二、 下周工作

除此之外，我还详细读了学长的代码，并且试着训练了一下，准备下一周试着想一些 idea 来进行改进，看有没有什么可以进一步改进的研究方向。除此之外，我还会继续进行 DOLG 那边的工作，看看改进效果如何。


weight_image_features, att_maps,att = attention(image_features,audio_att,args)
loss1 = batch_loss(weight_image_features,audio_att,cls_id,args)
loss += loss1 * args.gamma_att
if args.Loss_batch:
loss1 = batch_loss(image_output,audio_output,cls_id,args)
loss += loss1 * args.gamma_batch
loss.backward()
optimizer.step()
optimizer_trans.step()
optimizer_audio.step()
# record loss
loss_meter.update(loss.item(), B)
batch_time.update(time.time() - end_time)
if i % 40 == 0:
print('iteration = %d | loss = %f '%(i,loss))
#retrieval evaluate
if args.mode == 'retrieval' and args.eva == 'eva':
mAPa2i,mAPi2a,A_r1,I_r1,5 = eva.validate(audio_model,image_model,test_loader)
if bestmAP[1]+bestmAP[2] < mAPa2i + mAPi2a:


Welcome to MobaXterm 20.2 !
(SSH client, X-server and networking tools)
SSH session to xyq@202.117.21.82
SSH compression : ✓
SSH-browser : ✓
X11-forwarding : ✓ (remote display is forwarded through SSH)
DISPLAY : ✓ (automatically set on remote server)
For more info, ctrl+click on help or visit our website
Last login: Thu May 12 16:19:36 2022 from 115.154.115.149
(base) xyq@gpu-Super-Server:~$
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net