多媒体基础 2023 秋 结课报告

姓名：　　学号：

一、问答题：

假设你身处于 2001 年，此时 Google 公司已创立三年，百度公司已创立一年半，它们针对互联网的文本内容实现了建库和搜索，并从广告业务中获取了巨额利润。受此启发，你也计划创立一家公司，对互联网上的图片内容进行建库和搜索，请描述你公司的核心技术架构，请专注于在技术层面。（本题没有标准答案，请发挥你的想象力，所有看似合理的技术方案都可以）

答：公司名称：亿图图新

核心技术架构：

1.图片抓取与识别技术：亿图图新的核心技术之一是图片抓取与识别技术。我们使用先进的网络爬虫，能够自动抓取互联网上各类图片资源。同时，利用图像识别技术和深度学习，我们可以对抓取的图片进行分类、标签和索引，以便用户进行搜索。

2.图片搜索算法：我们的搜索算法基于深度学习，能够理解图片的内容并进行搜索。例如，用户可以上传一张图片，我们的算法就能在我们的数据库中找到相似的图片。此外，我们还会根据图片的标签和元数据提供相关的搜索建议。

3.大规模分布式存储系统：为了存储大量的图片数据，我们设计了一个大规模分布式存储系统。这个系统可以高效地存储和检索图片，同时也支持数据的备份和恢复。

4.实时索引技术：亿图图新还采用了实时索引技术，能够实时更新图片的索引信息，确保用户搜索到的内容是最新的。

5.人工智能辅助内容推荐：除了基础的图片搜索功能，亿图图新还利用人工智能技术进行内容推荐。根据用户的搜索历史和浏览行为，我们会为用户推荐相关的图片和主题。

6.安全与隐私保护：我们非常重视用户的数据安全和隐私保护。因此，我们在数据传输和存储过程中采用了先进的安全技术，如数据加密和访问控制，以确保用户数据的安全。

7.云计算基础设施：为了支持大规模的数据处理和高并发的用户请求，我们采用了云计算基础设施。通过云计算，我们可以灵活地扩展我们的计算和存储资源，以满足用户的需求。

以上就是亿图图新的核心技术架构。通过这些技术，我们希望能够为用户提供更高效、更准确的图片搜索服务。

二、编程题

请查看项目文件夹中的示例数据：《The Godfather I (1972)》片段

The Godfather I

编程语言：不做要求，C/C++、Python、Matlab 均可，请选用自己熟悉的语言，可以使用库函数。

1）数据处理：使用 ffmpeg 工具包（https://ffmpeg.org/download.html），将上述视频解析为一张一张的图片，解码时请将 fps 设置为 5 或者 10 即可，否则生成图片数量过多，请展示 ffmpeg 解析的命令行代码，并展示示例视频所解压的**首帧、中间帧**和**末尾帧**。

答：ffmpeg -i ./godfather.mp4 -vf "fps=10" ./pic/%4d.png

下面分别展示首帧（0001.png）、中间帧（0158.png）、末尾帧（0501.png）

首帧：0001.png

中间帧：0158.png

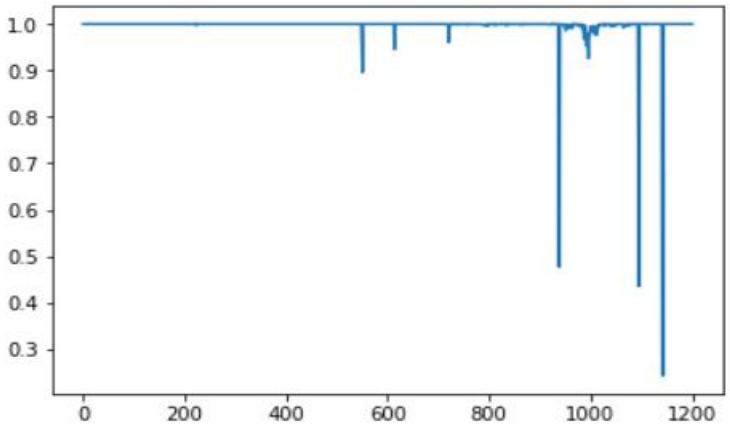
末尾帧：0501.png

2）镜头和场景分割：观察获取的图像帧，请说明视频中有哪些位置（按原视频中的时间）出现了镜头变换，并指出变换的类型，然后请将相同场景的镜头放在一起，以**层次结构图**展示。

编程：使用基于**彩色直方图的方法**检测镜头边界，并附上代码和帧间差值的柱状图展示，设定合适阈值后展示所检测到的镜头变换位置。并请说明基于直方图的镜头变换检测方法可以如何进一步改进？

答：

直方图如下图所示：



镜头变化的层次结构：

第一帧

第二帧

第三帧

第四帧

第五帧

第六帧

关键代码如下：

```
#计算直方图函数
def Computer_Histograms(frame):
    hist = cv.calcHist(images = [frame],channels = [0,1,2],mask = None,histSize = [8,8,8],
                       ranges = [0,256,0,256,0,256])
    hist = hist.flatten()
    hist = hist / hist.sum()
    return hist

#计算每帧图片的直方图
hist_list = []
for each in frame_list:
    hist = Computer_Histograms(each)
    hist_list.append(hist)
#基于相关系数比较直方图
correl_list = []
for i in range(len(hist_list)-1):
    correl = cv.compareHist(hist_list[i], hist_list[i+1], cv.HISTCMP_CORREL)
    correl_list.append(correl)
print(len(correl_list))

#基于相关系数的比较直方图相似度图像
fig,ax = plt.subplots()
ax.plot(correl_list)
plt.show()
```

为了进一步改进基于直方图的镜头变换检测方法，可以考虑以下几个方面：

1.使用更复杂的直方图比较算法：除了简单的帧间差值比较外，还可以尝试使用更复杂的直方图比较算法，如 Histogram Intersection 或 Earth Mover's Distance 等，以提高检测的准确性和鲁棒性。

2.考虑颜色空间和色彩特征：在计算直方图时，可以考虑使用不同的颜色空间（如 HSV 或 YCrCb），并提取更多的色彩特征，以提高检测的准确性和稳定性。

3.使用深度学习模型：深度学习模型在图像处理和计算机视觉领域表现出色，可以考虑使用深度学习模型来检测镜头变换位置。例如，可以使用卷积神经网络（CNN）或迁移学习等技术来训练一个分类器，用于判断镜头是否发生变换。这种方法可能能够提供更准确和可靠的结果。

3） MPEG 压缩实验：请**选择一个镜头**作为数据，将该镜头中的**首帧、中间帧和末尾帧**，设置为 MPEG 压缩中的 I 帧。

编程一：对选择的三个 I 帧图像做 JPEG 压缩，即①转换为 YUV 颜色空间，②将图像分为 8x8 的块，③对每个块做 DCT 变换，④对获得的 DCT 系数做量化，⑤z 字形编码，⑥用 zlib 压缩。请展示关键代码，和代码运行结果的证明数据（不需要很多，展示代码的运行结果即可）。

关键代码：

```
def jpeg_compress(image_path, quality=90):
    # 读取图像
    image = cv2.imread(image_path)

    # 转换为YUV颜色空间
    yuv_image = cv2.cvtColor(image, cv2.COLOR_BGR2YUV)

    # 将图像分为8x8的块
    blocks = [yuv_image[i:i + 8, j:j + 8] for i in range(0, yuv_image.shape[0], 8) for j in range(0, yuv_image.shape[1], 8)]

    # 对每个块做DCT变换
    dct_blocks = [cv2.dct(block.astype(np.float32)) for block in blocks]

    # 对获得的DCT系数做量化
    quantized_blocks = [np.round(dct_block * quality / 100).astype(np.uint8) for dct_block in dct_blocks]

    # z字形编码
    zigzagged_blocks = [zigzag(block) for block in quantized_blocks]

    # 用zlib压缩
    compressed_blocks = [zlib.compress(block.tobytes()) for block in zigzagged_blocks]

    # 将压缩的块拼接回图像并返回
    compressed_image = np.zeros((yuv_image.shape[0], yuv_image.shape[1], 3), dtype=np.uint8)
    for i, block in enumerate(zigzagged_blocks):
        compressed_image[i % yuv_image.shape[0]:i % yuv_image.shape[0] + 8, i // yuv_image.shape[0]:i // yuv_image.shape[0] + 8] = np.frombuff
    return compressed_image

def zigzag(block):
    # Zigzag扫描函数，用于将DCT系数从二维数组转换为一维数组
    scan = [[0, 1, 8, 16, 9, 2, 3, 10],
            [17, 24, 32, 25, 18, 11, 4, 5],
            [12, 19, 26, 33, 40, 48, 41, 34],
            [27, 34, 41, 48, 55, 62, 56, 63],
            [35, 42, 57, 64,  5,  6,  7]]
    return np.concatenate([block[scan[i][j]] for i in range(4) for j in range(4)])
```

编程二：选择**中间帧对应的 I 帧图像**，选择其**后面一帧作为 P 帧**，实现 P 帧的压缩算法，①对 P 帧中的每个 8x8 图像块，在其参考 I 帧中**对应位置**的周围 48x64 的范围内的所有的 8x8 图像块，计算均方误差(MSE)，选择 MSE 最小的块作为最佳匹配块。②计算 P 帧中每个 8x8 图像块与其最佳匹配块之间的差值，对差值重复 I 帧的编码过程。截取关键代码片段并解释其功能，需要有中间结果数据作为辅助说明。展示 I 帧相比于其原始 RGB 数据，压缩率是多少，P 帧的压缩率是多少？

答：关键代码：

```
# 对P帧进行分块和匹配
block_size = 8
num_blocks_x = p_frame.shape[0] // block_size
num_blocks_y = p_frame.shape[1] // block_size

for i in range(num_blocks_x):
    for j in range(num_blocks_y):
        block = p_frame[i*block_size:(i+1)*block_size, j*block_size:(j+1)*block_size]
        best_match = None
        min_mse = float('inf')
        search_range = 48
        search_start = max(0, i - search_range)
        search_end = min(num_blocks_x, i + search_range)
        for k in range(search_start, search_end):
            for l in range(search_start, search_end):
                reference_block = reference_frame[k*block_size:(k+1)*block_size, l*block_size:(l+1)*block_size]
                mse = np.mean((block - reference_block) ** 2)
                if mse < min_mse:
                    min_mse = mse
                    best_match = (k, l)
        # 这里可以进一步处理最佳匹配块和当前块的差值，并进行编码
        print(f"Block at ({i}, {j}) best matched with block at ({best_match[0]}, {best_match[1]}) with MSE: {min_mse}")
```

部分运行结果：

```
Block at (0, 0) best matched with block at (26, 41) with MSE: 31.140625
Block at (0, 1) best matched with block at (1, 45) with MSE: 25.015625
Block at (0, 2) best matched with block at (20, 41) with MSE: 24.46875
Block at (0, 3) best matched with block at (45, 37) with MSE: 24.890625
Block at (0, 4) best matched with block at (42, 4) with MSE: 2.453125
Block at (0, 5) best matched with block at (1, 29) with MSE: 4.0
Block at (0, 6) best matched with block at (0, 37) with MSE: 2.5625
Block at (0, 7) best matched with block at (1, 37) with MSE: 2.109375
Block at (0, 8) best matched with block at (30, 39) with MSE: 2.21875
Block at (0, 9) best matched with block at (43, 4) with MSE: 1.296875
Block at (0, 10) best matched with block at (28, 19) with MSE: 6.375
Block at (0, 11) best matched with block at (27, 28) with MSE: 1.265625
Block at (0, 12) best matched with block at (17, 40) with MSE: 5.578125
Block at (0, 13) best matched with block at (11, 45) with MSE: 2.375
```

三、综述题：

选择多媒体领域的某一个你感兴趣的研究方向，通过检索近 3 年（2020-2023）的最新研究成果，撰写一篇综述报告，综述报告的，具体要求如下：

（1）检索文献来源限定为： ACM MM（ACM International Conference on Multimedia）、AAAI（National Conference on Artificial Intelligence）、VLDB(International Conference on Very Large Data Bases)、ICCV(International Conference on Computer Vision)、CVPR(IEEE Conference on Computer Vision and Pattern Recognition)、ICASSP（International Conference on Acoustics, Speech, and Signal Processing）、ICMR(ACM International Conference on Multimedia Retrieval)、ICML（International Conference on Machine Learning）、ICME（International Conference on Multimedia & Expo）、ICIP（International Conference on Image Processing）、CIKM（International Conference on Information and Knowledge Management）、ICDM（IEEE International Conference on Data Mining）、DCC(Data Compression Conference)、MM(International Multimedia Modeling Conference)或者计算机学会列出的期刊或者会议列表（http://www.ccf.org.cn/sites/ccf/biaodan.jsp?contentId=2567518742937）。

（2）参考文献的篇数不少于 3 篇。

（3）字数不少于 2000 字。

（4）内容包括：研究现状、典型算法、存在的问题、未来的研究热点、参考文献。

标题：AI 赋能多媒体领域

一、研究现状:通过文献阅读，总结所选择的研究方向的最新研究进展。

随着人工智能（AI）技术的迅猛发展，其已在计算机多媒体领域得到了广泛应用。AI 技术不仅提高了多媒体内容的生成效率和个性化，也提升了多媒体数据的分析和理解能力。以下是一些应用 AI 技术的计算机多媒体领域的具体例子:

1.图像识别：这是 AI 在计算机多媒体领域中最广泛的应用之一。通过深度学习和神经网络等技术，AI 可以对图像进行精确的识别、分类和检索。例如，在安防监控领域，AI 可以实现人脸识别、物体识别等功能，帮助警方快速定位和追踪目标；在电商领域，AI 可以通过图像识别技术实现商品自动分类、智能推荐等功能，提升用户体验。

2.语音识别：AI 技术可以将音频数据转化为文本，实现语音转文字、语音翻译等功能。在语音助手、智能客服等领域，AI 技术可以帮助人们更高效地处理语音信息，提高语音交互的准确性和便捷性。

3.自然语言处理：AI 技术可以对自然语言文本进行分析和处理，例如文本分类、情感分析、机器翻译等。在社交媒体监测、舆情分析等领域，AI 可以帮助人们快速处理和分析大量的文本信息，提供有价值的洞察和预测。

4.视频分析：AI 可以对视频内容进行自动分析和理解，例如目标检测、场景分类、行为识别等。在安防监控、智能交通等领域，AI 可以帮助人们实时监测和预警潜在的安全风险；在智能电视、智能家居等领域，AI 可以根据用户的行为和兴趣推荐个性化的内容和服务。

5.虚拟现实：AI 技术可以提升虚拟现实的真实感和交互性，例如通过图像识别和跟踪技术实现更准确的头部定位和手势识别；通过语音识别和自然语言处理技术实现更自然的语音交互。

总之，随着 AI 技术的不断发展，其在计算机多媒体领域的应用也将越来越广泛和深入。通过结合多媒体数据的特点和需求，AI 技术有望为人们带来更加智能化、高效化和个性化的多媒体服务和体验。

二、典型算法：介绍三种典型的算法，给出算法的流程图，比较三种方法的优缺点。

算法一： DA-GAT, a double attention framework based on the graph attention network for the multi-label classification task.算法流程图如图 2.1 所示。

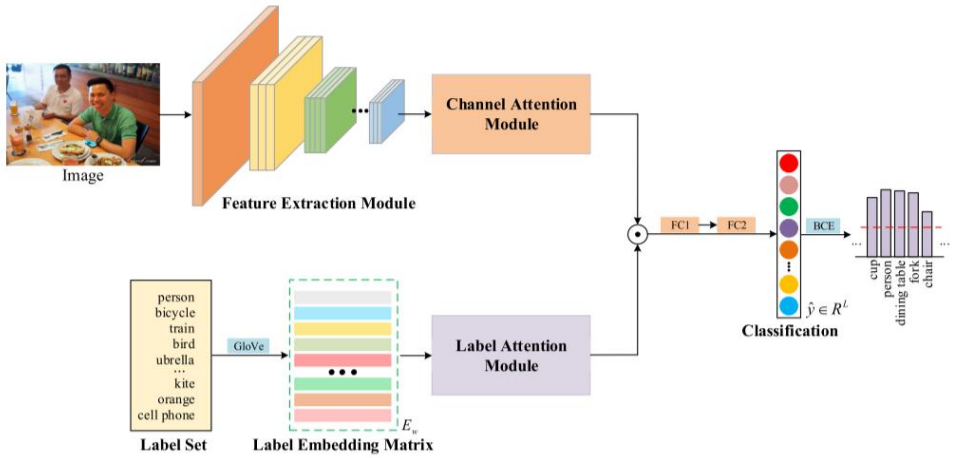


图 2.1

DA-GAT 模式的总体框架。特征提取模块利用卷积神经网络提取图像特征。通道注意模块利用图注意网络增强分组的高级特征通道图之间的相关性。标签注意模块使用图形注意网络来捕捉标签之间的相关性。然后将两个注意模块融合，得到最终的模型结构。图片和标签来自 MS-COCO 2014 数据集。GloVe，预训练的 GloVe[29]模型;FC，全连接层;BCE，二元交叉熵。

算法二： A dual hierarchical learning method to tackle the problem of HSI classification. 算法流程图如图 2.2 所示。

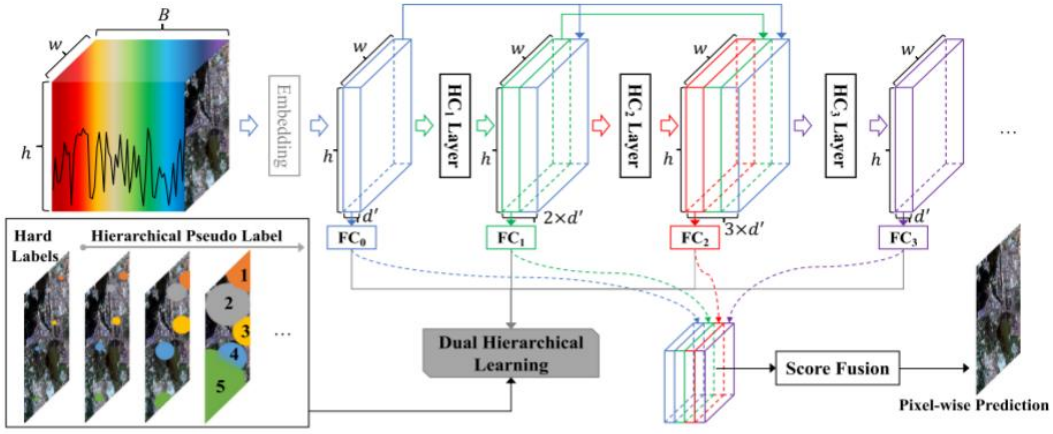
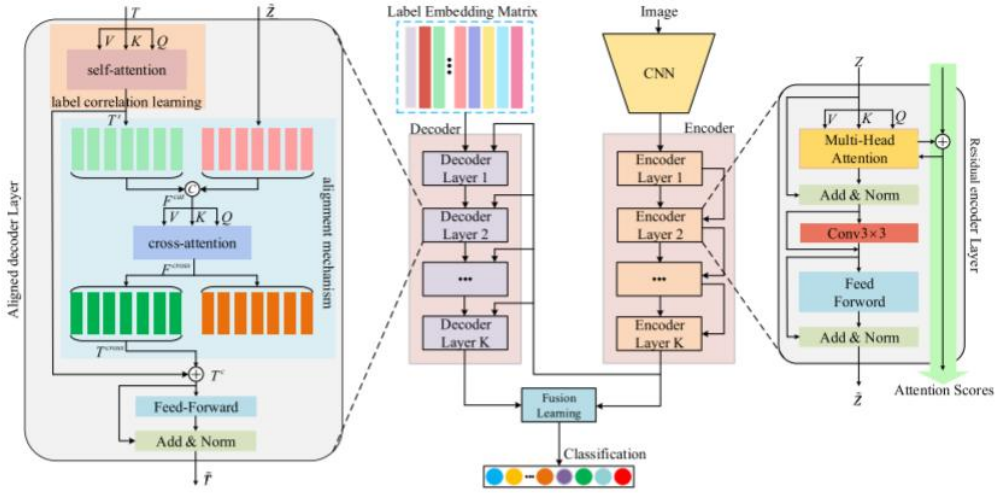


图 2.2

给定一幅带有标记训练像素的高光谱图像，我们使用多层 HC 计算不同接收域的区域。然后，我们将前面所有层的输出连接起来，并将它们作为当前计算层的输入。也就是说，较深的 HC 层不仅有较大的接受域，而且还包含其他接受域的内容。同时，我们利用高光谱像素的特性，从给定的样本中生成分层的伪标签，并利用这些伪标签来丰富监督。最后，我们开发了一种双重层次的训练优化策略，该策略利用了多层结构和伪标签来改善学习过程。在推理阶段，利用融合策略寻找更可靠的像素预测。

算法三： A novel image multi-label classification framework which aims to align Image Semantics with Label Concepts (ISLC). 算法流程图如图 2.3 所示。



特征提取模块利用 CNN 提取图像特征。剩余编码模块学习图像的显著语义特征。对齐解码器中的自注意层捕获标签之间的相关性，对齐解码器中的交叉注意层对图像语义特征和标签概念进行对齐。将编码器和解码器的最后一层输出进行融合，得到最终的输出特征。

三、存在问题：分析现有方法存在的问题。

1.数据稀疏性：在许多多媒体应用中，标注的数据往往是有限的，这导致了数据稀疏性的问题。如何利用无标注数据进行有效的训练和学习是当前面临的一个挑战。

2.多模态融合：多媒体数据通常包括图像、音频、视频等多种模态，如何实现这些模态之间的有效融合，以及从多模态数据中提取更丰富和准确的信息是一个重要的研究方向。

3.可解释性和鲁棒性：现有的许多 AI 模型往往是黑箱模型，难以解释其决策和行为。同时，由于数据分布的差异和噪声干扰，模型的鲁棒性也是一个需要解决的问题。

4.隐私和安全：随着 AI 技术在多媒体领域的应用越来越广泛，隐私和安全问题也越来越突出。如何在保护用户隐私的同时实现有效的多媒体处理和分析是需要关注的问题。

四、未来的研究热点

1.深度学习：深度学习作为AI的核心技术，其在多媒体领域的未来应用前景广阔。如何设计更有效的深度学习模型，以及如何利用无标注数据进行训练和学习是未来的重要研究方向。

2.强化学习：强化学习是一种通过试错进行学习的机器学习方法，其在多媒体领域的应用还处于起步阶段。如何设计适用于多媒体任务的强化学习算法是未来的一个研究热点。

3.联邦学习：随着AI技术的发展，数据隐私和安全问题越来越受到关注。联邦学习是一种在保护用户隐私的同时实现有效模型训练的方法，其在多媒体领域的应用前景广阔。

4.可解释性和鲁棒性：随着AI技术的发展，可解释性和鲁棒性成为了一个重要的研究方向。如何设计可解释性强、鲁棒性好的AI模型是未来的一个研究热点。

5.多模态融合：多模态融合是多媒体领域的一个重要研究方向。如何实现多模态数据的有效融合，以及如何利用多模态数据提取更丰富和准确的信息是未来的一个研究热点。

综上所述，随着AI技术的不断发展，其在计算机多媒体领域的应用也将越来越广泛和深入。未来需要关注的问题包括解决现有方法存在的问题、探索新的研究方向等，以期为人们带来更加智能化、高效化和个性化的多媒体服务和体验。

参考文献：（按照规范列出参考文献）

[1] Zhou W , Xia Z , Dou P ,et al.Double Attention Based on Graph Attention Network for Image Multi-Label Classification[J].ACM Transactions on Multimedia Computing, Communications and Applications, 2022, 19:1 - 23.DOI:10.1145/3519030.

[2] Zhou W, Xia Z, Dou P, et al. Aligning image semantics and label concepts for image multi-label classification[J]. ACM Transactions on Multimedia Computing, Communications and Applications, 2023, 19(2): 1-23.

[3] Wang S , Ben H , Hao Y ,et al.Boosting Hyperspectral Image Classification with Dual Hierarchical Learning[J].ACM transactions on multimedia computing communications and applications, 2023.

[4] Zhou W, Hou Y, Chen D, et al. Attention-augmented memory network for image multi-label classification[J]. ACM Transactions on Multimedia Computing, Communications and Applications, 2023, 19(3): 1-24.