# 大数据特色模块课程

# 结题文档

- **课题名称:** 风格随机化的图形数据增强算法和图文联合数据增强算法的开发
- **小组成员:** 刘东洋、蔡依璇、郭怡彤

## 目录：

- 技术细节

- 实验与评估

## 5. 课堂问题解答

- 问题 1

- 问题 2

- 问题 3

## 6. 结论与展望

- 项目总结

- 未来研究方向

### 一: 课题的主要内容

在深度学习领域，数据增强是一种常用的技术，旨在通过**扩大训练集**来提高模型的泛化能力。本课题聚焦于开发两种创新的数据增强算法：风格随机化的图形数据增强算法和图文联合数据增强算法，旨在进一步提升机器学习模型在复杂环境中的性能和鲁棒性。

### 二: 课题的任务

**本课题的核心任务围绕两种数据增强算法的开发与验证：风格随机化的图形数据增强算法和图文联合数据增强算法。以下是针对每种算法的具体任务概述：**

**•风格随机化的图形数据增强算法:**

1. **算法设计与实现**：开发基于样式迁移的数据增强算法，该算法能够调整图像的风格（颜色和纹理）而保持其形状和语义内容不变。这需要深入研究和应用最新的样式迁移技术。
2. **性能评估**：通过在多个计算机视觉任务上的实验来评估风格随机化数据增强算法的效果，包括但不限于**图像分类、物体检测和语义分割。**
3. **领域偏移与过拟合分析**：分析算法在减少领域偏移和过拟合方面的效能，验证其对提高模型在未见领域数据上的泛化能力的贡献。
4. **实践指南的制定**：基于实验结果，制定针对不同任务和数据集的风格随机化参数选择和应用的最佳实践指南。

**•图文多模态数据增强算法任务:**

1. **算法设计与实现**：开发一种新的图文联合数据增强方法，该方法能够有效地融合视觉和语言信息，以产生丰富的多模态训练样本。
2. **跨任务泛化能力验证**：在不同的下游任务（如**图像标注、视觉问答、图像文本匹配**等）上验证算法增强的模型的性能，以展现其对提升模型多模态学习能力和数据效率的影响。
3. **多模态表示学习分析**：深入分析通过图文联合数据增强学习到的多模态潜在表示，了解算法如何帮助模型捕捉图像和文本之间的复杂关系。
4. **应用案例研究**：探索和记录算法在具体应用场景中的使用案例，例如在有限数据环境下的性能提升，以及在特定领域（如医疗图像处理、社交媒体分析等）的应用效果。

## 三：风格随机化的图形数据增强算法研究

### 1.风格随机化的图形数据增强算法:

**- 原理简述：**

风格随机化的图形数据增强算法基于**样式迁移**的概念，其中样式迁移网络用于调整输入图像的风格（颜色和纹理）而不改变其形状和语义内容。算法的核心思想是利用深度神经网络学习到的图像表示来分离和重组图像的“内容”与“风格”特征。

**- 技术细节：**

**1.1 风格迁移：**

**内容与风格的分离**是风格迁移技术的核心部分，使用预训练的卷积神经网络 VGGNet 来实现。风格被表示为一组 Gram 矩阵，描述了低级卷积特征之间的相关性；而内容则由高级语义特征的原始值表示。

**（1）总损失函数**

总损失函数是内容损失、风格损失以及它们之间权重的组合，用来指导风格迁移过程。这个总损失函数可以表示为：

$$\boldsymbol{L_{total}}=\boldsymbol{\gamma L_{content}+\beta L_{style}}$$

$\gamma,\beta$是调节内容和风格重要性的超参数。

$\gamma$控制内容保持的强度，较大的$\alpha$使得生成图像更接近原始图像内容。

$\beta$控制风格匹配的强度，较大的$\beta$促进风格图像的风格特征在生成图像中的呈现。

通过调整这两个超参数，可以平衡内容和风格的影响，从而控制生成图像与内容图像和风格图像的相似度。

整个过程通过梯度下降算法来最小化总损失函数，不断更新生成图像，直至找到一个在保持原始内容的同时，又具有目标风格特征的图像。

**（2）预训练网络**

预训练网络选择：选择 **VGGNet** 作为预训练网络，因为它在图像识别和分类任务中表现优秀，已被广泛验证能有效提取图像的层次特征。VGGNet 的每一层都对输入图像的不同特征有不同的响应，这些特征随网络深度增加而变得越来越抽象。

**特征提取**：当输入一张图像到 VGGNet 时，网络会通过其多个卷积和池化层逐步处理图像，每一层都会输出一组特征图（feature maps），这些特征图代表了网络在当前层对图像的理解或解释。

网络的浅层（比如 VGG 的前几个卷积块）捕获图像的低级特征，如纹理、颜色和局部形状等。这些特征用于表征图像的风格。
网络的深层（比如 VGG 的第四卷积块之后的层）捕获高级抽象的特征，这些特征更多关注于图像的内容而非具体的纹理或颜色。例如，它们可能表示图像中存在的对象或对象的部分，而忽略这些对象的具体样式或外观细节。
**内容损失**：通过比较原始内容图像和生成图像在这些深层的响应差异来定义内容损失，目的是保持生成图像与原始内容图像在内容上的相似性。
**风格损失**：风格通过计算特征图的格拉姆矩阵（Gram matrix）来表示，**格拉姆矩阵捕获了特征之间的相关性**，反映了图像的纹理信息。生成图像的风格损失是通过比较原始风格图像和生成图像的**格拉姆矩阵的差异**来计算的，目的是使生成图像在风格上与目标风格图像相似。

### 1.2 样式转移管道：

样式转移管道机制：

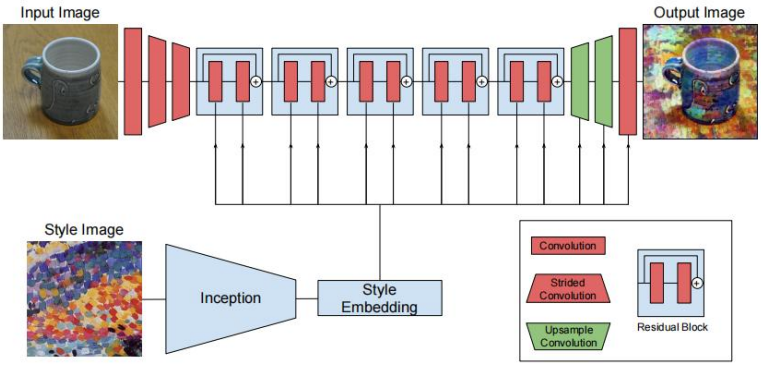



Figure 2: Diagram of the arbitrary style transfer pipeline of Ghiasi et al. [10].

**样式转移管道**通过采用**风格预测网络**来观察任意风格图像并输出风格嵌入 z。风格嵌入 z 通过条件实例归一化影响变压器网络的作用，其中激活通道基于样式嵌入进行了移位和重新缩放。
具体地说，如果 x 是归一化之前的特征映射，则**重正化**的特征映射如下：

$$x' = \gamma(\frac{x - \mu}{\sigma}) + \beta,$$

其中，μ和 σ 分别为特征图空间轴上的均值和标准差， β 和 γ 为将样式嵌入通过全连接层得到的标量。

### 1.3 随机化程序：

对上述样式转移管道添加随机化程序，使得网络能够生成多种风格的图片。首先引入随机采样的样式嵌入，该样式嵌入是从正态分布中进行抽取而来的。
为了提供对增强强度的控制，可以将随机采样的样式嵌入与输入图像的样式嵌入 P (c).进行**线性插值**:

$$z = \alpha\, \mathcal{N}(\mu, \Sigma) + (1 - \alpha) P(c)$$

其中 N 是正态分布中抽取得到的嵌入，， P（c）是输入图像的风格嵌入。 α 是样式转移

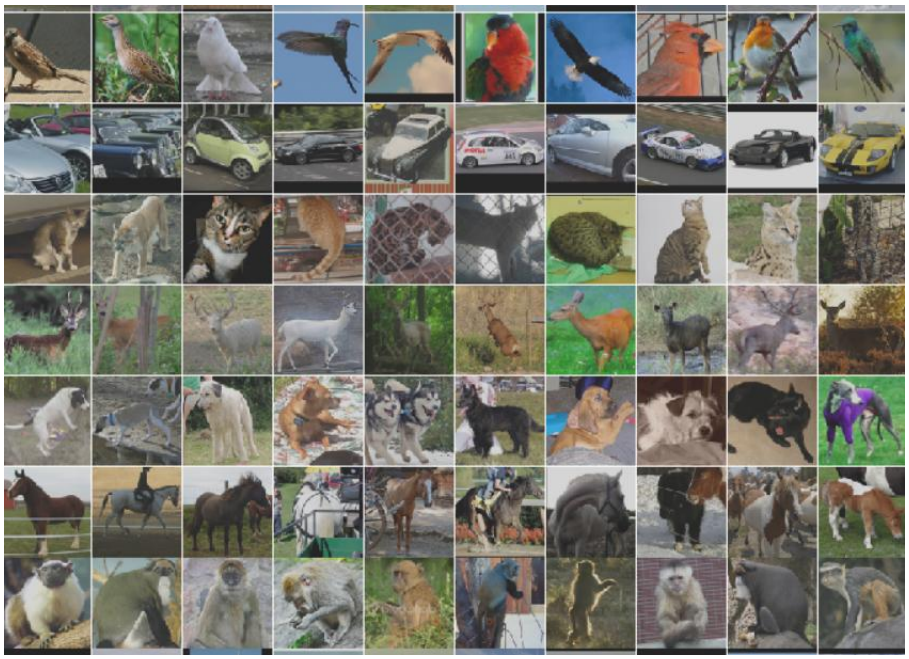
强度。

2.风格随机化的图形数据增强算法的实验:

- **实验环境和数据集描述：**
  - **实验环境**：Python 3.10　PyTorch
  - **数据集概述**：使用 **STL-10** 数据集来进行分类任务。

**STL-10** 的图像来自 **ImageNet**，由十个类组成（类别包括飞机、鸟、车等），共有 113000 张 96 x 96 分辨率的 RGB 图像，其中训练集为 5000 张，测试集为 8000 张，其余 100000 张均为无标签图像。

| id | name |
|---|---|
| 1 | airplane |
| 2 | bird |
| 3 | car |
| 4 | cat |
| 5 | deer |
| 6 | dog |
| 7 | horse |
| 8 | monkey |
| 9 | ship |
| 10 | truck |



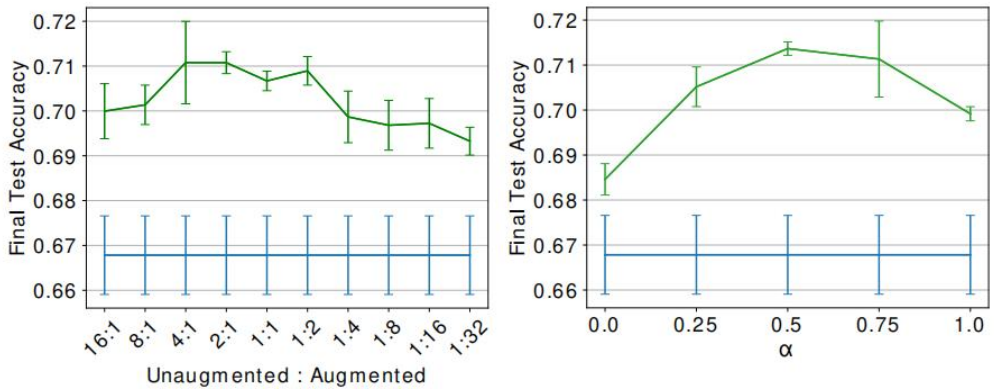
**数据集问题：**

- STL-10 数据集中的**图像分辨率较低**。这可能会导致模型在捕捉图像细节方面存在困难。
- 数据集中存在部分**噪声数据**

- **实验方法：**

1， 确定未增广图片与增广图片的比例和样式转移调度α的最优值。

对于未增广和增广图片的比例，我们从 16：1（未增广：增广）到 1： 32 的两倍系数差值进行实验。由于我们不知道α的最优值，我们在这些实验中从区间[0,1]中均匀随机抽样。
下图左展示了此搜索的结果。
对于α的最优值，我们用不同的随机种子进行测试，最终确定 0.5 是最优值。

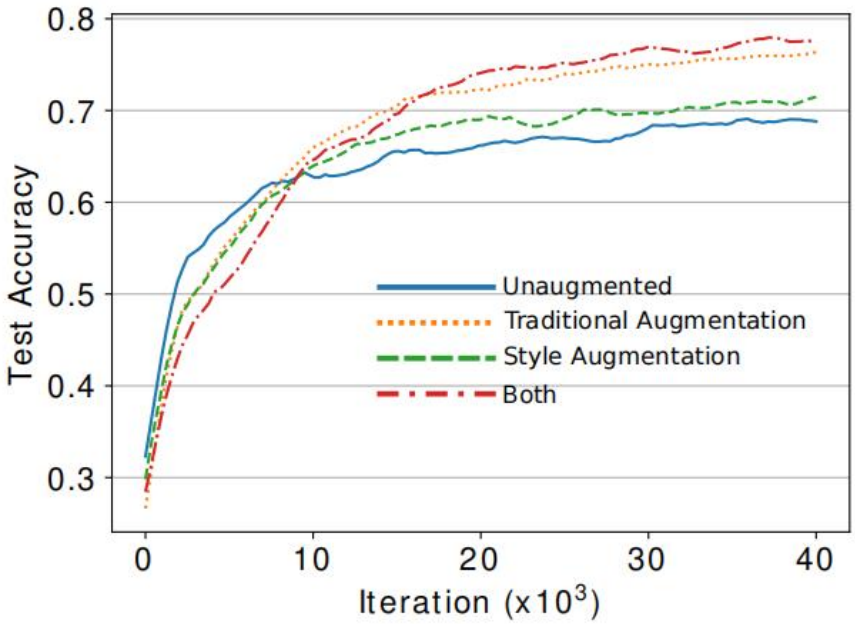



2， 对单张图片进行风格转换

我们对 mug.png 图片进行风格转换，以此来验证风格随机化网络的适用性和正确性。以下为两次执行风格随机化数据增强算法代码得到的结果，两次得到的结果是不一样的风格图片。

3， 在 **STL-10** 数据集上进行分类实验以评估该算法的实际效果。

为了验证该风格随机化数据增强算法是否真正增强了数据集的质量并使实际任务的准确率得到提升，我们用 STL-10 数据集进行分类任务，并对比了经过数据增强的数据集、经过传统数据增强的数据集（如进行旋转，剪切等操作）、经过随机化风格增强的数据集和传统数据增强和随机化风格增强共同作用的数据集的结果

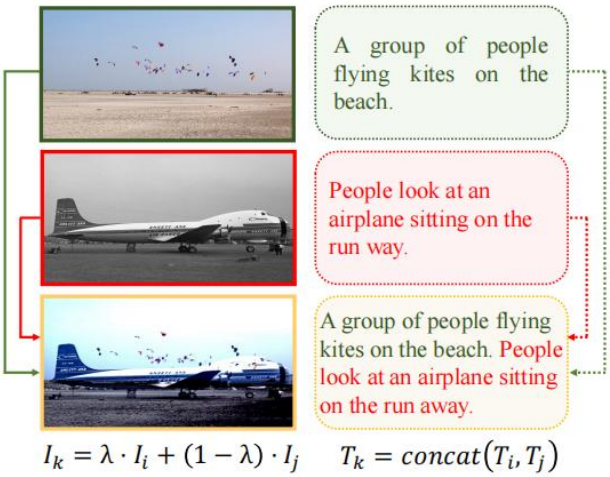



通过实验结果可知，数据增强算法本身会导致更快的收敛性和更好的准确性，但最终的提升上不如传统的增强方法。而将传统的增强方法和随机样式增强共同作用于数据集，则会得到更好的效果。

## 四：图文联合数据增强算法研究

1.图文联合数据增强算法:

- 原理简述：




图文联合数据增强算法是一种针对视觉-语言联合表示学习的多模态数据增强方法。它通过线性插值图像和连接文本序列来生成具有语义关系的新的图像-文本对。核心在于保持新生成的图像-文本对中的语义关系大多数情况下仍然匹配。

这种算法的最大特点是**简单易用**，可以集成到现有的处理流程中。

本项目通过在四种现有的流行视觉-语言（Vision-Language）预训练模型架构（包括 CLIP、ViLT、ALBEF 和 TCL）上进行评估，并且在五个经典的下游视觉-语言任务上展示其通用性和有效性。

例如，在 ALBEF 预训练中添加 MixGen，可以在下游任务上带来显著的性能提升，如在 COCO 的图像-文本检索任务上微调后提高了 6.2%，在 Flicker30K 的零样本任务上提高了 5.3%。

- Baseline 问题解读：

我们的算法基于上面提到的四种现有的视觉语言预训练模型架构。但是它们普遍存在一些问题。

1.**大规模数据集的需求**：这些模型通常需要利用大量的图像-文本配对数据来进行训练。例如，CLIP 模型使用了 4 亿个图像-文本对，以达到 ResNet-50 在 ImageNet 上的准确度，但这种方法对数据的需求极高。

2.**高昂的计算资源成本**：为了训练这些模型，需要使用成千上万的 GPU 资源。CLIP 模型的训练就花费了 12 天时间在 256 个 V100 GPU 上进行。这表明了在计算资源方面的巨大需求。

3.**数据集的可获取性问题**：大规模的数据集往往不公开，限制了研究的可复现性和进一步的改进。即使这些数据集可用，对于计算资源有限的研究人员来说，复现和改进现有方法仍然是一个挑战。

而且最重要的是，在图像-文本对中，图像和文本都包含彼此匹配的丰富信息。直观地说，我们希望他们的语义在数据增强后仍然匹配。为了保持语义关系，这些模型对视觉或文本模态只进行了温和的数据增强,存在或多或少的问题，不能尽如人意。

例如，ViLT 和后续的工作采用了 RandAugment 进行图像增强，但没有进行颜色反转。CLIP 和 ALIGN 仅使用随机大小裁剪而不采用其他图像增强手段。这意味着视觉数据的增强是相当

有限的，可能没有充分利用数据增强在提高模型泛化能力方面的潜力。

- **技术细节：**

**1.多模态表示：**首先，为每个图像及其相关文本构建联合表示。这通常涉及使用卷积神经网络处理图像和使用自然语言处理模型（如 BERT）处理文本。

**2.样本生成：**

基于线性插值和文本拼接的原则来生成新的图像-文本对。假设我们有一个包含 N 个图像-文本对的数据集，其中图像和文本分别以 I 和 T 表示，带有下标。给定任意两个图像-文本对 $(Ii,Ti)$和$(Ij,Tj)$，其中 $i,j∈\{1,…,N\}$且 $i≠j$，通过以下方式生成新的训练样本$(Ik,Tk)$：

$$Ik=λ.Ii+(1−λ).Ij$$

$$Tk=concat(Ti,Tj)$$

在这里，$λ$是一个介于 0 和 1 之间的超参数，用于指示两幅图像 $Ii$和 $Ij$的原始像素之间的线性插值；而 $concat$操作符直接连接两个文本序列 $Ti$和 $Tj$，以最大限度地保留原始信息。通过这种方式，新生成的图像-文本对$(Ik,Tk)$在大多数情况下仍能保持语义关系.这种随机组合图像-文本样本的方法也增加了模型训练的多样性，有助于模型学习到罕见概念。

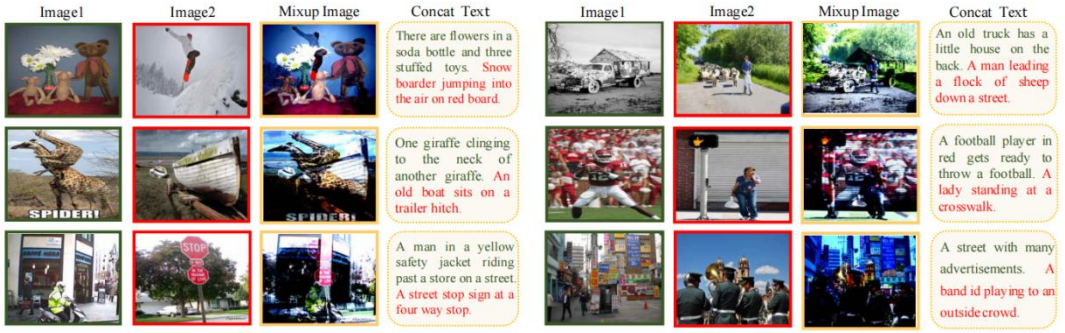



Figure 2. **More image-text pairs generated by MixGen**. The new pairs not only preserve original semantic relationships, but also increase diversity to model training. Figure best viewed in color.

**3.伪代码：**

**Algorithm 1** Pseudocode of MixGen data augmentation

```
# image, text:  a batch of B randomly sampled {image, text}
# M: number of new image-text pairs generated by MixGen
# λ:  image mixup ratio

def mixgen(image, text, M, λ):
    for i in range(M):
      # image mixup
      image[i,:]  = (λ * image[i,:]  + (1-λ) * image[i+M,:])
      # text concatenation
      text[i] = text[i] + " " + text[i+M]
    return image, text
```

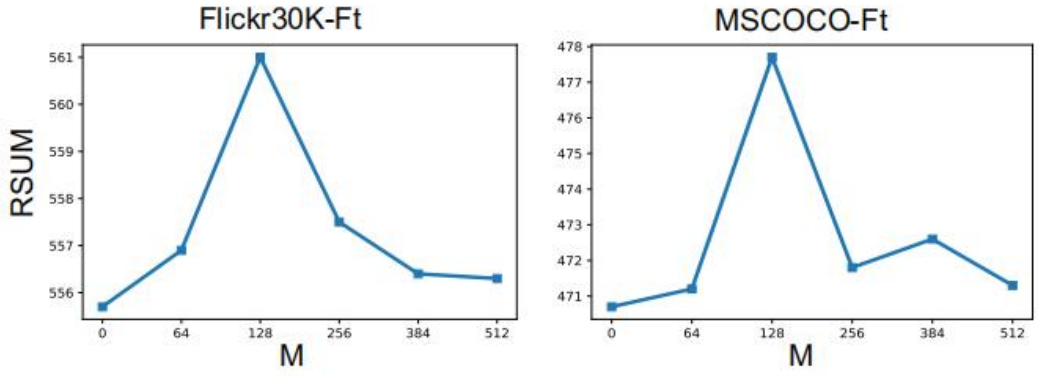
给定一个随机采样的图像-文本对，该算法用新生成的 M 个图像-文本对来替换前 M 个训练样本。因此，批处理大小 B、总训练迭代次数和总体训练管道保持不变。默认情况下，我们在算法 1 中设置了λ = 0.5 和 M = B/4.

这种即插即用的技术可以很容易地集成到大多数视觉语言表示学习方法和任务中：它只需要几行代码，且只有很小的计算开销。

- **消融研究：**

**算法参数优化：**

我们研究了在使用 MixGen 生成的样本中，最佳的样本比例。在伪代码中，我们用 MixGen 新生成的对替换了第一个 M 个训练样本，以便总批量大小和训练迭代次数保持不变。我们尝试随机打乱现有对并完全用新样本替换它们，即 M = B。在这里，我们进行了一项消融研究，以探索最优的新对数量。




如图 6 所示，在固定批量大小 B = 512 的情况下变化 M 时，M = 128（即 M = B/4）带来了最佳性能。

**确定算法的最优变体(Variant)**

该算法本身是一个简单的算法，它通过混合（mixup）图像和拼接（concatenation）文本来生成保持语义关系的新的图像-文本对。然而，除了这种基础的方法外，理论上可以通过采用不同的参数取值方法来创建多种 MixGen 的变体。这些变体可能比基础方法有更好的效果。

变体（a）是在默认的 MixGen 基础上引入了一个变化，即λ值从 Beta 分布中采样（λ ~ Beta(0.1, 0.1)），这一点遵循了原始 mixup 方法的做法。

变体 (b)：混合（mixup）两幅图像，并从两段文本中均匀地选择一段文本序列。

变体 (c)：拼接（concatenate）两段文本序列，并从两幅图像中均匀地选择一幅图像。

变体 (d)：基于λ值（与图像 mixup 相似）从两段文本序列中按比例选择 tokens（词元或文本片段），然后将这些 tokens 拼接起来。

变体 (e)：首先将所有 tokens 拼接起来，然后随机保留一半，以生成新的文本序列。

Table 6. **MixGen variants.** For (d) and (e), $|T|_\lambda$ indicates randomly keeping $\lambda \cdot |T|$ tokens in text sequence $T$, where $|T|$ is the number of tokens in this sequence. See text for more details.

| Variants | Image | Text | $\lambda$ |
|---|---|---|---|
| MixGen | $I_k = \lambda \cdot I_i + (1-\lambda) \cdot I_j$ | $T_k = \text{concat}(T_i, T_j)$ | $\lambda = 0.5$ |
| (a) | $I_k = \lambda \cdot I_i + (1-\lambda) \cdot I_j$ | $T_k = \text{concat}(T_i, T_j)$ | $\lambda \sim \text{Beta}(0.1, 0.1)$ |
| (b) | $I_k = \lambda \cdot I_i + (1-\lambda) \cdot I_j$ | $T_k = T_i$ or $T_j$ | $\lambda = 0.5$ |
| (c) | $I_k = I_i$ or $I_j$ | $T_k = \text{concat}(T_i, T_j)$ | - |
| (d) | $I_k = \lambda \cdot I_i + (1-\lambda) \cdot I_j$ | $T_k = \text{concat}(\lvert T_i\rvert_\lambda, \lvert T_j\rvert_{1-\lambda})$ | $\lambda \sim \text{Beta}(0.1, 0.1)$ |
| (e) | $I_k = \lambda \cdot I_i + (1-\lambda) \cdot I_j$ | $T_k = \lvert\text{concat}(T_i, T_j)\rvert_{0.5}$ | $\lambda \sim \text{Beta}(0.1, 0.1)$ |

Table 7. **Ablation of MixGen design**. For image-text retrieval task, we report the RSUM metric. Ft: fine-tuned setting. Zs: zero-shot setting.

| MixGen variants | Flickr30K-Ft (1K test set) | MSCOCO-Ft (5K test set) | Flickr30K-Zs (1K test set) | SNLI-VE (test) | NLVR$^2$ (test-P) | VQA (test-dev) |
|---|---|---|---|---|---|---|
| ALBEF-base | 555.7 | 470.7 | 518.0 | 78.91 | 78.09 | 73.62 |
| MixGen | **561.0** | **477.7** | **524.3** | **79.65** | **79.42** | 73.84 |
| (a) | 553.2 | 467.4 | 512.9 | 78.68 | 77.22 | 73.15 |
| (b) | 557.2 | 470.0 | 516.2 | 78.78 | 78.23 | 73.76 |
| (c) | 555.4 | 472.3 | 518.2 | 78.87 | 78.60 | 73.34 |
| (d) | 555.2 | 463.7 | 506.8 | 79.07 | 78.30 | 73.29 |
| (e) | 559.2 | 477.1 | 523.7 | 79.36 | 78.33 | **73.95** |

各个变体的对于图片和文本的具体处理方法如表 6 所示。正如我们将在表 7 中看到的，我们的默认 MixGen 实现了整体上的最佳性能，并且在四种不同的视觉语言下游任务中始终优于其他变体。

## 2.实验

**在实验部分中，我们首先介绍基础模型的预训练过程。**

**随后，我们将进行图像-文本检索任务，并展示应用 MixGen 后的实验结果。**

**最后，我们将通过可视化展示使用 MixGen 的效果。**

- **模型预训练：**
- **预训练方法：**

我们采用了插入式（plug-and-play）的方式，在四种流行的不同架构方法上应用 MixGen，包

括 CLIP（双编码器）、ViLT（单一融合编码器）、ALBEF（双编码器后跟融合编码器）和 TCL（双编码器后跟融合编码器）。虽然这些方法的预训练目标各不相，但仍然 MixGen 可以轻松地整合到这些架构中，仅需几行代码即可实现。具体而言，一旦我们从数据加载器中获取了一个小批量数据，我们就应用 MixGen 生成新的图像-文本对并更新这个小批量数据。然后，我们将更新后的批次输送给网络，而不修改它们原有的训练设置。

**- 预训练数据集：**

如今，有 4 个被广泛采用的视觉-语言模型预训练的数据集，包括 COCO、Visual Genome (VG)、SBU Captions 和 Conceptual Captions (CC)。大多数相关文献使用这四个数据集的组合作为标准预训练设置，这导致了总共 4M 个独特图像和 5.1M 个图像-文本对。然而，对于 SBU 和 CC 数据集，图像-文本对是以 URL 格式提供的，部分 URL 现在已经无法访问。例如，我们只能从 CC 数据集的 3M 图像中下载到 2.2M 图像。因此，我们的最终训练集只有 3.3M 个独特图像和 4.4M 个图像-文本对。与之前能够访问原始 5.1M 图像文本对的文献相比，我们的训练样本少了大约 700K。

**- 实施细节：**

我们的大部分实验都是基于 ALBEF 进行的，因为它在一系列下游任务上具有优越的性能。

**- 下游任务 1 图像文本检索(Image-text retrieval)：**

图像-文本检索包括两个子任务：

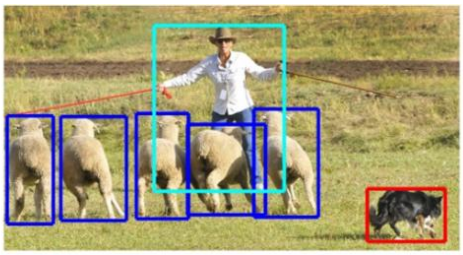
(1)检索具有给定文本的图像（图像检索）

(2)检索具有给定图像的文本（文本检索）

**实验数据集：**

MSCOCO datasets


(a) Image classification



(b) Object localization

MSCOCO 数据集是一个非常大型且实用的数据集，其中包括了物体检测、关键点检测、实例分割、全景分割、图片标注五种类型的标注。在我们的实验中只用到了图片标注。

MSCOCO 数据集总共有 91 个类别，每个类别的图片数量不相等。每张图片对应 5 段文本描述。

**MSCOCO 数据集问题：**

**- 数据集不完整：**对 MSCOCO 数据集，图像-文本对是以 URL 格式提供的，部分 URL 现在已经无法访问。我们只能从 MSCOCO 数据集的 3M 图像中下载到 2.2M 图像。

**- 实例目标个数多：**平均每张图片实例目标个数为 7.7 个。而且小目标多，特征提取困难。

**Flickr30K datasets**

Flickr30K 数据集是一个广泛应用于图像标注和自然语言处理任务的数据集。该数据集由 Flickr 在线图片共享平台的 30313 图片组成，每张图片都标注了 5 段文本描述。这些文本中描述了图片的主题、场景、动作等内容。

| | image | caption |
|---|---|---|
| 0 | 1000092795.jpg#0 | Two young guys with shaggy hair look at their ... |
| 1 | 1000092795.jpg#1 | Two young , White males are outside near many ... |
| 2 | 1000092795.jpg#2 | Two men in green shirts are standing in a yard . |
| 3 | 1000092795.jpg#3 | A man in a blue shirt standing in a garden . |
| 4 | 1000092795.jpg#4 | Two friends enjoy time spent together . |
| 5 | 10002456.jpg#0 | Several men in hard hats are operating a giant... |
| 6 | 10002456.jpg#1 | Workers look down from up above on a piece of ... |
| 7 | 10002456.jpg#2 | Two men working on a machine wearing hard hats . |
| 8 | 10002456.jpg#3 | Four men on top of a tall structure . |
| 9 | 10002456.jpg#4 | Three men on a large rig . |
| 10 | 1000268201.jpg#0 | A child in a pink dress is climbing up a set o... |
| 11 | 1000268201.jpg#1 | A little girl in a pink dress going into a woo... |
| 12 | 1000268201.jpg#2 | A little girl climbing the stairs to her playh... |
| 13 | 1000268201.jpg#3 | A little girl climbing into a wooden playhouse . |
| 14 | 1000268201.jpg#4 | A girl going into a wooden building . |
| 15 | 1000344755.jpg#0 | Someone in a blue shirt and hat is standing on... |
| 16 | 1000344755.jpg#1 | A man in a blue shirt is standing on a ladder ... |
| 17 | 1000344755.jpg#2 | A man on a ladder cleans the window of a tall ... |
| 18 | 1000344755.jpg#3 | man in blue shirt and jeans on ladder cleaning... |

Flickr30K 数据集问题：

- **数据集小**
- 文本描述中**无用词语较多**，且不是所有图片中出现的目标都被提及。

**评估指标：**

为了便于比较，我们使用 RSUM 作为度量来揭示模型的总体性能，RSUM 定义为图像和文本检索任务的召回度量的总和。

**实验结果:**

Table 1. **Fine-tuned image-text retrieval** on Flickr30K and MSCOCO datasets.

| Method | #Images | MSCOCO(5K test set) | | | | | | | Flickr30K(1K test set) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Text Retrieval | | | Image Retrieval | | | | Text Retrieval | | | Image Retrieval | | | |
| | | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 | RSUM | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 | RSUM |
| UNITER-base [3] | 4M | 64.4 | 87.4 | 93.1 | 50.3 | 78.5 | 87.2 | 460.9 | 85.9 | 97.1 | 98.8 | 72.5 | 92.4 | 96.1 | 542.8 |
| VILLA-base [10] | 4M | - | - | - | - | - | - | - | 86.6 | 97.9 | 99.2 | 74.7 | 92.9 | 95.8 | 547.1 |
| OSCAR-base [27] | 4M | 70.0 | 91.1 | 95.5 | 54.0 | 80.8 | 88.5 | 479.9 | - | - | - | - | - | - | - |
| UNIMO-base [26] | 4M | - | - | - | - | - | - | - | 89.7 | 98.4 | 99.1 | 74.7 | 93.4 | 96.1 | 551.4 |
| ViLT-base [19] | 4M | 61.5 | 86.3 | 92.7 | 42.7 | 72.9 | 83.1 | 439.2 | 83.5 | 96.7 | 98.6 | 64.4 | 88.7 | 93.8 | 525.7 |
| ALBEF-base [23] | 4M | 73.1 | 91.4 | 96.0 | 56.8 | 81.5 | 89.2 | 488.0 | 94.3 | 99.4 | 99.8 | 82.8 | 96.7 | 98.4 | 571.4 |
| ALBEF-base [23] | 3M | 72.5 | 91.7 | 95.9 | 55.8 | 81.3 | 88.4 | 485.6 | **95.1** | 99.1 | 99.7 | 81.4 | 96.0 | **98.2** | 569.5 |
| ALBEF-base+**MixGen** | 3M | **74.2** | **92.8** | **96.4** | **57.3** | **82.1** | **89.0** | **491.8**$_{+6.2}$ | 94.8 | **99.4** | **100.0** | **82.4** | **96.3** | 98.0 | **570.9**$_{+1.4}$ |

在表 1 中，我们可以看到 MixGen 在两个数据集上一致地改善了 ALBEF 的基线性能。在 3M 设置下，仅仅添加 MixGen 而不做任何修改，就在 COCO 数据集上带来了 6.2%的 RSUM 得分提升，在 Flicker30K 数据集上带来了 1.4%的 RSUM 得分提升。

**需要注意的是**，由于数据集的缺失问题，我们重现的 ALBEF 模型（在 3.3M 对图像-文本对上训练）的性能略低于原始论文中报告的性能（在 4M 对图像-文本对上训练，表 1 中的灰色行）。然而，在添加 MixGen 之后，尽管我们的模型使用的是少了 700K 对图像-文本对的训练集，其性能在 COCO 数据集上甚至超过了原始论文的结果，在 Flicker30K 数据集上也具有竞争力。这清楚地表明 MixGen 提高了模型训练的数据效率。

- **模型兼容性**

在图像文本检索这一实验中我们同时检验了 MixGen 与其他视觉-语言（VL）预训练方法（如 CLIP、ViLT 和 TCL）的兼容性。
在集成 MixGen 的过程中，除了添加 MixGen 外，没有修改这些模型的原始训练设置。
考虑到 ViLT 训练的成本非常高（例如，需要 64 个 V100 GPU 训练 3 天），因此在这次实验中，仅使用了 COCO、VG 和 SBU 三个数据集，而不是四个，在预训练期间我们使用的数据集由 1M 个独特图像和 2.2M 个图像-文本对组成。

Table 3. **Compatibility to other vision-language pre-training methods**. Adding MixGen leads to consistent performance boost in terms of image-text retrieval. Note the models here are pre-trained on three datasets (COCO, VG and SBU) with 1M images. Ft: fine-tuned setting. Zs: zero-shot setting.

| Methods | Venue | Flickr30K-Ft (1K test set) | MSCOCO-Ft (5K test set) | Flickr30K-Zs (1K test set) |
|---|---|---|---|---|
| ViLT [19] | ICML 21 | 232.2 | 172.5 | 135.5 |
| ViLT + **MixGen** | - | **241.4**$_{+9.20}$ | **189.7**$_{+17.2}$ | **150.4**$_{+14.9}$ |
| CLIP [38] | ICML 21 | 538.2 | 450.5 | 485.3 |
| CLIP + **MixGen** | - | **541.0**$_{+2.80}$ | **454.6**$_{+4.10}$ | **492.1**$_{+6.80}$ |
| ALBEF [23] | NeurIPS 21 | 555.7 | 470.7 | 518.0 |
| ALBEF + **MixGen** | - | **561.0**$_{+5.30}$ | **477.7**$_{+7.00}$ | **524.3**$_{+6.30}$ |
| TCL [60] | CVPR 22 | 561.5 | 487.0 | 537.3 |
| TCL + **MixGen** | - | **563.6**$_{+2.10}$ | **490.2**$_{+3.20}$ | **539.5**$_{+2.20}$ |

如表 3 所示，仅在这些 Baseline 模型上添加 MixGen，就一致地提高了性能。
在 COCO 上进行的图像-文本检索任务中，MixGen 展示了显著的准确率提升：ViLT（+17.2%）、CLIP（+4.1%）、ALBEF（+7.0%）和 TCL（+3.2%）。
这证明了 MixGen 作为图像-文本数据增强在预训练中的多功能性。

- **数据效率提升**

随后我们研究了 MixGen 对数据效率的提升。为了评估这一点，研究减少了用于预训练的独特图像数量，从 3M 减少到 2M、1M 和 200K。结果可以在图 3 中看到。

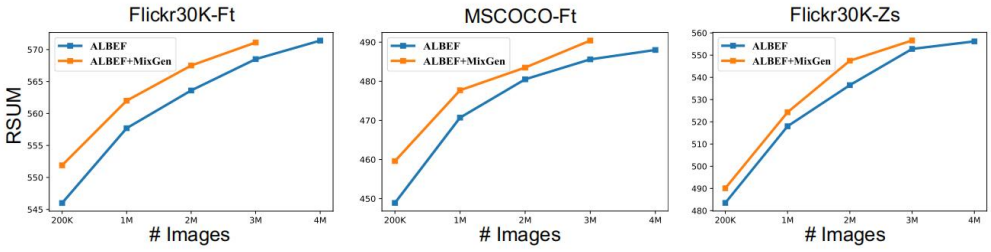



Figure 3. **Image-text retrieval performance given various number of training images**. Note that ALBEF with MixGen trained on 3M images achieves competitive performance or outperform baseline ALBEF trained on 4M images. This indicates the data efficiency of MixGen as an effective data augmentation method.

首先注意到，使用 MixGen 总是比不使用它更好，特别是在**低数据量**的情况下，改进更为显著。
其次，当在 1M、2M 和 3M 样本上训练的 ALBEF 模型加入 MixGen 后，其性能分别与在 2M、3M 和 4M 样本上训练的基线 ALBEF 模型匹配。这再次表明了 MixGen 对数据效率的提升。

- 其他四个下游任务

MixGen 在四个下游任务——视觉问答（VQA）、视觉推理（VR）、视觉蕴含（VE）和视觉定位（VG）上的评估结果。

- **视觉问答（VQA）**：目标是根据给定的图像和相应的问题预测一个答案。实验在 VQA 2.0 数据集上进行，将其视为一个答案生成任务，并限制微调后的答案解码器从 3,192 个候选答案中生成答案，使用官方评估服务器报告准确率。

- **视觉蕴含（VE）**：是一个视觉推理任务，预测图像和文本之间的关系是蕴含、中立还是矛盾。将此任务视为三分类问题，在 SNLI-VE 数据集上报告分类准确率。

- **视觉推理（VR）**：要求模型判断文本陈述是否描述了一对图像。在 NLVR2 数据集上进行实验，并报告标准的每个样本预测准确率。

- **视觉定位（VG）**：定位与文本描述相对应的图像区域。在弱监督设置下，对 RefCOCO+ 数据集进行评估。

Table 4. **Comparison with state-of-the-art methods on downstream vision-language tasks**. MixGen consistently improve across VQA, VR and VE.

| Method | #Images | VQA | | NLVR$^2$ | | SNLI-VE | |
|---|---|---|---|---|---|---|---|
| | | test-dev | test-std | dev | test-P | val | test |
| VisualBERT-base [24] | 4M | 70.80 | 71.00 | 67.40 | 67.00 | - | - |
| OSCAR-base [27] | 4M | 73.16 | 73.44 | 78.07 | 78.36 | - | - |
| UNITER-base [3] | 4M | 72.70 | 72.91 | 77.18 | 77.85 | 78.59 | 78.28 |
| ViLT-base [19] | 4M | 71.26 | - | 75.70 | 76.13 | - | - |
| UNIMO-base [26] | 4M | 73.79 | 74.02 | - | - | 80.00 | 79.10 |
| VILLA-base [10] | 4M | 73.59 | 73.67 | 78.39 | 79.30 | 79.47 | 79.03 |
| ALBEF-base [23] | 4M | 74.54 | 74.70 | 80.24 | 80.50 | 80.14 | 80.30 |
| ALBEF-base [23] | 3M | 74.38 | 74.51 | 79.47 | 80.05 | 79.49 | 79.69 |
| ALBEF-base+**MixGen** | 3M | **74.51**$_{+0.13}$ | **74.79**$_{+0.28}$ | **80.23**$_{+0.76}$ | **80.94**$_{+0.89}$ | **80.05**$_{+0.56}$ | **80.05**$_{+0.36}$ |

性能比较显示，与其他视觉-语言预训练基线相比，MixGen 在 VQA、VR 和 VE 任务上一致地提升了性能。在 3M 设置下，使用 MixGen 的 ALBEF 在 VQA test-std、NLVR2 test-P 和 SNLI-VE test 上分别相对于其对应基线提升了 0.28%、0.89%和 0.36%的绝对准确率。值得注意的是，在 3M 设置下，使用 MixGen 的 ALBEF 甚至在 VQA 和 NLVR2 上超过了使用 4M 图像训练的原始 ALBEF。

Table 5. **Weakly-supervised visual grounding** on RefCOCO+

| Method | #Images | Val | TestA | TestB |
|---|---|---|---|---|
| ARN [29] | 14M | 32.78 | 34.35 | 32.13 |
| CCL [67] | 14M | 34.29 | 36.91 | 33.56 |
| ALBEF$_{itc}$ [23] | 14M | 51.58 | 60.09 | 40.19 |
| ALBEF$_{itm}$ [23] | 14M | 58.46 | 65.89 | 46.25 |
| ALBEF$_{itm}$ [23] | 3M | 57.76 | 65.08 | 45.57 |
| ALBEF$_{itm}$+**MixGen** | 3M | 56.72 | **65.35**$_{+0.27}$ | **46.47**$_{+0.9}$ |

在视觉定位（VG）任务上，特别是在 RefCOCO+数据集上的性能比较表明，使用 MixGen 的模型不仅超过了其对应基线在两个测试集上的性能，而且还超过了在 14M 数据集上训练的 ALBEFitm 在 TestB 上的表现。所有这些经验结果均表明 MixGen 带来的卓越数据效率。

- 实验结果可视化


Figure 5. **Text-to-image retrieval on MSCOCO**. Given a text query, $A \leftarrow B$ indicates the rank of the retrieved ground-truth image improves from $B$ (wo MixGen) to $A$ (w MixGen).

图像-文本检索：在 MSCOCO 数据集上进行的文本到图像检索任务中，通过比较使用 MixGen 和未使用 MixGen 的 ALBEF 模型，我们可以看到 MixGen 通常能够在检索中提高最匹配图像的排序，表现显著优于基线 ALBEF 模型。

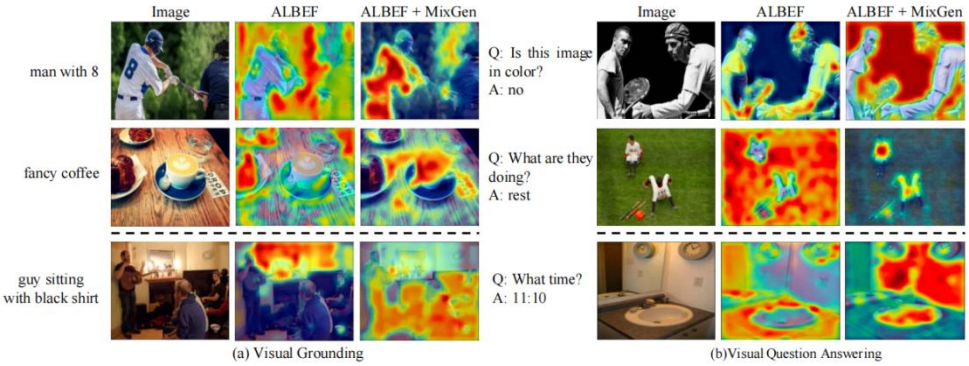



Figure 4. **Grad-CAM visualization for visual grounding and VQA**. First two rows are successful cases where MixGen helps, and the third row shows failure cases. Figure best viewed in color.

**视觉定位和视觉问答：**通过 Grad-CAM 视觉化技术，我们可以更好地理解 MixGen 为何有益。在 RefCOCO+数据集上进行的**视觉定位任务(VG)**中，使用 MixGen 训练的模型能够更精确地根据文本定位到图像区域。即使在失败案例中，使用 MixGen 训练的模型也能够更好地关注到“坐着的人”，而不是集中在墙上（未使用 MixGen 的模型）。
在 VQA 2.0 数据集上进行的 VQA 任务中，MixGen 能够关注到导致正确答案的重要线索（例如，为了预测这是不是一个黑白图像而关注黑色背景）。对于失败案例，判断时间可能过于具有挑战性，其中 ALBEF 和 ALBEF + MixGen 均未成功。

总的来说，MixGen 不仅在定位匹配图像和更精确地识别图像区域方面有所帮助，而且在关注解决 VQA 任务时的重要线索方面也显示出其优势。

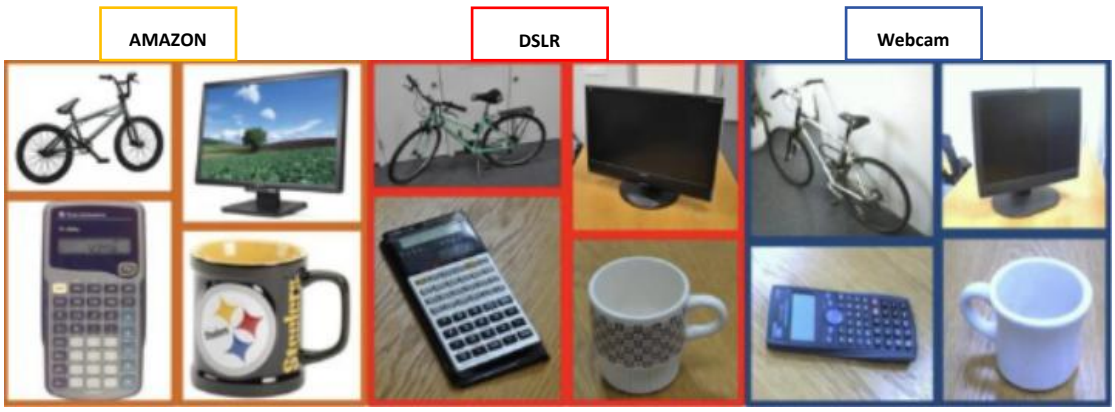
## 五：问题解答

## 问题 1：

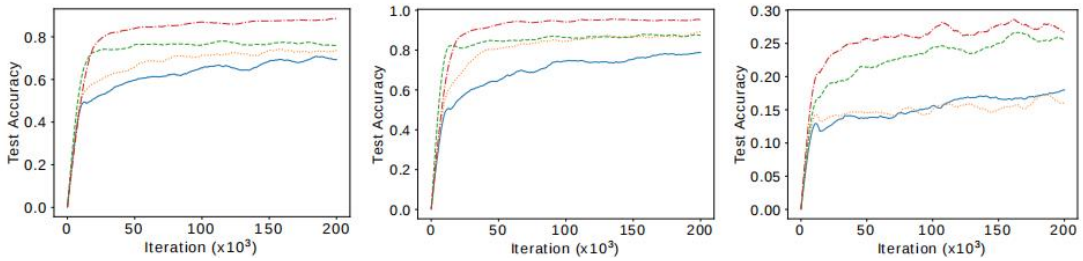
**验证第一个算法——风格随机化数据增强算法增强了模型的鲁棒性和稳定性。**

**答：**我们将该算法应用到了 Office-31 跨领域分类数据集的分类任务中。

首先介绍一下 **Office-31 数据集**，该数据集由 31 个类组成，分为三个领域：亚马逊（A）、DSLR（D） 和网络摄像头（W）。亚马逊域名包含 2817 张从亚马逊产品列表中抓取的图片，而 DSLR 和网络摄像头分别包含 498 张和 795 张图片，分别是在办公室环境中用 DSLR 相机和网络摄像头拍摄的。




我们通过在两个领域上运用 **InceptionV3 卷积架构**联合训练分类模型，并在另一个领域来进行测试以得到风格增强后的分类效果。




上图为我们得到的结果，我们依旧是对比了运用四种不同增强方法的数据集进行分类任务的效果，其中第一张图片是 AD—>W 的效果，第二张是 AW->D,第三张是 DW->A。我们可以很好的看出无论是哪种情况下，风格随机化的数据增强算法都提升了分类任务的准确性。而将风格随机化图形数据增强方法与传统数据增强方法联合增强的数据集准确性更高。

在对比结果的过程中，我们同时注意到第三张图片中分类的准确性十分低。这是由于亚马逊域（A）图像和其他两个域的图片之间的差异较大，使得分类任务十分困难。但即使如此，我们的算法依然提升了该分类任务的准确性。

## 问题 2：

**在 Flickr30K 和 MSCOCO 数据集上的图像-文本检索结果:**

Table 1. **Fine-tuned image-text retrieval** on Flickr30K and MSCOCO datasets.

| Method | #Images | MSCOCO(5K test set) | | | | | | | Flickr30K(1K test set) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Text Retrieval | | | Image Retrieval | | | | Text Retrieval | | | Image Retrieval | | | |
| | | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 | RSUM | R@1 | R@5 | R@10 | R@1 | R@5 | R@10 | RSUM |
| UNITER-base [3] | 4M | 64.4 | 87.4 | 93.1 | 50.3 | 78.5 | 87.2 | 460.9 | 85.9 | 97.1 | 98.8 | 72.5 | 92.4 | 96.1 | 542.8 |
| VILLA-base [10] | 4M | - | - | - | - | - | - | - | 86.6 | 97.9 | 99.2 | 74.7 | 92.9 | 95.8 | 547.1 |
| OSCAR-base [27] | 4M | 70.0 | 91.1 | 95.5 | 54.0 | 80.8 | 88.5 | 479.9 | - | - | - | - | - | - | - |
| UNIMO-base [26] | 4M | - | - | - | - | - | - | - | 89.7 | 98.4 | 99.1 | 74.7 | 93.4 | 96.1 | 551.4 |
| ViLT-base [19] | 4M | 61.5 | 86.3 | 92.7 | 42.7 | 72.9 | 83.1 | 439.2 | 83.5 | 96.7 | 98.6 | 64.4 | 88.7 | 93.8 | 525.7 |
| ALBEF-base [23] | 4M | 73.1 | 91.4 | 96.0 | 56.8 | 81.5 | 89.2 | 488.0 | 94.3 | 99.4 | 99.8 | 82.8 | 96.7 | 98.4 | 571.4 |
| ALBEF-base [23] | 3M | 72.5 | 91.7 | 95.9 | 55.8 | 81.3 | 88.4 | 485.6 | **95.1** | 99.1 | 99.7 | 81.4 | 96.0 | **98.2** | 569.5 |
| ALBEF-base+**MixGen** | 3M | **74.2** | **92.8** | **96.4** | **57.3** | **82.1** | **89.0** | **491.8**$_{+6.2}$ | 94.8 | **99.4** | **100.0** | **82.4** | **96.3** | 98.0 | **570.9**$_{+1.4}$ |

**在 MSCOCO(5K test set)数据集上，进行的 Image Retrieval 中，为什么 R@1 这一列的结果比同一数据集的不同任务或者不同数据集的任务都要显著低？**

**答：**

在**图像文本检索任务**中，**"R@K"**代表**"Recall at K"**,**R@1, R@5, R@10** 指的都是召回率指标，其中 K 的值分别为 1、5、10。

具体来说：

- **R@1** 表示给定一个查询（无论是文本查询图像还是图像查询文本），检索系统返回的第一个结果中包含正确答案的概率。
- **R@5** 表示在前五个返回的结果中包含正确答案的概率。
- **R@10** 则是前十个结果中包含正确答案的概率。

MSCOCO 数据集中 Image Retrieval 中 R@1 结果相对较低的问题，这是因为 R@1 是最严格的评估指标，它要求模型的第一个返回结果就是正确的，这自然比在前 5 个或前 10 个结果中找到正确答案要难得多。因此，通常 **R@1 的分数会比 R@5 和 R@10 的分数低**，这是因为随着评估标准的放宽（考虑更多的顶部返回结果），找到正确答案的概率增加了。

在实际应用中，一个较低的 R@1 得分意味着在第一次尝试时模型找到正确答案的能力较弱，但随着更多选项的提供，它找到正确答案的机会增加。这在信息检索和推荐系统中是常见的现象，因为通常不太可能在第一次就完美命中，但随着提供更多的选择，系统找到用户需要的内容的可能性增加。

1. **为什么 MSCOCO 数据集下的 Image Retrieval 的 R@1 结果相比于 Text Retrieval 的 R@1 结果低？**

- **数据集规模和复杂性：**MSCOCO 是一个**大规模且复杂**的数据集，含有多样化的图像和丰富的标注。**图像检索**需要从大量的图像中找到与文本描述精确匹配的一张图，难度往往高于**文本检索**，因为**图像的细节和场景**可能更加复杂多变。这也解释了为什么在两种数据集下 Image Retrieval 的 R@1 结果相比于 Text Retrieval 的 R@1 结果都更低。
- **模态的差异：** **文本**能够以非常具体和抽象的方式描述**图像内容**，而图像本身的信息是直观和分布式的。**从图像中检索文本相比于从文本中检索图像，在语义表征上可能更容易，因为人类在描述图像时往往会提到图像中最显著的元素（比如“一只狗在草地上玩耍”），所以模型可以通过学习这些常见的、显著的视觉特征来提高检索准确率。**也就是说，由于文本描述通常是针对图像中的显著内容而作出的，从图像中检索文本可能在某种程度上相对简单。而要从文本中找到一个完全匹配的图像，模型需要能够理解和解析文本描述中的各种概念，并将这些概念与视觉内容准确对应起来，这可能是一个更加复杂的过程。

**2. MSCOCO 与 Flickr30K 数据集的 R@1 对比差异：**

- **数据集内在特性：** Flickr30K 相对较小，图像和对应的文本描述可能更加具体和直接。这可能使得对于 Flickr30K 数据集，**图像与文本之间的一致性更高**，从而提高了模型的检索性能。

- **培训和评估设置：** 如果两个数据集在图像的种类、数量和分布上有所不同，这可能影响模型的性能。例如，如果 Flickr30K 在某些类别上具有更多样的图像，而这些类别正好是模型更擅长识别的，那么在这个数据集上的表现自然会更好。

- **任务难度：** **图像检索任务**要求模型理解文本描述，并找到一个与之高度相关的图像。这需要模型捕捉到更细微的视觉特征，而且这些特征必须与文本描述中的语义紧密对应。由于 **MSCOCO 的多样性和复杂性**，执行此任务可能比在 Flickr30K 上更具挑战性。

在表格中，我们看到 **ALBEF-base** 在使用 MixGen 之后，在 Flickr30K 数据集上 R@1 的成绩只有小幅提升（从 81.4%提高到 82.4%），而在 MSCOCO 上的提升更为显著（从 55.8%提高到 57.3%）。这说明 MixGen 在较难的数据集上提升了模型的表现，显示出其对数据效率的改善，尤其是在**图像检索**任务上，这可能是因为 **MixGen 增强了模型对图像特征的理解和匹配能力**。

对于**深度学习模型**而言，提高 R@1 分数通常是一个重要目标，因为这意味着模型能够更准确地理解和匹配用户的查询。而在本项目中，MixGen 通过数据增强改善了模型的整体性能，这表明它能够帮助模型更好地学习图像和文本之间的对应关系，从而在更少的数据上达到或超过原始模型在更多数据上的性能。

**综合来看**，这些结果表明，MixGen 方法通过在训练时引入新的图像和文本配对，能够有效地提高模型在不同数据集上的表现和数据效率，特别是在有限数据条件下。

## 问题 3:


**1. 对于查询“一个男人拿着遥控器玩游戏”，相关图像在引入"MixGen"技术之前的排名是 19，在引入之后，它的排名提升到了第 3 位，为什么不是第 1 位呢？**

- **其他因素的影响**。虽然 MixGen 技术可以显著提高相关图像的排名，但是检索模型可能受到其他影响因素的作用，如模型的内在结构、预训练的数据集、算法等，这些因素都可能影响最终的检索结果。

- **图像库中的相似图像**: MSCOCO 是一个广泛的数据集，包含各种各样的图像。可能有多个

图像与查询相似，因此即使检索的排名提升了，最准确的图像也可能并不是第一个被检索到的。

在实际应用中，通常更关注模型整体性能的提升，而不是单个查询的排名结果。排名提高到前三，也已经表示模型在定位相关图像方面取得了显著的进步。

### 2.引入 MixGen 之后，排在检索结果前两张图片可能是什么？

- **高度相关性**：这两张图像可能在某些关键的视觉特征上与查询“一个男人拿着遥控器玩游戏”非常吻合，例如，可能更加明显地展示了一个男人在使用遥控器，或者游戏的场景更加突出。涵盖了查询中的一些关键视觉特征，但不一定完全符合查询的所有文本描述。
- **数据增强效果**：MixGen 通过生成新的图像-文本对来提高数据效率，可能在这个过程中为这两张图像创建了更多的、更精确的文本描述，这反过来又提高了它们的检索排名。

影响检索排名的因素复杂多样，即使是先进的数据增强技术也不能保证在每个查询中都获得最优排名。

## 六：结论与展望

**结论：**

在本次项目中，我们针对**数据增强**提出了两种算法——**风格随机化的数据质量增强算法**和**图文联合的数据质量增强算法。**

**1.风格随机化的数据质量增强算法**

在风格随机化的数据质量增强算法中，我们在 **STL-10 数据集**中进行**分类任务**来验证该算法的有效性。**STL-10 数据集本身存在数据质量问题：图片像素低和存在部分噪声图片**。针对该数据质量问题我们首先进行了取出噪声图片的处理，然后使用风格随机化算法对数据集进行**增广**，丰富了数据集的语义多样性，增强了模型的鲁棒性。

在实验中我们不但使用了风格随机化数据质量增强算法，也使用了常见的数据质量增强算法——对图片进行裁剪，旋转，复制粘贴。我们将进行数据增强的数据集、应用常见数据质量增强的数据集、应用风格随机化数据质量增强算法的数据集和应用常见数据质量增强算法和随机化风格数据质量增强的数据集这四种数据集在分类任务上的结果进行对比。

我们发现未增强的数据集准确度只有 60%，而进行了风格随机化数据质量增强的数据集准确度有 70%。该算法对于分类任务有 10%的提升，但提升依然不是很明显。但将常见的数据质量增强算法和随机化风格算法共同使用时，准确度达到 80%。有了极大提升。

**2.图文联合数据质量增强算法**

在图文联合数据质量增强算法中，我们使用 MSCOCO 数据和 Flickr30K 数据集进行图文匹配任务。对于 **MSCOCO 数据集**，其存在以下两个问题：**数据集不完整**；每张图片中存在的目标个数多，提取特征较为困难。对于 **Flickr30K** 数据集，其存在以下两个问题：**数据集较小**；图文描述存在很多无效词语，且未能涵盖图片中所有目标特征。

针对这些数据集问题，我们使用**图文联合数据质量增强算法**对数据集进行处理。首先我们进行模型的预训练。在该实验中我们主要采用 ALBEF 预训练模型进行图文匹配任务。我们

将 MIxGen 算法插入到 ALBEF 预训练模型中，并使用它进行图文匹配得到结果。

对比我们得到的实验结果和论文中的实验结果，我们可以发现，虽然我们训练出来的 ALBEF 预训练模型性能略低，但是在图文匹配任务上的准确度却高于论文。这也清楚地表明了 MIxGen 算法的性能。

其次我们还研究了 MixGen 算法对于数据效率的影响。我们将数据集中的独特图片数量从 3M 减少到 2M.1M 和 200K。得到的结论是 MIxGen 算法总是在原来基础上对于任务准确度进行了提升。特别是在样本数量少时尤为明显。

**展望：**

虽然 MixGen 在多个设置中展示了其有效性，但还有许多因素未考虑。

1. 存在许多流行的单模态数据增强技术，未来的研究可以将这些技术与 MixGen 进行比较。
2. 由于计算限制，只在最广泛采用的有限的数据集上进行了预训练，并与其他工作进行了公平比较。
3. 随着更大规模的数据集变得公开可用，如 LAION-400M，未来的工作可以探索在这些更大的数据集上应用 MixGen。

探索 MixGen 对这些方法和数据的泛化能力将是未来研究的一个有趣方向。

**除此之外，在前面提到的问题 3 中，我们对于图片 2 的"3->19"仍有需要研究空间。**


A man playing a game with a remote controller.

我们可以提出一些问题：

1.为什么经过算法提升后该图片仍然不能像第一张那样达到第一？

2.既然它排在第三，前 2 张图片可能是什么情况？

情况 1：前两张图片是和该图片一样的甚至更符合文本描述。

情况 2：前两张图片和该图片相比完全不匹配文本的描述。

针对情况 1：我们有没有优化的空间，让该图片更大概率地排到第一？

针对情况 2：这两张完全不相关的图片会出现在前面的原因是什么？

对于问题 1，我们在前面的问题解答 3 里面已经做出了猜想和回答。

对于问题 2，我们仍然需要更进一步的研究和讨论。

我们目前的一些研究思路：

**情况 1：如果前两张图片与查询文本一样甚至更符合描述：**

- **模型训练和特征学习：** 对于视觉-文本匹配模型，训练过程中学习的特征及其权重分配对于检索任务至关重要。模型可能更倾向于一些特定的特征，比如特定的背景、颜色或物体形状。如果其他图片中有类似特征出现得更频繁或更突出，模型可能会给予这些图片更高的相关性评分。
- **优化空间：** 为了提升特定图片的排名，可以采用**细粒度的特征学习方法**，强化模型对于关键元素（如人物动作）的理解。此外，可以使用**注意力机制**来帮助模型更好地聚焦于文本描述中的关键信息。增强数据集中的正样本和负样本的对比学习，也可以帮助模型更准确地区分相关与不相关的图片。

**情况 2：如果前两张图片与查询文本不符：**

- **模型的泛化能力：** 这可能是模型对文本中的某些词汇或短语的泛化能力不强，对某些视觉元素的理解存在偏差，导致评分高的图像与查询文本不完全吻合。
- **不相关图片出现的原因：** 可能是因为模型在某些情况下对于视觉特征的理解过于字面化或简化，比如将手中的任何物品误认为是“remote controller”。此外，训练数据集可能存在偏差，某些与查询文本不相关的图片因为在训练集中出现频率高，被模型误认为与查询文本相关。
- **优化空间：** 增强模型对于关键词的理解，可能需要通过**改进模型的语义分析能力**，例如使用**更先进的自然语言处理技术**来理解查询中的意图。此外，利用**负样本的对比学习**，教导模型识别出与查询文本不相关的图像特征，也是提高准确性的一种方法。