

**机器智能课程设计报告**



**题目: 基于编解码框架的方法的图像描述生成**

|  |  |  |  |
| --- | --- | --- | --- |
| **姓名** | **学号** | **学院** | **班级** |
| **郭鸿铭** | **2021213392** | **人工智能学院** | **2021219110** |
| **耿翊中** | **2021213382** | **人工智能学院** | **2021219110** |
| **吴秀鹏** | **2021213370** | **人工智能学院** | **2021219110** |

**2023年12月**

**一、题目：**

基于编解码框架的方法的图像描述生成

**二、任务描述：**

1、任务目标:

开发一个能够自动分析图像内容并生成相应文字描述的系统。这个系统应能理解图像中的关键元素（服饰元素）并用自然语言准确描述它们。

2、步骤和程序:

(1)数据准备: 使用AIC-ICC数据集，210,000幅图像组成的训练集，30,000幅图像组成的验证集，30,000幅图像组成的测试集A和30,000幅图像组成的测试集B，每幅图像对应5个人工标注的中文句子描述。

(2)模型设计:

编码器: Transformer编码器。

解码器:Transformer解码器。

(3)训练: 使用标注数据训练模型，优化其性能以准确地生成描述。

(4)验证和调整: 在独立的测试集上评估模型，进行必要的调整以提高准确性和自然性。

3、参与者:

郭鸿铭、耿翊中、吴秀鹏

4、评估标准:

METEOR、ROUGE-L、CIDEr-D、SPICE

**三、预期目标：**

成功实现图像描述生成，能够自动理解和解释图像的内容，并生成准确、相关且自然的文字描述。

**四、相关工作：**

完成本次课程设计项目，涉及已有的相关工作有：

1、编码器，解码器结构：

编码器-解码器架构是一种神经网络架构，广泛应用于自然语言处理和计算机视觉领域。这种架构由两部分组成：编码器和解码器。编码器的作用是将输入数据（如文本或图像）编码为一个定长的向量，这个向量称为上下文向量，它包含了输入数据的主要信息和特征。解码器的作用是根据上下文向量解码出目标输出，如文本翻译、语音合成或图像描述。这种架构的优势在于能够将不同形式的输入和输出数据进行有效的处理和转换，尤其是在跨模态的任务中（例如，将图像转换为文字描述）。该架构具有灵活性和高效性，能够适应各种复杂的任务和数据类型。

2、Transformer：

Transformer是一种深度学习模型，它在序列数据的处理上有着突出的表现，特别是在自然语言处理领域中有着广泛的应用。它的核心创新是自注意力机制，它能够让模型在处理序列时关注每个元素与其他元素的联系，从而有效地捕捉序列中的长距离依赖关系。Transformer架构由两部分组成：编码器和解码器。编码器的任务是将输入序列编码为一系列与上下文相关的向量，而解码器的任务是根据这些向量生成输出序列。在图像描述的任务中，Transformer的解码器发挥了重要作用：它能够利用来自图像处理模型（如卷积神经网络或Vision Transformer）的上下文向量，并根据这些向量生成流畅、有意义的自然语言描述。这种结合了自注意力和解码器的技术，使得Transformer在生成精确、细致的图像描述上非常有效，尤其是在需要理解和表达视觉中的复杂元素和关系时。

3、Vision-transformer：

Vision Transformer (ViT) 是一种创新的模型，它将原本主要应用于自然语言处理的Transformer架构成功应用于计算机视觉任务。ViT 的核心思想在于将图像分割成多个小块，即“patches”，并将这些块转换为序列化的数据，从而使得Transformer架构能够用于图像处理。每个图像块被转化为向量，并输入到标准的Transformer模型中进行分析。这种方法利用了Transformer的自注意力机制，允许模型有效地理解和捕捉图像不同区域之间的复杂关系，这对于图像内容的深入理解至关重要。Vision Transformer的一个显著优势在于其能够并行处理多个图像块，同时它摆脱了传统卷积神经网络对局部特征依赖的限制。因此，ViT 在图像分类、对象检测等多种视觉任务上展现了优异的性能。自从首次推出以来，Vision Transformer引起了计算机视觉领域的广泛关注，并促进了众多变体和改进技术的发展。

**五、技术方案：**

1、VIT与Transformer解码器结合:

使用Vision Transformer (VIT) 提取图片信息。

将图片切割成多个patches，每个patch作为一个token输入VIT Encoder。

通过Encoder获得视觉表示，这些表示作为键（K）和值（V）。

Transformer解码器产生的输出作为查询（Q）。

通过交叉注意力机制，迭代地输出自然语言描述。

训练阶段使用Teacher forcing方法，推理阶段应用beam search策略。

2、VIT整体表示与GRU的结合:

首先，通过Vision Transformer处理图像。

VIT将图像分割成patches，通过其Transformer编码器层处理这些patches。

生成的class token作为图像的整体表示。

设计基于GRU的解码器，以class token为初始状态。

解码器的目标是基于图像表示生成自然语言描述，每次生成一个单词，直至形成完整句子。

训练时使用交叉熵损失函数，采用teacher forcing模式，端到端优化模型。

推理时使用beam search生成自然语言。

3、网格表示与Transformer编解码器:

使用CNN autoencoder（如ResNet或VGG）提取图像特征，得到一个压缩的网格表示。

将网格特征转换为序列，添加位置嵌入以保留空间信息。

这些特征输入到Transformer编码器，通过自注意力机制处理。

设计一个Transformer解码器基于编码器输出和之前生成的单词生成下一个单词。

以端到端方式训练整个系统。

4、网格表示与注意力机制解码器:

通过CNN autoencoder（如ResNet或VGG）提取图像特征，形成网格表示。

网格特征转换为序列，加入位置嵌入。

使用基于注意力机制的解码器（GRU或LSTM）处理特征，生成自然语言描述。

解码器在生成每个单词时聚焦于图像的不同部分。

与传统Transformer解码器不同，此处直接使用CNN输出作为解码器输入。

**六、开发环境：**

环境：Google colab，ubuntu

IDE：vscode,Jupyter Notebook

显卡：Nvidia Tesla T4,3090

框架：Pytorch，Pytorch Lightening

**七、组员分工及检查点：**

1、组员分工：

|  |  |
| --- | --- |
| **姓名** | **任务分工** |
| **郭鸿铭** | 模型结构代码；训练\测试流程接口设计 |
| **耿翊中** | 评估代码；训练，测试，调参 |
| **吴秀鹏** | 数据处理；训练，测试，调参 |

1. 检查点：

|  |  |  |
| --- | --- | --- |
| **检查点1** | **吴秀鹏** | 数据处理部分：dataloader代码 |
| **检查点2** | **耿翊中** | 评估部分：evaluation模块代码 |
| **检查点3** | **郭鸿铭** | 模型部分：pytorch模型代码 |
| **检查点4** | **郭鸿铭** | 接口设计部分：pytorch lightening实现代码 |
| **检查点5** | **吴秀鹏**  **耿翊中** | 训练，测试，调参部分：  训练日志 tensorboard记录 |