

基于 Stable-diffusion 和 HR-VITON 的 T 恤图案设计与虚拟试衣

1 实际操作部分

1.1 基于 Stable-diffusion 训练 T 恤生成模型

1.1.1 数据预处理

我们使用的服装数据集为 Kaggle 上的 Fashion Product Images Dataset 数据集 (<https://www.kaggle.com/datasets/paramaggarwal/fashion-product-images-dataset>)。该数据集包含 44441 张 Fashion Product 图片，图片内容如下（以 T 恤为例）：



图 1 Fashion Product Images Dataset 图片

该数据集中还附有一个 csv 文件，记录了这些 Fashion Product 的 id、适用人群、主要类别、次要类别、颜色、适用季节、名称（含特征）等信息。

观察数据集中的服装类别分布，我们发现上衣（Topwear）样本的占比超过 1/3（bottomwear 只有 2694，另外比较多的是 shoe[7344]、bag[3055]和 Watch[2542]）。为了扩充 T 恤的可选特征，我们并没有把非 T 恤的上衣样本排除在训练样本之外。

我们从中筛选出 subCategory 为 Topwear 的样本作为我们的训练样本，共计 15405 份。然后我们利用 csv 里面的数据给这些样本生成标签，具体的生成公式如下：

$$Prompt = \left\{ \begin{array}{l} productDisplayName, subCategory, \\ gender, articleType, baseColour, season, usage \end{array} \right\}$$

其中 productDisplayName 是名称、subCategory 是次要类别（也就是 Topwear）、gender 是适用性别、articleType 是类型（细分的服装类型）、baseColour 是底色、season 是适用季节、usage 是使用场景。

如果遇到 productDisplayName 为空值，那就放弃该样本；如果遇到其它属性为空，则忽略该属性。

生成完标签之后，我们就能进行 Stable-diffusion 的训练了。

1.1.2 训练 Stable-diffusion

训练代码引用自 <https://github.com/Victarry/stable-dreambooth> 与 <https://github.com/CrazyBoyM/dreambooth-for-diffusion>。

代码逻辑如下：

- 1) 利用自编码器将图像转化到 latent space（隐空间）。
- 2) 生成与图像等大的噪声矩阵。
- 3) 给每张图像添加噪声。
- 4) 获取文本 Embedding。
- 5) 计算 noise residual。
- 6) 计算 loss。

在训练之前，我们还要对图像进行裁剪。我们需要的输入图像是长宽 1:1 的图像，于是，我们可以对我们的图像先进行 CenterCrop。由于我们的上衣图片基本上都位于图片正中间，因此 CenterCrop 也不会误删关键信息。

训练参数如下：分辨率 512*512；batch_size 大小为 4；学习率 2e-6；训练步数 12000 步；模型使用半精度训练，降低显存占用；优化器使用 Adam。在这些参数的设置下，模型在单卡 A5000 上训练 3 个小时即可微调完成。

1.1.3 纯 Stable-diffusion 训练效果展示

训练完成后，生成效果如下（以 T 恤为例，prompt 选取 “tshirt, rabbit”）：





图 2 纯 Stable-diffusion 训练效果展示

可以看到，生成的图像符合 prompt 中 tshirt 和 rabbit 图案的需求。但不足的是，由于我们需要做的是虚拟换衣，而 HR-VITON 的衣服输入需要正面无褶皱的衣服，最好是没有模特的。因此，我们还需要给模型做微调，让模型的输出是单件 T 恤。这就需要用到下面的 Dreambooth。

1.2 基于 Dreambooth 进行 Stable-diffusion 微调

1.2.1 选取模板图片

我们希望 Stable-diffusion 生成的图像是单件 T 恤，于是我们引入 Dreambooth，导入四张正面的、无褶皱的单件 T 恤图片模板（没有引入有图案的模板，避免 Stable-diffusion 学到不必要的特征），对 Stable-diffusion 进行进一步的微调。

其中，我们选取的模板图片如下（来自 HR-VITON 的数据集）：



图 3 模板图片

1.2.2 训练

从原理部分我们知道，Dreambooth 引入了一个叫“特殊标识符”的东西。这个“特殊标识符”是一个没有训练过的文本 token，在语义上代表我们希望 Stable-diffusion 学到的特征。于是，在我们这次微调中，我们把这个“特殊标识符”命名为“[T]”，代表了单件 T 恤的特征。然后将它的类别划分在 tshirt 的范围内，利用了 tshirt 的先验特征。

代码依然是参考前一部分 Stable-diffusion 章节的两个仓库。与之前的训练代码稍有不同的是，在 Stable-diffusion 的训练中我们冻结了 Text Encoder（也就是文本编码器），但是引入

Dreambooth 后我们需要开启 Text Encoder 的微调，因为增加了一个文本 token。

综合上述分析，我们在本次微调中设置的 instance_prompt 为 “a photo of [T]”，class_prompt 为 “a photo of tshirt”。此外，基于 Dreambooth 训练 Stable-diffusion 的训练参数如下：分辨率 512*512；batch_size 大小为 1；学习率 2e-6；训练步数 1000 步；模型使用半精度训练，降低显存占用；优化器使用 Adam。在这些参数的设置下，模型在单卡 A5000 上训练 10 分钟即可微调完成。

1.2.3 基于 Dreambooth 微调 Stable-diffusion 效果展示

微调完成后，我们把“特殊标识符[T]”添加到 prompt 中，作为图片生成的其中一个条件。当我们选取 prompt 为 “tshirt, rabbit, [T]” 时，生成图片如下：



图 4 基于 Dreambooth 微调 Stable-diffusion 效果展示

可以看到，生成的效果比前面没有引入 Dreambooth 时好了很多。生成的 T 恤图片满足 prompt 中生成 rabbit 图案的需求，而且生成的图片中均没有包含模特。这说明我们的 Dreambooth 起作用了，Stable-diffusion 确实学习到了单件 T 恤的特征。

至此，T 恤图案设计器圆满完成，我们可以根据 prompt 中的请求生成不同图案的 T 恤。

1.3 训练 U2Net 获取 T 恤 mask

1.3.1 数据集介绍

在 U2Net 的训练中，我们选取的数据集是 iMaterialist (Fashion) 2019 at FGVC6 dataset。这份数据集来自 Kaggle，是一个服装图片分割比赛的赛题数据集。

该数据集包含室内人物照、室外人物照、正面照、侧面照、不含模特的服装照片等等，内容非常多，各种条件的服装照片都有，可以很好地进行服装分割模型的训练。此外，该数

数据集还提供了一个 csv 文件，里面是对应每一张图片的分割数据，精确到每一个位置的像素的属于哪一类的衣服。于是，我们可以用这个数据集来训练 U2Net。

1.3.2 训练

由于在数据集的分割数据中，分割的类别达到了 42 类，而我们不需要这么多。于是我们只把和 T 恤相关的分割点标记为有效的分割点，其它的分割数据都舍弃，最后把不含 T 恤的删去。

训练代码参考了 <https://github.com/levindabhi/cloth-segmentation> 仓库。

训练的思路很简单，每一个 step 的步骤如下：

- 1) 获取 U2Net 各层得到的分割图；
- 2) 对比预测分割图 and 实际分割图的差异；
- 3) 计算 loss；
- 4) 最后根据 loss 对模型进行调整。

训练优化器使用 Adam，学习率设为 0.001；损失函数选取交叉熵损失；batchsize 设置为 12。

1.3.3 获取服装 mask 效果展示

U2Net 训练完成后，我们将其用于给前面生成的 T 恤制作 mask。效果如下：



图 5 利用 U2Net 获取 mask

可以看到，总体上，mask 分割出了背景和衣服。虽然 mask 在边角处会出现毛刺状，但是这对虚拟换衣不会有太大影响。

1.4 利用 HR-VITON 实现虚拟试衣

1.4.1 背景预处理

在进行虚拟试衣的实验之前，我们首先需要把生成 T 恤的背景换成 RGB 为 (246, 246, 246) 的白色背景（实验表明一定要换白色背景，否则生成图就会偏白）。实现方法如下：利用 mask 图片，遍历每一个像素；当某一个位置的像素值为 0 时，说明 mask 将该位置分割到背景区域；于是，我们把原图中对应背景区域的点赋 RGB 值 (246, 246, 246)，即可实现更换白色背景。效果如下：



图 6 背景替换

完成背景替换后，我们就能够顺利地使用 HR-VITON 了。

1.4.2 数据集介绍

本部分的数据集是 VITON-HD 数据集，该数据集是一份专门为虚拟试衣项目打造的数据集。该数据集包含服装图片、服装 mask 图片、模特图片（穿着相应服装）、模特服装分割图片以及姿态识别图片。下面展示一组数据：





图 7 VITON-HD 数据集展示

在本部分实验中，由于我们自己生成的 T 恤没有真实试衣图片，所以我们无法用程序获得对应试衣图片的分割图和姿态图，也就无法自己训练 HR-VITON。因此，在本次实验中，我们使用了官方的预训练模型，没有办法做微调。幸运的是，最后实现的效果还不错，这也说明了该模型框架的泛用性好。

1.4.3 实现虚拟试衣

我们的实验代码参考了官方代码仓库 <https://github.com/sangyun884/HR-VITON>。代码的逻辑如下：

- 1) 首先，读取 ConditionGenerator 和 SPADEGenerator 的权重。
- 2) 将衣服与其 mask 的下采样图像和 parse_agnostic 和 densepose 输入进 ConditionGenerator，输出得到 flow_list, fake_segmap, warped_cloth_paired, warped_clothmask_paired。
- 3) 生成 warped_clothmask。
- 4) warped_cloth 由需要更换的衣服和 flow_list 生成。
- 5) parse 由 fake_segmap 生成。
- 6) 最后，将 agnostic、densepose、warped_cloth 拼接起来与 parse 一齐作为 SPADEGenerator 的输入生成最终的图像。

总体流程图如下：

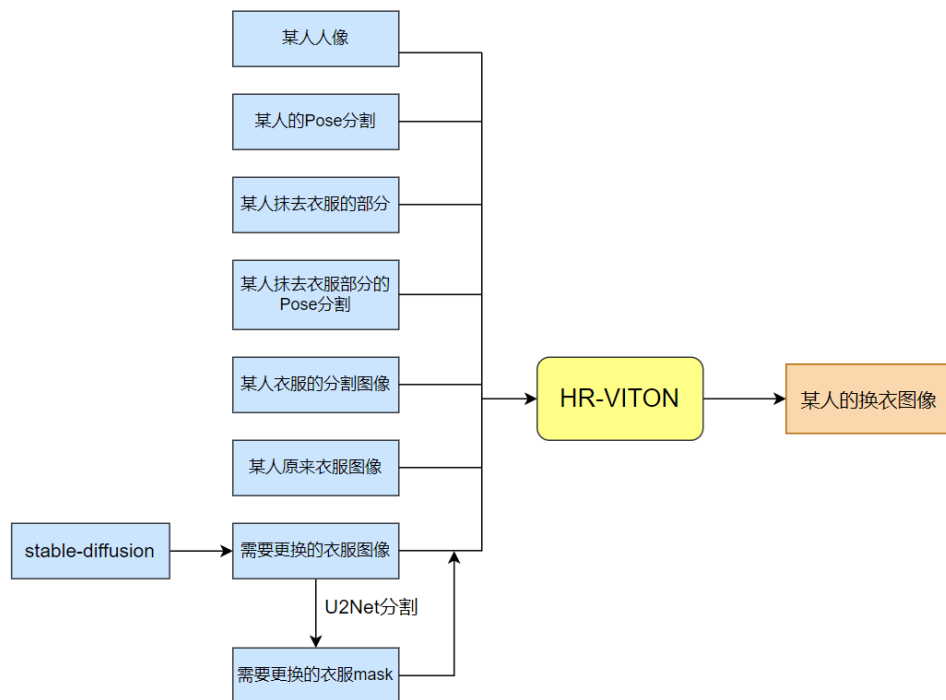


图 8 HR-VITON 流程图

其中，模特的服装分割图和姿态图使用的是 VITON-HD 数据集的原始数据，衣服图像与 mask 图像使用的是我们自己生成的数据。

1.4.4 HR-VITON 虚拟试衣效果展示

最终效果如下：

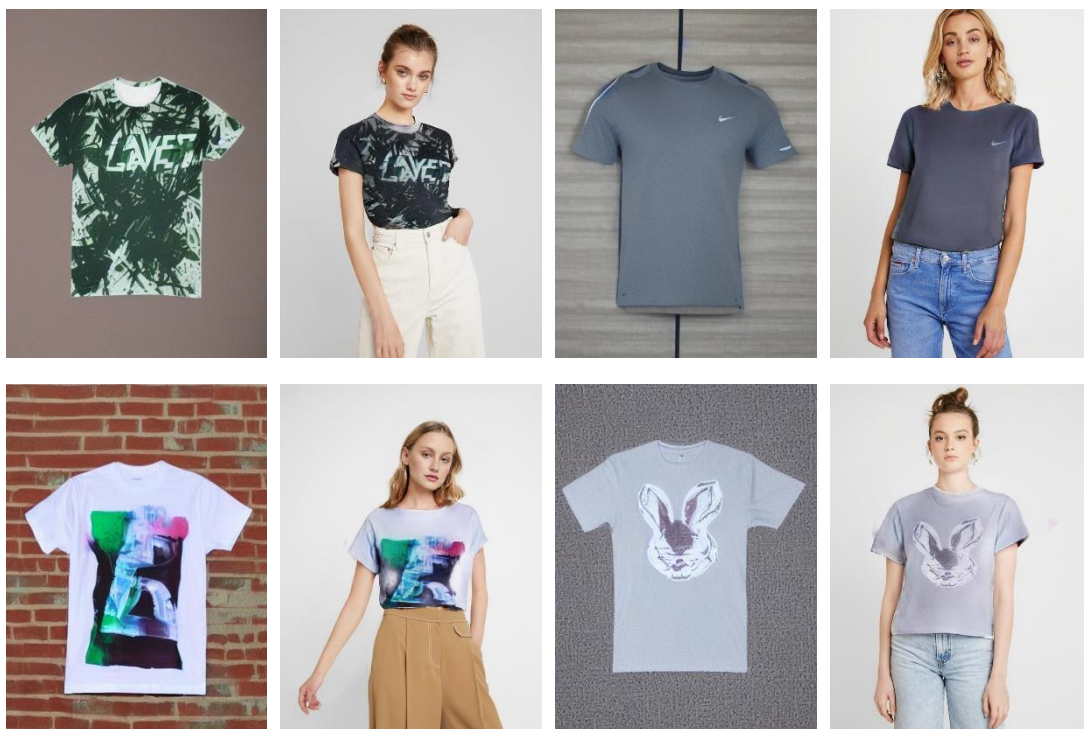


图 9 虚拟试衣效果展示

2 最终成品展示

2.1 定性展示——T 恤图案设计

首先展示基于 Dreambooth 微调 Stable-diffusion 的 T 恤图案设计效果（只展示一部分，作业压缩包内还有更多）。

(1) prompt: tshirt, rabbit, [T]



图 10 tshirt, rabbit, [T]

(2) prompt: tshirt, leaves, [T]





图 11 tshirt, leaves, [T]

(3) prompt: tshirt, nike, [T]

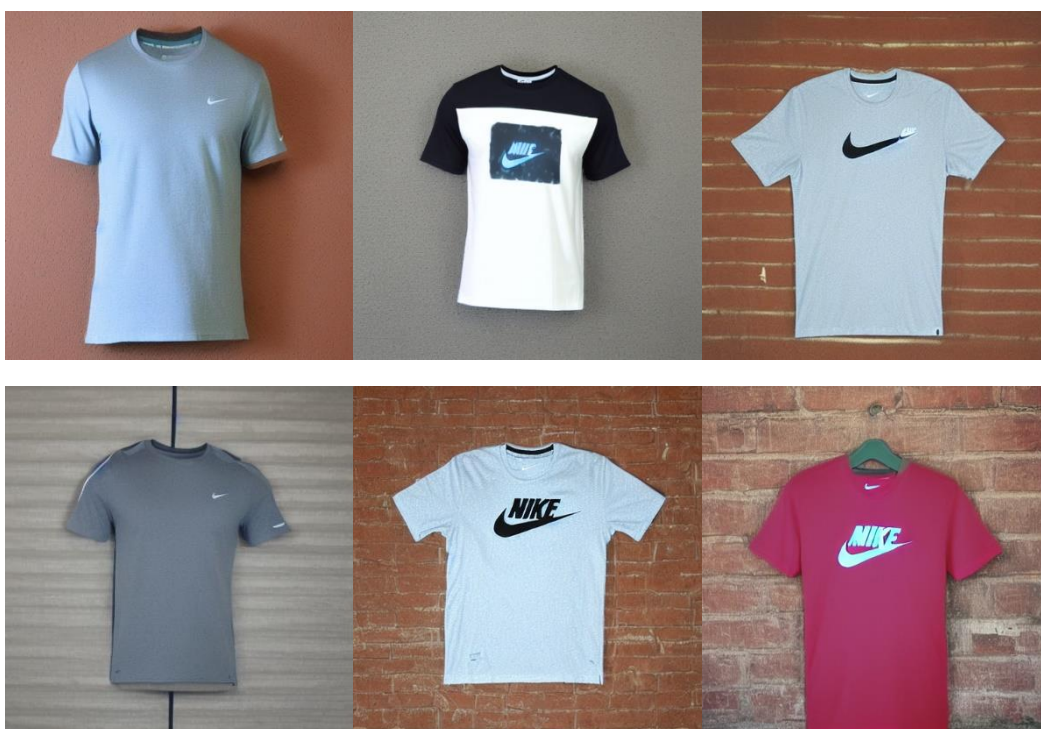


图 12 prompt: tshirt, nike, [T]

2.2 定性展示——虚拟试衣

本部分展示运用 HR-VITON 实现虚拟试衣的效果（只展示一部分，作业压缩包内还有更多）。图像展示格式如下：左上角为模特原图像，其余五张为虚拟试衣图像。

(1) prompt: tshirt, rabbit, [T]



图 13 虚拟试衣 (tshirt, rabbit, [T])

(2) prompt: tshirt, leaves, [T]





图 14 虚拟试衣 (tshirt, leaves, [T])

(3) prompt: tshirt, painting, [T]



图 15 虚拟试衣 (tshirt, painting, [T])

2.3 定量展示

我们选用 FID 分数来进行结果的定量展示。比较的对象为：T 恤图案设计模型（Stable-diffusion+Dreambooth）的生成结果（白色背景）与 VITON-HD 数据集中的 T 恤图片。

最后的 FID 分数为：53.234701930528686。该分数不算非常好，也不算太差。我们推测

原因如下：①由于我们的生成图片数量有限（显存小，批量生成比较慢），所以计算 FID 分数的时候有可能有偏差。②我们是基于 Stable-diffusion 的预训练模型微调的 T 恤图案设计模型，预训练模型里面有大量的 T 恤图片，与 VITON-HD 的风格不同，这会有影响。而且，我们微调 Stable-diffusion 时主要用的是 Fashion Product Images Dataset 数据集，而 VITON-HD 只用在了 Dreambooth 微调部分，所以生成的结果不会完全按照 VITON-HD 的风格来。综合上述分析，得到一个 50+ 的结果也不算差。

3 代码文件夹内容

3.1 clothes_generator 文件夹

存放 Stable-diffusion 代码与 Dreambooth 代码。

名称	说明
my_train_style.sh	训练 Stable-diffusion 的脚本
my_train_object.sh	训练 Dreambooth 的脚本
test_model.py	训练代码
tools 文件夹	存放辅助代码，包含类与函数的定义
test_prompts_style.txt	训练 Stable-diffusion 的验证 prompts
test_prompts_object.txt	训练 Dreambooth 的验证 prompts
README.txt	说明文档

3.2 图片筛选与标签创建文件夹

名称	说明
create_label.ipynb	从 Fashion Product Images Dataset 中筛选 Topwear 图片，并生成 label

3.3 clothes_mask 文件夹

存放 U2Net 代码。

名称	说明
data 文件夹	存放数据处理代码
networks 文件夹	存放网络结构代码
options 文件夹	存放参数设置代码
utils 文件夹	存放辅助代码

Infer.py	推理文件，用于获取衣服的 mask
model_surgery.py	生成初始权重，用于训练
train.py	训练代码
README.txt	说明文档

3.4 HR-VITON

存放 HR-VITON 代码。

名称	说明
data 文件夹	用于存放训练数据与测试数据
eval_models 文件夹	存放网络结构代码，用于存放权重
sync_batchnorm 文件夹	sync_batchnorm 类与函数定义
cp_dataset.py	训练数据集处理
cp_dataset_test.py	测试数据集处理
evaluate.py	验证代码
get_norm_const.py	参数设置
get_parse_agnostic.py	人像图片分割代码
network_generator.py	特征融合网络定义代码
network.py	条件生成器代码
test_condition.py	条件生成器测试代码
test_generator.py	特征融合网络测试代码
train_condition.py	条件生成器训练代码
train_generator.py	特征融合网络训练代码
utils.py	辅助代码
preprocessing.md	数据预处理流程
README.md	说明文档（原仓库）
README.txt	说明文档

3.5 FIDscore 文件夹

名称	说明
FIDscore.ipynb	生成 FID 分数

From: Tsing B.L Jiao Q.R LI J.X Sun Z.X Wei Z.M