**项目名称**  
基于VITON模型的虚拟试衣  
  
**项目背景**  
基于本学期学到的机器视觉的知识和相关实验，我们大部分的实验作业都是基于人脸和人手的识别和处理，所以在该项目中，我们希望机器视觉和深度学习的知识从头部和手部延伸到人的躯干。因此，我们决定以虚拟试衣来实现这一想法。

虚拟试衣使得用户能够在不用进行物理穿戴的情况下体验穿衣效果，因此，用户能够迅速的确定是否喜欢一件衣服然后决定是否进行购买。

基于图像的虚拟试穿系统，将虚拟试穿转化为条件图像生成问题，提供了一种更经济的方法，也实现了比较好的结果。给定两张图片：一张是人，另一张是商店里的衣服，上述方法的目的是合成一张满足以下要求的图像：

1.人穿着新的衣服

2.原始的身体和姿态都得到了保留

3.衣服被填充到了人体上，并且能够实现保真、平滑、无缝

4.衣服的特征，比如质地，logo，文字都被良好的保留，并且没有醒目的变形和缺陷

**大致流程**  
1. 加载VITON数据集并对其进行预处理，作为GMM-data  
2. 训练GMM模型  
3. 测试GMM模型  
4. 基于GMM的结果生成TOM-data  
5. 训练TOM模型  
6. 测试TOM模型

**代码结构说明**  
1. 数据加载和预处理部分：  
   - 从data/train文件夹中加载数据集，数据集包括cloth,cloth-mask,image,image-parse,

pose五部分。

定义了一个名为 Dataset 的数据集类和一个名为 DataLoader 的数据加载器类，这些类用于处理衣服图像、人体图像和对应的遮罩，以及姿势关键点等信息，为了在几何匹配模块（Geometric Matching Module）中使用。

首先，Dataset 类是一个自定义的数据集类，它继承自 PyTorch 中的 data.Dataset 类。这个类用于加载并预处理衣服图像、人体图像、遮罩和姿势关键点等数据。它实现了 \_\_init\_\_、\_\_getitem\_\_ 和 \_\_len\_\_ 方法：

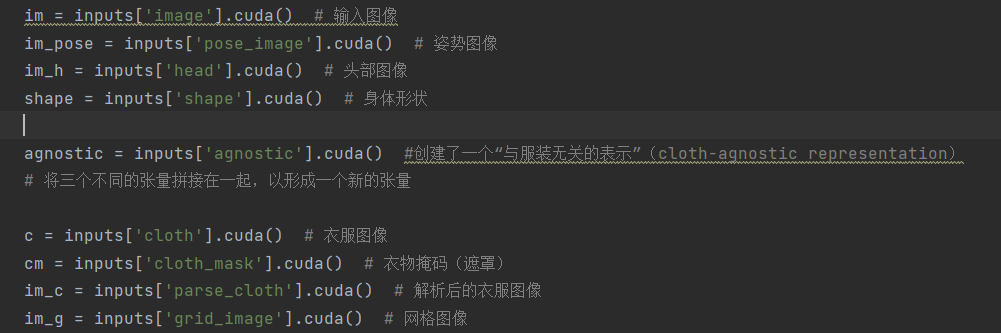
\_\_init\_\_ 方法：初始化数据集，设置数据集根目录、数据模式、阶段、数据列表等参数，并加载相关数据信息。

\_\_getitem\_\_ 方法：根据给定的索引获取数据集中特定索引位置的数据项，对衣服图像、人体图像、遮罩和姿势关键点等进行预处理。

\_\_len\_\_ 方法：返回数据集中图像的数量。

另外，DataLoader 类是一个数据加载器类，它用于批量加载数据集中的样本，并支持对数据进行迭代访问。它接受数据集对象并根据参数设置创建 PyTorch 的数据加载器。next\_batch 方法用于获取数据加载器的下一个批次数据。

代码的最后部分用命令行参数设置了各种数据集相关的选项，创建了数据集对象和数据加载器对象，并展示了数据集和数据加载器的大小信息。最后，通过调用 IPython 库中的 embed() 方法，进入交互式的 IPython 环境，以便用户可以进一步探索和操作数据。

   -在cp\_dataset.py中加载数据和对数据进行预处理，最终会形成训练需要用到的如下数据。  
     
2. 参数设置：分别在train和test文件开头通过get\_opt方法设置好参数,下图是train文件设置的参数  
    
3. 训练：在train文件中分别对gmm和tom进行训练。

开始训练：确认 train.py 中的 opt 参数正确配置后，运行 train\_gmm() 函数，该函数会引用 networks.py 中定义的 GMM 类作为模型来开始训练。训练大约需要进行 200,000 个步骤。

保存权重：

训练结束后，会生成 finnal\_gmm.pth 文件，这个文件将作为测试阶段的权重。在上述的参数设置的图里面有一个step参数，使其50000为一个循环，这样可以解决训练过程中途训练崩掉的情况。

1. 网络模型：networks文件给出了网络的代码，VITON由gmm和tom组成。

5. 测试：测试 GMM 模块

设置命令行参数，修改命令行参数以进行测试：python test.py --datamode test --stage GMM

运行测试：运行测试过程会在 result 文件夹内生成两个新的数据集 warp\_cloth 和 warp\_mask。将测试数据放入训练集中，将生成的 warp\_cloth 和 warp\_mask 文件放入 data/train 中，以便后续训练 TOM 模块时使用。TOM模块的训练同理   

Networks**说明**

**原理:**

Geometric Matching Module（GMM）

它的输入是人物表示p和目标产品图片c，p和c的高级特征会由两个网络提取，然后被一个相关层组合到一起。与VITON的编码器-解码器生成器不同，GMM的输出并不是目标变形衣服形状的mask，而是变形后的衣服。

GMM通过训练一个 regression network 预测空间变换参数并使用TPS转换来生成变形后的衣服，其代价函数为变形后的衣服与Ground True 的L1损失。直接使用像素对像素的误差能提升网络的效率，不过不一定能获得更好的生成效果。GMM不像VITON使用预先训练的VGG19，而是从头开始训练。

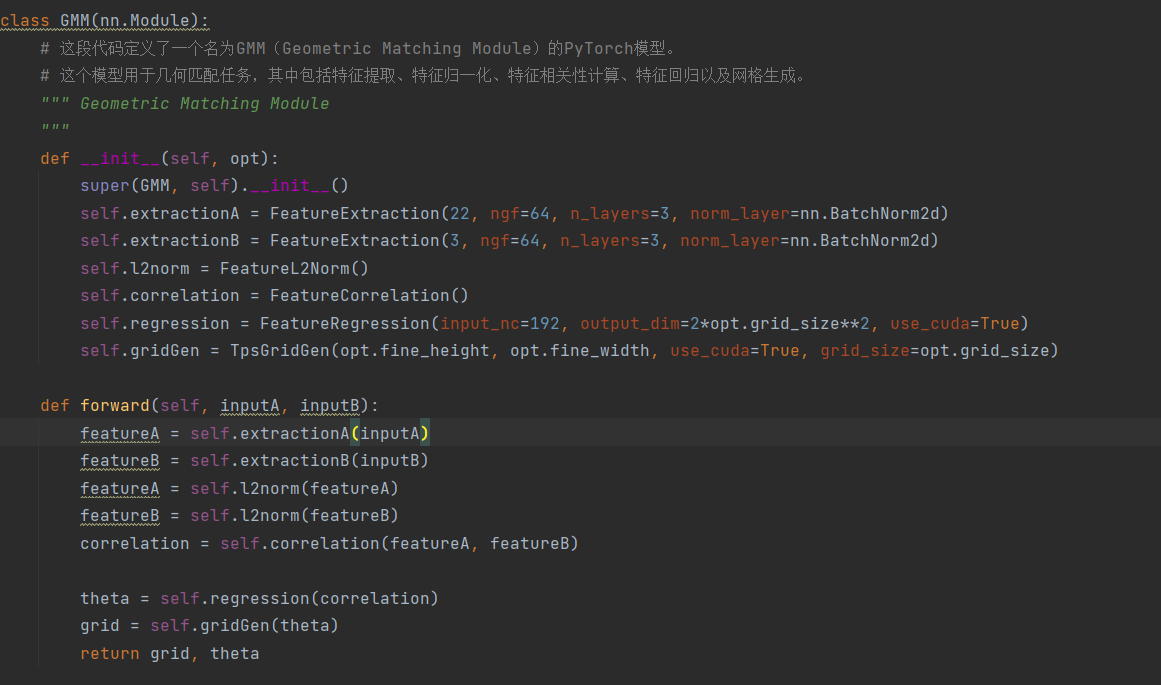
在 GMM部分，主要分为 Feature Extraction Network （用于人物表示和衣服的特征提取）和 Regression Network 两部分。在 Feature Extraction Network 部分中，包含四个2-stride的下采样卷积层，后接两层卷积层使用了 1-stride以维持输出的维数。Regression Network 包含两个 2-stride 的卷积层、两个 1-stride 的卷积层、一个全连接的输出层。文章也给出了 SCMM 与 GMM 的效果对比，结果显示 GMM 在处理较大幅度的变形的时候能获得比 SCMM 更好的效果。

Try-on Module（TOM）

给定人物表示和变形衣服，UNet同时生成渲染人物与Composition Mask，渲染人物与变形衣服将通过Composition Mask合成最终结果。

和 VITON 的 Refinement Network 差不多，比较具体的不同是在于代价函数的选择。在 VITON 中，代价函数只用了 VGG19感知损失，而在 CP-VTON 中，作者加上了 pixel-wise 的 L1 损失。此外，由于 CP-VION 在 GMM 中并没有得到 segmentation mask，在计算 Composition Mask 的误差的时候无法照搬 VITON 的损失函数。基于特征保留的目的，作者使用了 1 - M 的 L1范数来增强服装部分像素点对 loss 函数的影响。

Networks主要由GMM和UnetGenerator（即TOM）两个模型组成



1.GMM：这个模块整合了特征提取、归一化、相关性计算、回归和网格生成等过程，用于完成几何匹配任务。

此外，还包含了保存和加载模型的函数 save\_checkpoint 和 load\_checkpoint，用于保存和加载模型的参数。

这些模块和函数共同组成了一个完整的神经网络，用于处理图像、特征提取、几何匹配等多种任务。



2.tom在初始化过程中，通过循环构建 UNet 结构的各个模块 UnetSkipConnectionBlock。

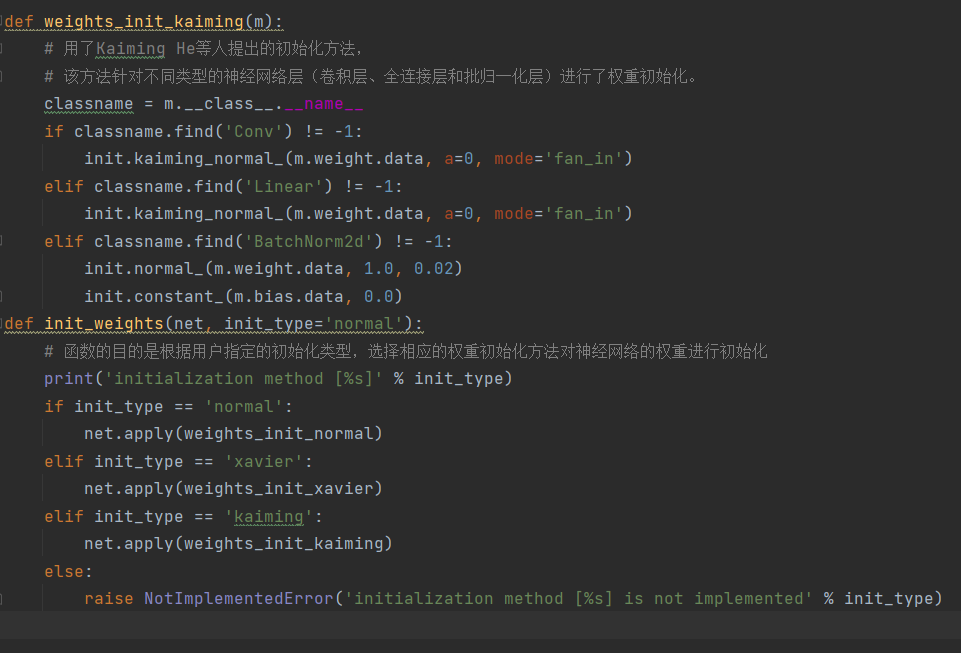
从 innermost=True 开始，逐步构建编码器部分（下采样过程），并将每个模块作为前一个模块的子模块。

num\_downs - 5 控制了 UNet 结构中下采样模块的数量。每个 UnetSkipConnectionBlock 模块都接受相应的输入通道数和输出通道数，并在堆叠 UNet 结构的过程中逐渐减小通道数，形成对称结构。最后一个 UnetSkipConnectionBlock 设置为 outermost=True，代表解码器部分的最外层，输入通道数为 output\_nc。

forward 方法：定义了模型的前向传播过程。它接受输入 input 并将其传递给 UNet 结构的模型，最终返回生成器模型的输出。

总的来说，这段代码构建了一个 UNet 生成器，该生成器在给定输入图像后能够进行编码和解码过程，用于执行图像到图像的转换任务，例如图像分割、图像翻译等。

3.不同类型的神经网络层（卷积层、全连接层和批归一化层）对权重进行初始化。这是一开始为后面模型的建立做准备。



5.

FeatureExtraction:

作用：这个类用于从输入图像中提取特征。

功能：它包括多个卷积层和归一化层，用于逐级提取输入图像的特征表示。通过堆叠卷积和激活函数，逐步提高特征的抽象层次。

FeatureL2Norm:

作用：在特征提取后，这个类用于执行特征的 L2 归一化。

功能：L2 归一化是一种常见的归一化方法，用于使特征向量的范数（L2范数）等于1。这有助于减少特征之间的尺度差异，使得特征更具鲁棒性和可比性。

FeatureCorrelation:

作用：执行特征之间的相关性计算。

功能：根据输入的特征张量计算特征之间的相关性。通常用于计算两个特征图之间的相关性矩阵，以便在几何匹配任务中对特征进行匹配。

FeatureRegression:

作用：执行特征的回归。

功能：用于从特征相关性中学习几何变换的参数。这个类通过卷积和全连接层将特征相关性转换为几何变换的参数，例如仿射变换或其他变换的矩阵参数。这些参数用于估计如何将一个特征图中的特征点映射到另一个特征图中的对应点，从而实现几何变换的目的。

AffineGridGen：

作用：用于生成仿射变换的网格。

功能：根据输入的仿射变换参数（通常是仿射变换矩阵），该类可以生成一个仿射变换的网格。仿射变换是一种线性变换，可以实现平移、旋转、缩放和剪切等操作。

TpsGridGen：

作用：用于生成 Thin-Plate Spline (TPS) 变换的网格。

功能：Thin-Plate Spline 是一种非线性的网格变换方法，它使用一组控制点来对输入的点进行变换。TPS 网格变换是基于这些控制点计算的，能够更灵活地处理非线性变换，适用于处理复杂的形变。

这两个类通常用于计算几何变换的网格，其中 AffineGridGen 用于较简单的仿射变换，而 TpsGridGen 则适用于更复杂的非线性变换。在许多几何匹配任务中，这些类可以用来生成变换网格，从而将一个图像或特征点集中的点映射到另一个图像或点集中的对应点，实现几何上的对齐或匹配。

5和6都是用于后续的网络构建。

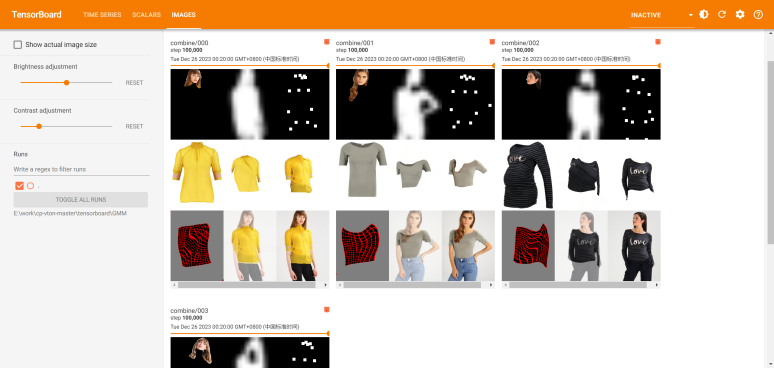
**环境配置**

CUDA 和 cuDNN：用于 GPU 加速的计算

Anaconda：搭建python虚拟环境

PyTorch：基于此深度学习框架进行训练

opencv：加载处理图像数据

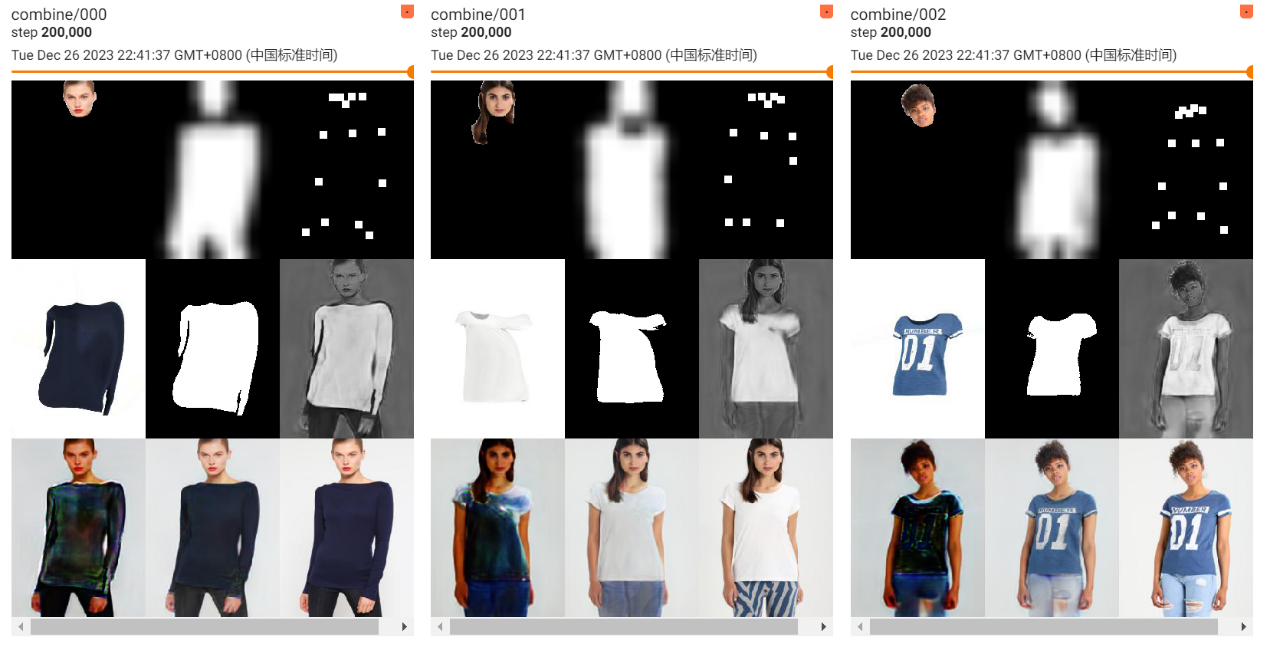
tensorflow、tensorboardX：可视化训练过程  
  
**运行效果**  


训练完gmm后的效果

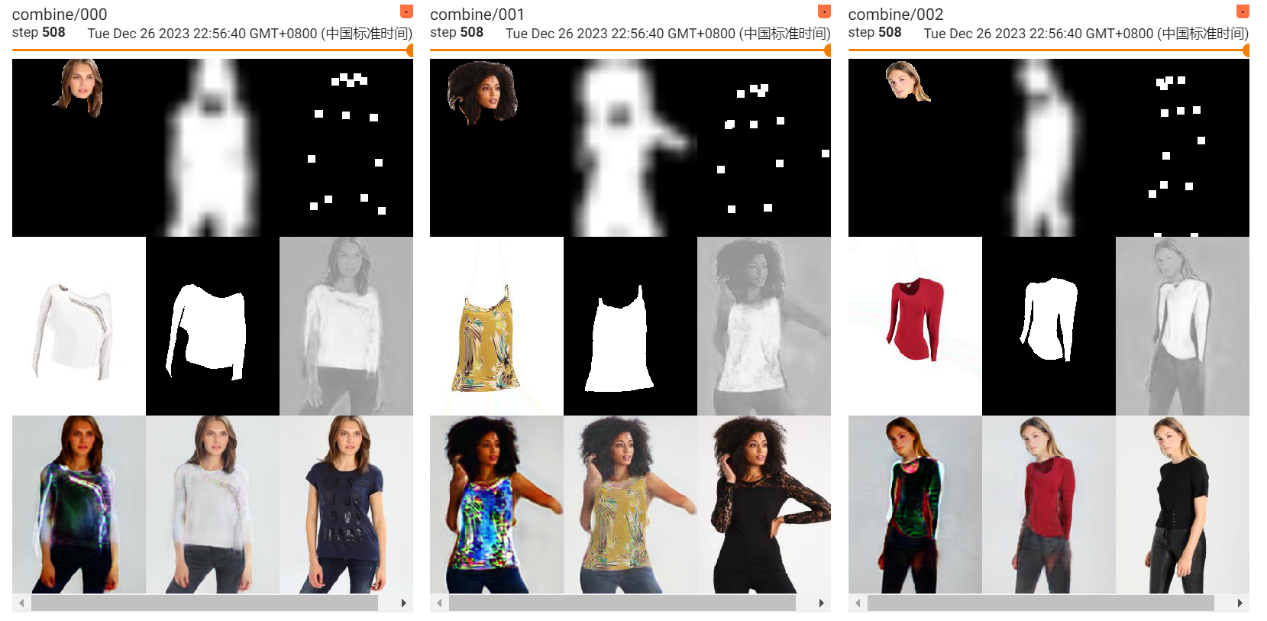
图形用户界面

描述已自动生成

测试完gmm后的效果



训练完tom后的效果



卡通人物

描述已自动生成

测试完tom后的效果

**引用:**

模型参考自论文

Toward Characteristic-Preserving Image-based Virtual Try-On Network

Bochao Wang, Huabin Zhang, Xiaodan Liang, Yimin Chen, Liang Lin