

# 数字图像处理

## 基于昇腾 CANN 的推理应用开发——黑白图像上色

信息与电子工程学院 信息工程

2023 年 6 月 10 日

### 1 实验内容及目标

#### 1.1 实验内容

本实验主要介绍了基于昇腾 CANN 平台的黑白图像上色应用的开发方法。该应用选用的是 `colorization` 模型，以黑白图片作为输入，对黑白图像上色，并把预测的结果保存为彩色图片。通过本实验，您可以了解如何在昇腾平台上实现一个基于 `colorization` 模型的推理应用。

#### 1.2 实验目标

1. 掌握一个基于昇腾 CANN 平台的推理应用的基本结构。
2. 理解 `colorization` 模型的网络结构及其数据预处理/后处理方式。

### 2 网络结构

黑白图像上色模型原理如下图所示，对物件（包括背景色）（L 通道代表的灰度图）使用卷积运算提取特征，然后同样用卷积进行分类，从而尝试给出对灰度图片每个像素点的色彩预期（ab 通道）。将 ab 通道的值 `resize` 到原始图片宽高并与 L 通道叠加后，转为 RGB 图片即可得到彩色图像。

其中模型的输入为 Lab 图像色域中分离出的 L 通道的数据，数据 `shape` 为 `(224, 224, 1)`，模型的输出为 ab 通道的数据，数据 `shape` 为 `(56, 56, 2)`。

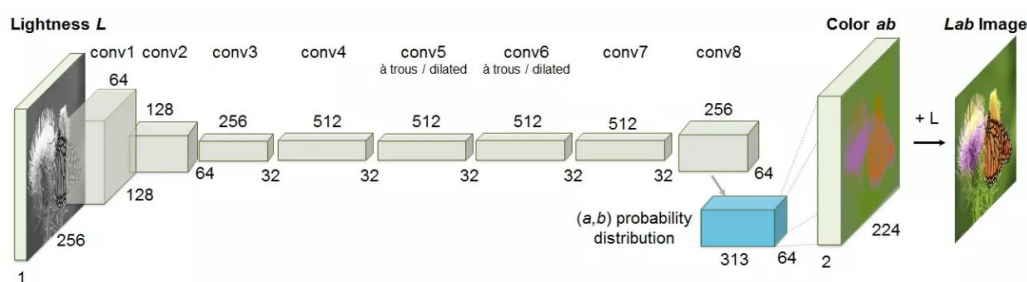


图 1: 网络结构

### 3 代码实现

---

```
import sys
import os
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

home_path = !echo ${HOME}
sys.path.append(os.path.join(home_path[0] , "jupyter-notebook/"))
print('System init success.')
```

# atlas\_utils 是本团队基于 pyACL 封装好的一套工具库, 如果您也想引用的话, 请首先将  
[https://gitee.com/ascend/samples/tree/master/python/common/atlas\\_utils](https://gitee.com/ascend/samples/tree/master/python/common/atlas_utils)  
 # 这个路径下的代码引入您的工程中

```
from atlas_utils.acl_resource import AclResource
from constants import *
from acl_model import Model
```

# 创建一个 AclResource 类的实例

```
acl_resource = AclResource()
#AscendCL 资源初始化 (封装版本)
acl_resource.init()
```

# 上方“init”方法具体实现 (仅供参考), 请阅读“init()”方法, 观察初始化和运行时资源申请的逻辑

```
def init(self):
    """
    Init resource
    """
    print("init resource stage:")
    ret = acl.init()
    utils.check_ret("acl.init", ret)
    # 指定用于运算的 Device
    ret = acl.rt.set_device(self.device_id)
    utils.check_ret("acl.rt.set_device", ret)
    print("Set device n success.")

    # 显式创建一个 Context
    self.context, ret = acl.rt.create_context(self.device_id)
    utils.check_ret("acl.rt.create_context", ret)

    # 创建一个 Stream
    self.stream, ret = acl.rt.create_stream()
    utils.check_ret("acl.rt.create_stream", ret)
```

```

# 获取当前昇腾 AI 软件栈的运行模式
#0: ACL_DEVICE, 表示运行在 Device 的 Control CPU 上或开发者版上
#1: ACL_HOST, 表示运行在 Host CPU 上
self.run_mode, ret = acl.rt.get_run_mode()
utils.check_ret("acl.rt.get_run_mode", ret)

print("Init resource success")

# 请阅读下方代码, 观察释放运行时资源的详细操作步骤
def __del__(self):
    print("acl resource release all resource")
    resource_list.destroy()

# 调用 acl.rt.destroy_stream 接口释放 Stream
if self.stream:
    acl.rt.destroy_stream(self.stream)
    print("acl resource release stream")

# 调用 acl.rt.destroy_context 接口释放 Context
if self.context:
    acl.rt.destroy_context(self.context)
    print("acl resource release context")

# 调用 acl.rt.destroy_context 接口释放 Context
acl.rt.reset_device(self.device_id)
acl.finalize()
print("Release acl resource success")

```

---

## 4 实验结果

黑白图像上色结果如下:



(a) 黑白图像



(b) 上色图像

图 2: 黑白图像上色