实时风格迁移实验

1953729 吴浩泽

实验目的

掌握如何使用 TensorFlow 实现实时风格迁移算法中的图像转换网络的推断模块，并进行图像的风格迁移处理。具体包括：

1) 掌握使用 TensorFlow 定义完整网络结构的方法;
2) 掌握使用 TensorFlow 恢复模型参数的方法;
3) 以实时风格迁移算法为例，掌握在 CPU 平台上使用 TensorFlow 进行神经网络推断方法
4) 理解 DLP 高性能算子库集成到 TensorFlow 框架的基本原理;
5) 掌握在 DLP 平台上使用 TensorFlow 对模型进行量化并实现神经网络推断的方法。

实验工作量：约 20 行代码，约需 2 个小时。

背景介绍

图像转换网络

该网络由三个卷积层、五个残差块、两个转置卷积层再接一个卷积层构成。除了输出层，所有非残差卷积层后面都加了批归一化（batchnormalization，BN）[16] 和 ReLU 操作，输出层使用 tanh 函数将输出像素值限定在$[0, 255]$ 范围内；第一层和最后一层卷积使用 9 × 9 卷积核，其它卷积层都使用 3 × 3 卷积核；每个残差块中包含两层卷积。

残差块

图像转换网络中包含了五个残差块，其基本结构：输入 $\mathbf{x}$ 经过一个卷积层，再做 ReLU，然后经过另一个卷积层得到 $\mathcal{F}(\mathbf{x})$，再加上 $\mathbf{x}$ 得到输出 $\mathcal{H}(\mathbf{x}) = \mathcal{F}(\mathbf{x}) + \mathbf{x}$，然后做 ReLU 得到基本块的最终输出 $\mathbf{y}$。当输入 $\mathbf{x}$ 的维度与卷积输出 $\mathcal{F}(\mathbf{x})$ 的维度不同时，需要先对 $\mathbf{x}$ 做恒等变换使二者维度一致，然后再加和。

与常规的卷积神经网络相比，残差块增加了从输入到输出的直连（shortcut connection），其卷积拟合的是输出与输入的差（即残差）。由于输入和输出都做了批归一化，符合正态分布，因此输入和输出可以做减法， $\mathcal{F}(x) = \mathcal{H}(x) - x$。残差网络的优点是对数据波动更灵敏，更容易求得最优解，因此能够改善深层网络的训练。

转置卷积

转置卷积[19] 又可以称为小数步长卷积。输入矩阵 Input-Data 是 2×2 的矩阵，卷积核 Kernel 的大小为 3×3，卷积步长为 1，输出 OutputData 是 4×4 的矩阵

实例归一化

图像转换网络中，每个卷积计算之后激活函数之前都插入了一种特殊的跨样本的批归一化层。该方法由谷歌的科学家在 2015 年提出，它使用多个样本做归一化，将输入归一化到加了参数的标准正态分布上。这样可以有效避免梯度爆炸或消失，从而训练出较深的神经网络。批归一化的计算方法见公式

$$y_{tijk} = \frac{x_{tijk} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}, \quad \mu_i = \frac{1}{HWN}\sum_{t=1}^{N}\sum_{l=1}^{W}\sum_{m=1}^{H} x_{tilm}, \quad \sigma_i^2 = \frac{1}{HWN}\sum_{t=1}^{N}\sum_{l=1}^{W}\sum_{m=1}^{H}(x_{tilm} - \mu_i)^2$$

其中，*x*ti jk 表示输入图像集合中的第 *ti jk* 个元素，*k*、*j* 分别表示其在 H、W 方向的序号，*t* 表示输入图像在集合中的序号，*i* 表示特征通道序号。

批归一化方法是在输入图像集合上分别对 NHW 做归一化以保证数据分布的一致性，而在风格迁移算法中，由于迁移后的结果主要依赖于某个图像实例，所以对整个输入集合做归一化的方法并不适合。2017 年有学者针对实时风格迁移算法提出了实例归一化方法，不同于批归一化，该方法使用如下公式来对 HW 做归一化，从而保持每个图像实例之间的独立，在风格迁移算法上取得了较好的效果，比较显著的提升了生成图像的质量。因此本实验中，用实例归一化方法来替代批归一化方法。

$$y_{tijk} = \frac{x_{tijk} - \mu_{ti}}{\sqrt{\sigma_{ti}^2 + \epsilon}}, \quad \mu_{ti} = \frac{1}{HW}\sum_{l=1}^{W}\sum_{m=1}^{H} x_{tilm}, \quad \sigma_{ti}^2 = \frac{1}{HW}\sum_{l=1}^{W}\sum_{m=1}^{H}(x_{tilm} - \mu_{ti})^2$$

TensorFlow 中模型参数的恢复

在 TensorFlow 中，采用检查点机制（Checkpoint）周期地记录（Save）模型参数等数据并存储到文件系统中，后续当需要继续训练或直接使用训练好的参数做推断时，需要从文件系统中将保存的模型恢复(Restore) 出来。检查点机制由 saver 对象来完成，即在模型训练过程中或当模型训练完成后，使用 saver=tf.train.Saver() 函数来保存模型中的所有变量。当需要恢复模型参数来继续训练模型或者进行预测时，需使用 saver 对象的 restore() 函数，从指定路径下的检查点文件中恢复出已保存的变量。在本实验中，图像转换网络和特征提取网络的参数均已经提前训练好并保存在特定路径下，在使用图像转换网络进行图像预测时，直接使用 restore() 函数将这些模型参数读入程序中并实现实时的风格迁移。

实验环境

本节实验所涉及的硬件平台和软件环境如下：

- 硬件平台：CPU、DLP
- 软件环境：TensorFlow 1.14，Python 编译环境及相关的扩展库，包括 Python 2.7.12，Pillow 4.2.1，Scipy 1.0.0，NumPy 1.16.6、CNML 高性能算子库、CNRT 运行时库

实验内容

由于基于 DLP 的实验平台已经提供了采用高性能库实现的 TensorFlow 框架，所以本

节的主要实验内容是完成风格迁移模型在 DLP 定制版本 TensorFlow 上的运行，并和 CPU 版本的 TensorFlow 进行性能对比。具体实验内容包括：

1. 模型量化：由于 DLP 平台支持定点数据类型运算，为了提升模型处理的效率，先将原始的用 Float32 类型表示的 pb 模型文件量化为用 INT8 类型表示的模型文件。

2. 模型推断：将 pb 模型文件通过 Python 接口在 DLP 平台上运行推断过程，并和第 5 章在 CPU 上运行推断的性能进行对比。

实验步骤

本实验主要包括以下步骤：读取图像、CPU 上实现、DLP 上实现、实验运行与对比等。

读取图像

使用与上一实验相同的方法读取一张图片，如果程序中指定了图片尺寸，就将该图像缩放至指定的尺寸。该部分代码定义在实验环境的 src/utils.py 文件中。

```
1  def get_img(src, img_size=False):
2      #TODO: 使用 scipy.misc 模块读入输入图像 src 并转化成'RGB' 模式, 返回 ndarray 类型数组 img
3      img = scipy.misc.imread(src,mode="RGB")
4      img = scipy.misc.imresize(img,img_size) if img_size else img
5      return img
```

CPU 上实现实时风格迁移

为了在 CPU 上实现实时风格迁移，需要使用图像转换网络对应的 pb 模型文件处理输入图像，得到风格迁移后的输出图像。主要包括实时风格迁移函数和实时风格迁移主函数的定义等。

1. 实时风格迁移函数定义

该定义在 stu_upload/evaluate_cpu.py 文件中。

2. 实时风格迁移主函数定义

该定义同样在 stu_upload/evaluate_cpu.py 文件中。

3. 执行实时风格迁移

在 CPU 上运行命令，实现图像的实时风格迁移。其中，模型文件*.pb 保存在 pb_models/目录下，输入的内容图像保存在 data/train2014_small/目录下，风格迁移后的图像保存在 out/目录下。

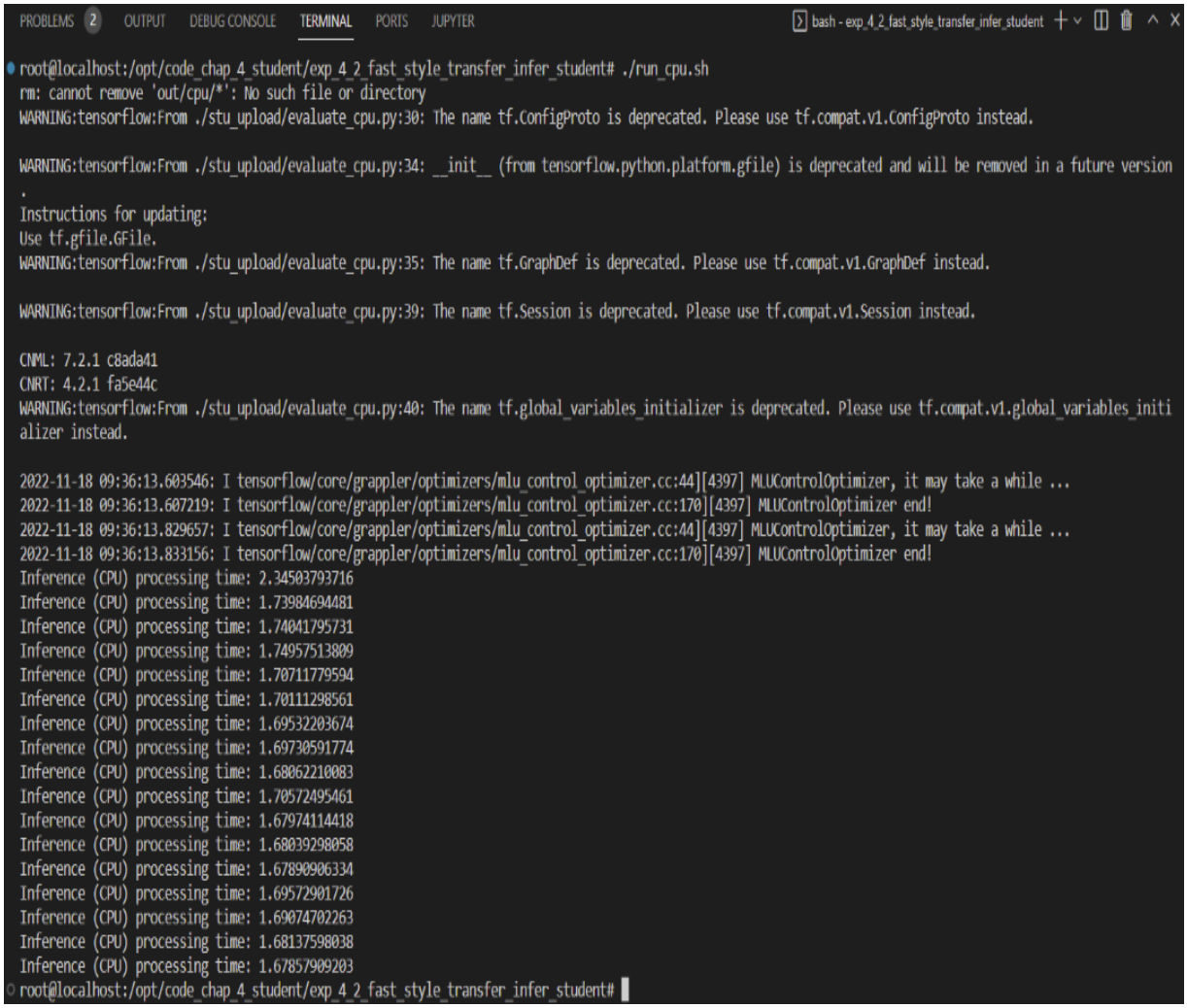
```python
# get img_shape
def ffwd(data_in, paths_out, model, device_t='', batch_size=1):
    assert len(paths_out) > 0
    is_paths = type(data_in[0]) == str
    # TODO: 如果 data_in 是保存输入图像的文件路径，即 is_paths 为 True，则读入第一张图像，由于 pb 模型的输入维度为 1 ×
    # 如果 data_in 是已经读入图像并转化成数组形式的数据，即 is_paths 为 False，则直接获取图像的 shape 特征 img_shape
    img_shape = get_img(data_in[0],(256,256,3)).shape if is_paths else data_in[0].shape

    g = tf.Graph()
    config = tf.ConfigProto(allow_soft_placement=True,
                    inter_op_parallelism_threads=1,
                    intra_op_parallelism_threads=1)
    with g.as_default():
        with tf.gfile.FastGFile(model,'rb') as f:
            graph_def = tf.GraphDef()
            graph_def.ParseFromString(f.read())
            tf.import_graph_def(graph_def, name='')

        with tf.Session(config=config) as sess:
            sess.run(tf.global_variables_initializer())
            input_tensor = sess.graph.get_tensor_by_name('X_content:0')
            output_tensor = sess.graph.get_tensor_by_name('add_37:0')
            batch_size = 1
            # TODO: 读入的输入图像的数据格式为 HWC，还需要将其转换成 NHWC
            batch_shape = (batch_size,) + img_shape
            num_iters = int(len(paths_out)/batch_size)
            for i in range(num_iters):
                pos = i * batch_size
                curr_batch_out = paths_out[pos:pos+batch_size]
                # TODO: 如果 data_in 是保存输入图像的文件路径，则依次将该批次中输入图像文件路径下的 batch_size 张图像读
                # 如果 data_in 是已经读入图像并转化成数组形式的数据，则将该数组传递给 X
                X = [get_img(path,img_shape) for path in data_in[pos:pos+batch_size]] if is_paths else da

                start = time.time()
                # TODO: 使用 sess.run 来计算 output_tensor
                _preds = sess.run(output_tensor,feed_dict={input_tensor:X})
                end = time.time()
                for j, path_out in enumerate(curr_batch_out):
                #TODO: 在该批次下调用 utils.py 中的 save_img() 函数对所有风格迁移后的图片进行存储
                    save_img(path_out,_preds[j])
                delta_time = end - start
                print("Inference (CPU) processing time: %s" % delta_time)
```

实时风格迁移主函数定义：

```
def main():
    parser = build_parser()
    opts = parser.parse_args()
    check_opts(opts)

    if not os.path.isdir(opts.in_path):
        if os.path.exists(opts.out_path) and os.path.isdir(opts.out_path):
            out_path = os.path.join(opts.out_path,os.path.basename(opts.in_path))
        else:
            out_path = opts.out_path
        # 执行风格迁移预测，输入图像为 opts.in_path，转换后的图像为 out_path，模型文件路径为 opts.model
        ffwd_to_img(opts.in_path, out_path, opts.model, device=opts.device)
    else:
        files = list_files(opts.in_path)
        full_in = [os.path.join(opts.in_path,x) for x in files]
        full_out = [os.path.join(opts.out_path,x) for x in files]
        if opts.allow_different_dimensions:
            ffwd_different_dimensions(full_in, full_out, opts.model,
                    device_t=opts.device, batch_size=opts.batch_size)
        else :
            # 执行风格迁移预测，输入图像的保存路径为 full_in，转换后的图像为 full_out，模型文件路径为 opts.model
            ffwd(full_in, full_out, opts.model, device_t=opts.device, batch_size=opts.batch_size)
```

执行实时风格迁移,在 CPU 上运行如下命令：

```
root@localhost:/opt/code_chap_4_student/exp_4_2_fast_style_transfer_infer_student# ls
data  fppb_to_intpb  main_exp_4_2.py  out  pb_models  readme.txt  run_cpu.sh  run_mlu.sh  src  stu_upload
root@localhost:/opt/code_chap_4_student/exp_4_2_fast_style_transfer_infer_student# cat ./run_cpu.sh
rm out/cpu/*

python ./stu_upload/evaluate_cpu.py --model pb_models/udnie.pb --in-path data/train2014_small/  --out-path
root@localhost:/opt/code_chap_4_student/exp_4_2_fast_style_transfer_infer_student#
```



输出结果为：

```
CNML: 7.2.1 c8ada41
CNRT: 4.2.1 fa5e44c
Inference (CPU) processing time: 2.34503793716
Inference (CPU) processing time: 1.73984694481
Inference (CPU) processing time: 1.74041795731
Inference (CPU) processing time: 1.74957513809
Inference (CPU) processing time: 1.70711779594
Inference (CPU) processing time: 1.70111298561
Inference (CPU) processing time: 1.69532203674
Inference (CPU) processing time: 1.69730591774
Inference (CPU) processing time: 1.68062210083
Inference (CPU) processing time: 1.70572495461
Inference (CPU) processing time: 1.67974114418
Inference (CPU) processing time: 1.68039298058
Inference (CPU) processing time: 1.67890906334
Inference (CPU) processing time: 1.69572901726
Inference (CPU) processing time: 1.69074702263
Inference (CPU) processing time: 1.68137598038
Inference (CPU) processing time: 1.67857909203
```

DLP 上实现实时风格迁移

在 DLP 上实现实时风格迁移的实验步骤分为：模型量化和模型推断。

1. 模型量化

已经提前训练好的图像转换网络的数据类型为 Float32，需要经过量化后才可以在 DLP 上运行。在 fppb_to_intpb 目录下运行以下命令，使用量化工具完成对模型的量化，生成新模型 udnie_int8.pb。

```
python fppb_to_intpb.py udnie_int8.ini
```

2. 模型推断

通过 DLP 定制的 TensorFlow 版本（其中大部分风格迁移的算子都通过 DLP 的高性能库支持）完成风格迁移模型的前向推断。为了使上层用户不感知底层硬件的迁移，定制的 TensorFlow 维持了上层的 Python 接口，用户可以通过 session config 配置 DLP 运行的相关参数以及使用相关接口进行量化。具体的运行时配置信息如图 4.13 所示，可以设置运行的核数和使用的数据类型等信息。在配置完 DLP 硬件相关的参数后，推断时模型的算子可自动运行在 DLP 上。

在运行完成后，统计 sess.run() 前后的运行时间，并与在 CPU 上的运行时间进行对比。

```python
# get img_shape
def ffwd(data_in, paths_out, model, device_t='', batch_size=1):
    assert len(paths_out) > 0
    is_paths = type(data_in[0]) == str
    # TODO: 如果 data_in 是保存输入图像的文件路径，即 is_paths 为 True，则读入第一张图像，由于 pb 模型的输入维度为 1 ×
    # 如果 data_in 是已经读入图像并转化成数组形式的数据，即 is_paths 为 False，则直接获取图像的 shape 特征 img_shape
    img_shape = get_img(data_in[0], (256, 256, 3)).shape if is_paths else data_in[0].shape
    g = tf.Graph()
    # setting mlu configurations
    config = tf.ConfigProto(allow_soft_placement=True,
                    inter_op_parallelism_threads=1,
                    intra_op_parallelism_threads=1)
    config.mlu_options.data_parallelism = 1
    config.mlu_options.model_parallelism = 1
    config.mlu_options.core_num = 16 # 1 4 16
    config.mlu_options.core_version = "MLU270"
    config.mlu_options.precision = "int8"
    config.mlu_options.save_offline_model = False
    with g.as_default():
        with tf.gfile.FastGFile(model,'rb') as f:
            graph_def = tf.GraphDef()
            graph_def.ParseFromString(f.read())
            tf.import_graph_def(graph_def, name='')

        with tf.Session(config=config) as sess:
            sess.run(tf.global_variables_initializer())
            input_tensor = sess.graph.get_tensor_by_name('X_content:0')
            output_tensor = sess.graph.get_tensor_by_name('add_37:0')
            batch_size = 1
            # TODO: 读入的输入图像的数据格式为 HWC，还需要将其转换成 NHWC
            batch_shape = (batch_size,) + img_shape
            num_iters = int(len(paths_out)/batch_size)
            for i in range(num_iters):
                pos = i * batch_size
                curr_batch_out = paths_out[pos:pos+batch_size]
                # TODO: 如果 data_in 是保存输入图像的文件路径，则依次将该批次中输入图像文件路径下的 batch_size 张图像读
                # 如果 data_in 是已经读入图像并转化成数组形式的数据，则将该数组传递给 X
                X = [get_img(path, img_shape) for path in data_in[pos:pos+batch_size]] if is_paths else da

                start = time.time()
                # TODO: 使用 sess.run 来计算 output_tensor
                _preds = sess.run(output_tensor, feed_dict={input_tensor:X})
                end = time.time()
                for j, path_out in enumerate(curr_batch_out):
                    #TODO: 在该批次下调用 utils.py 中的 save_img() 函数对所有风格迁移后的图片进行存储
                    save_img(path_out, _preds[j])
                delta_time = end - start
                print("Inference (MLU) processing time: %s" % delta_time)
```

运行 run_mlu.sh

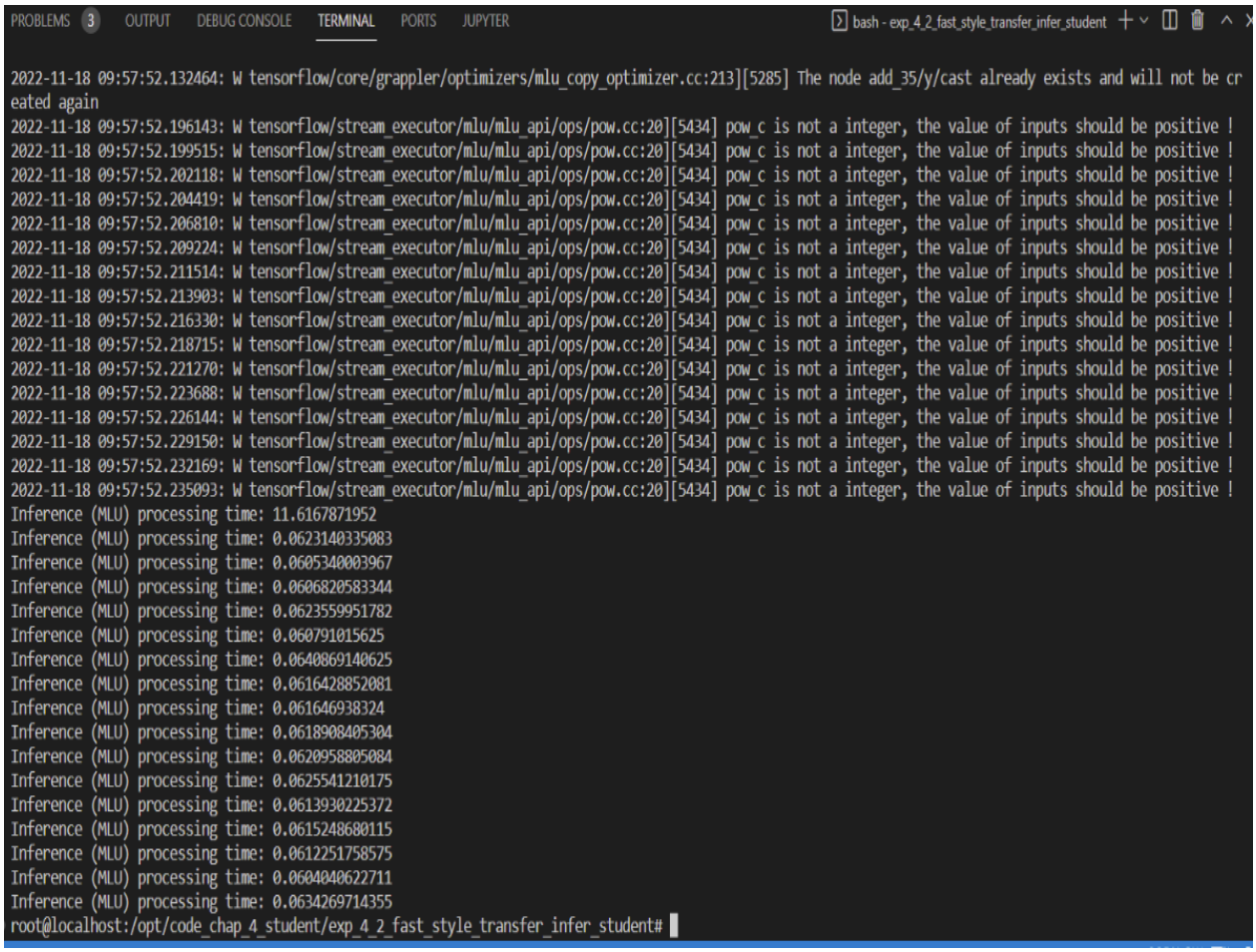
```
# export CNRT_PRINT_INFO=ON
# export CNRT_GET_HARDWARE_TIME=ON

rm out/mlu/*

python ./stu_upload/evaluate_mlu.py --model pb_models/udnie_int8.pb --in-path data/train2014_small/ --out-
```

输出结果为：

```
rm: cannot remove 'out/mlu/*': No such file or directory
WARNING:tensorflow:From ./stu_upload/evaluate_mlu.py:30: The name tf.ConfigProto is deprecated. Please us

WARNING:tensorflow:From ./stu_upload/evaluate_mlu.py:40: __init__ (from tensorflow.python.platform.gfile)
Instructions for updating:
Use tf.gfile.GFile.
WARNING:tensorflow:From ./stu_upload/evaluate_mlu.py:41: The name tf.GraphDef is deprecated. Please use t

WARNING:tensorflow:From ./stu_upload/evaluate_mlu.py:45: The name tf.Session is deprecated. Please use tf

CNML: 7.2.1 c8ada41
CNRT: 4.2.1 fa5e44c
CNML: 7.2.1 c8ada41
2022-11-18 09:57:51.821520: I tensorflow/stream_executor/mlu/mlu_stream_executor.cc:470][5285] MLU OP Cac
CNML: 7.2.1 c8ada41
2022-11-18 09:57:51.821664: I tensorflow/core/common_runtime/mlu/mlu_device.cc:396][5285] __LOG_MLU__, Cu
2022-11-18 09:57:51.821689: I tensorflow/core/common_runtime/mlu/mlu_device.cc:115][5285] __LOG_MLU__, sa
2022-11-18 09:57:51.821706: I tensorflow/core/common_runtime/mlu/mlu_device.cc:121][5285] __LOG_MLU__, co
2022-11-18 09:57:51.821723: I tensorflow/core/common_runtime/mlu/mlu_device.cc:123][5285] __LOG_MLU__, co
WARNING:tensorflow:From ./stu_upload/evaluate_mlu.py:46: The name tf.global_variables_initializer is depr

2022-11-18 09:57:51.839492: I tensorflow/core/grappler/optimizers/mlu_control_optimizer.cc:44][5285] MLUC
```

```
2022-11-18 09:57:51.839492: I tensorflow/core/grappler/optimizers/mlu_control_optimizer.cc:44][5285] MLUCo
2022-11-18 09:57:51.841425: I tensorflow/core/grappler/optimizers/mlu_control_optimizer.cc:170][5285] MLUC
2022-11-18 09:57:51.960418: I tensorflow/core/grappler/optimizers/mlu_control_optimizer.cc:44][5285] MLUCo
2022-11-18 09:57:51.962327: I tensorflow/core/grappler/optimizers/mlu_control_optimizer.cc:170][5285] MLUC
2022-11-18 09:57:52.101254: I tensorflow/compiler/tf2mlu/convert/convert_graph.cc:654][5285] MLU fusion no
2022-11-18 09:57:52.131822: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.131929: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.131949: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.131967: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.131982: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.131997: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132014: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132031: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132055: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132069: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132085: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132105: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132125: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132141: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132159: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132244: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132267: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132282: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132296: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132313: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132327: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132344: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
```

```
2022-11-18 09:57:52.132359: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132374: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132388: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132403: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132419: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132435: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132449: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.132464: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The noc
2022-11-18 09:57:52.196143: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
2022-11-18 09:57:52.199515: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
2022-11-18 09:57:52.202118: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
2022-11-18 09:57:52.204419: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
2022-11-18 09:57:52.206810: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
2022-11-18 09:57:52.209224: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
2022-11-18 09:57:52.211514: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
2022-11-18 09:57:52.213903: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
2022-11-18 09:57:52.216330: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
2022-11-18 09:57:52.218715: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
2022-11-18 09:57:52.221270: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
2022-11-18 09:57:52.223688: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
2022-11-18 09:57:52.226144: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
2022-11-18 09:57:52.229150: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
2022-11-18 09:57:52.232169: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
2022-11-18 09:57:52.235093: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a i
Inference (MLU) processing time: 11.6167871952
Inference (MLU) processing time: 0.0623140335083
Inference (MLU) processing time: 0.0605340003967
Inference (MLU) processing time: 0.0606820583344
Inference (MLU) processing time: 0.0623559951782
Inference (MLU) processing time: 0.060791015625
Inference (MLU) processing time: 0.0640869140625
Inference (MLU) processing time: 0.0616428852081
Inference (MLU) processing time: 0.061646938324
Inference (MLU) processing time: 0.0618908405304
Inference (MLU) processing time: 0.0620958805084
Inference (MLU) processing time: 0.0625541210175
Inference (MLU) processing time: 0.0613930225372
Inference (MLU) processing time: 0.0615248680115
Inference (MLU) processing time: 0.0612251758575
Inference (MLU) processing time: 0.0604040622711
Inference (MLU) processing time: 0.0634269714355
```


PROBLEMS 3
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
JUPYTER
bash - exp_4_2_fast_style_transfer_infer_student
2022-11-18 09:57:52.132464: W tensorflow/core/grappler/optimizers/mlu_copy_optimizer.cc:213][5285] The node add_35/y/cast already exists and will not be created again
2022-11-18 09:57:52.196143: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
2022-11-18 09:57:52.199515: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
2022-11-18 09:57:52.202118: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
2022-11-18 09:57:52.204419: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
2022-11-18 09:57:52.206810: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
2022-11-18 09:57:52.209224: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
2022-11-18 09:57:52.211514: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
2022-11-18 09:57:52.213903: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
2022-11-18 09:57:52.216330: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
2022-11-18 09:57:52.218715: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
2022-11-18 09:57:52.221270: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
2022-11-18 09:57:52.223688: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
2022-11-18 09:57:52.226144: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
2022-11-18 09:57:52.229150: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
2022-11-18 09:57:52.232169: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
2022-11-18 09:57:52.235093: W tensorflow/stream_executor/mlu/mlu_api/ops/pow.cc:20][5434] pow_c is not a integer, the value of inputs should be positive !
Inference (MLU) processing time: 11.6167871952
Inference (MLU) processing time: 0.0623140335083
Inference (MLU) processing time: 0.0605340003967
Inference (MLU) processing time: 0.0606820583344
Inference (MLU) processing time: 0.0623559951782
Inference (MLU) processing time: 0.060791015625
Inference (MLU) processing time: 0.0640869140625
Inference (MLU) processing time: 0.0616428852081
Inference (MLU) processing time: 0.061646938324
Inference (MLU) processing time: 0.0618908405304
Inference (MLU) processing time: 0.0620958805084
Inference (MLU) processing time: 0.0625541210175
Inference (MLU) processing time: 0.0613930225372
Inference (MLU) processing time: 0.0615248680115
Inference (MLU) processing time: 0.0612251758575
Inference (MLU) processing time: 0.0604040622711
Inference (MLU) processing time: 0.0634269714355
root@localhost:/opt/code_chap_4_student/exp_4_2_fast_style_transfer_infer_student#