·实验教学与教学实验·

基于华为昇腾 Atlas 200DK 的黑白图片上色实验

武　畅，夏玉林，杨小漫，王　宏

（电子科技大学 信息与通信工程学院，成都 611731）

摘要：随着人工智能(AI)技术的发展，大学教学中相关课程和技术应用的比重不断上升，使学生对人工智能的实验需求爆发式增长。但是，目前的人工智能实验体系并不完善，实验形式主要以软件实现为主，缺乏基于硬件的平台和实验项目，与实际工程应用和行业需求脱节。结合 CDIO(构思、设计、实现、运作)模式，针对实验中的深度学习网络推理速度缓慢、耗时过长等问题，提出了以华为昇腾 Atlas 200DK 硬件平台为基础，利用深度学习网络，完成黑白图像上色的硬件人工智能实验。通过该实验，学生可以学习人工智能的原理、神经网络的结构和推理过程，搭建软件开发环境和配置硬件系统，实现软硬件协同的神经网络的部署，完成功能和相关指标测试。该实验有助于学生立足于实际应用需求理解神经网络的运行机理，掌握其在实际硬件平台上的应用方法，从而全面提升软硬件的工程实践能力。

关　键　词：人工智能；Atlas 200DK；卷积神经网络；图像上色；产教融合

中图分类号：G642; TN85　　**文献标志码：**A　　DOI: 10.12179/1672-4550.20240537

Black-and-White Image Coloring Experiment Based on Huawei's Ascend Atlas 200DK

WU Chang, XIA Yulin, YANG Xiaoman, WANG Hong

（School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China）

Abstract: With the development of artificial intelligence (AI) technologies, the proportion of related courses and technological applications in university curricula has been increasing, leading to a surge in students' demand for AI experimentation. However, the current AI experimental systems are not well-established, with most experiments being software-based and lacking hardware platforms and projects, which results in a disconnect from real-world engineering applications and industry needs. This experimental instruction integrates the CDIO (conceive, design, implement, operate) model, and to address issues such as slow inference speeds and long processing times in deep learning networks during experiments, an AI experiment based on the Huawei Ascend Atlas 200DK hardware platform is proposed, which utilizes deep learning networks to perform colorization of grayscale images. Through this experiment, students can learn about the principles of AI, the structure and inference processes of neural networks, set up software development environments, configure hardware systems, and deploy neural networks that integrate both software and hardware. The experiment also involves the completion of functional and performance testing. This hands-on experience helps students understand the operational mechanisms of neural networks from a practical application perspective, master their application methods on actual hardware platforms, and thereby enhance their overall engineering and practical skills in both software and hardware.

Key words: artificial intelligence; Atlas 200DK; convolutional neural networks; image colorization; industry-education integration

在现代信息社会的背景下，人工智能(AI)正在快速渗透进各个行业和领域，对我们的生活和社会结构产生着深远而独特的影响。作为一个前沿和多元化的研究领域，人工智能在教育界也受

收稿日期：2024−10−22

基金项目：教育部−华为“智能基座”产教融合协同育人项目；电子科技大学研究生教研教改重点项目(JYJG2021401)；电子科技大学本科实验教学和教学实验室建设研究项目(BK-SYJY-202423)；人工智能技术赋能本科教学教改项目(2024AIXM007)。

作者简介：武畅，博士，副教授，主要从事信号与信息处理、通信系统、软件无线电与人工智能方面的研究。E-mail: changwu@uestc.edu.cn

到了越来越多的关注。教育专家和学者普遍认为，通过整合人工智能的原理和应用技术，不仅可以增强学生的学习动机，还能促进他们的创新思维，并为其未来职业生涯打下坚实的基础[1]。

然而，在人工智能的实验教学中，传统的软件实验难以让学生深入理解 AI 系统的底层工作机制，也无法满足实践教学、工程应用和行业需求。更重要的是，这样的教学模式往往忽视了培养学生在硬件平台上的实践能力，因此难以符合当今教育目标的多元需求[2]。鉴于这些挑战和局限性，关注如何通过硬件平台实施人工智能实验教学，为学生提供一个更为贴近实际、更富有操作性的实验环境，帮助他们更全面地掌握人工智能的各个方面，是新一代人工智能教学的重点[3]。

因此，本文开发了基于华为昇腾 Atlas 200DK 的黑白图像上色实验，通过人工智能理论学习、黑白图像上色算法掌握以及硬件平台的使用，同时结合 CDIO 模式，完成整个实验过程，从而增强学生对实际工程中的人工智能应用的理解，提升工程实践能力，取得更有效和深入的教学效果。

1　实验教学

本实验教学结合 CDIO 模式，旨在通过项目驱动的方式，培养学生的工程实践能力和创新思维。CDIO 模式覆盖了从概念构思到产品实现的整个工程生命周期，强调学生的主动学习和实践操作。实验内容与 CDIO 模式的 4 个层面紧密对应，确保学生能够在实践中全面提升自己的能力。

1.1　实验教学与 CDIO 模式的融合

1）创意构思（conceive）

学生识别和理解黑白图像上色的问题，进行需求分析和概念设计。此阶段对应实验的引言和理论学习部分，学生通过调研和分析，发现问题的本质、需求和目标，并提出可能的解决方案。

2）方案设计（design）

在构思阶段的基础上，学生进行详细的设计工作，包括方案的可行性分析、系统设计和技术方案的优化。此阶段对应实验的模型选择和网络结构设计部分，学生思考问题并将设计转化为具体的技术实现方案。

3）技术实现（implement）

学生将设计转化为实际可操作的产品或系统。此阶段对应实验的模型训练和环境部署部分，学生通过实验、原型制作、技术实现等，解决具体问题，并对设计进行调整和改进。

4）运作创新（operate）

学生对产品或系统进行测试、优化，确保其功能的稳定性和持续性。此阶段对应实验的推理应用开发和结果分析部分，学生将对模型进行测试和优化，确保其在实际应用中的有效性。

1.2　实验教学规范

1）实验准备

学生需提前阅读实验手册，确保理解实验目的、原理和步骤。实验前，检查小组分配的设备，包括 Atlas 200DK 开发板和 MindStudio 环境，确保所有配置正确无误。

2）实验执行

按照实验手册指导，逐步执行实验操作。如遇问题，记录并寻求指导。

3）实验记录

详细记录实验步骤、参数设置、观察结果和数据。小组应有统一的记录格式，确保信息的准确性和可追溯性。

4）实验报告

实验结束后，撰写实验报告，包括摘要、引言、原理、材料与方法、结果、讨论和结论。报告应由个人独立完成。

以下给出了基础实验的案例，在实际过程中，学生需参照基础实验，自行选择并设计优化模型。

2　黑白图像上色

黑白图像上色是指给黑白图像添加彩色信息的过程，它在影视处理、老照片修复等方面有着广泛的应用场景。黑白图像上色的研究意义在于提高图像的视觉效果和信息量，增强图像的真实感和美感，以及探索图像的语义信息和颜色分布规律[4]。然而，黑白图像上色也是一个具有挑战性的问题，因为一幅黑白图像可能对应多种合理的彩色图像，而且缺乏颜色信息会导致图像的语义信息不完整[5]。因此，如何利用深度学习的方法，根据黑白图像的内容和上下文，自动生成真实和多样的彩色图像，是本实验的主要研究目标[6]。

由于黑白图像上色存在颜色的多模态性和颜色空间量化的问题，所以实现黑白图像彩色化有一定的难度。并且由于图像处理对算力的要求较

高，也会有一系列的问题[7]。主要在于以下两个方面：一是运行的速度慢，实现黑白图像上色，通常需要通过网络或服务器进行处理，这会增加传输和计算的时间，导致上色效果不能及时呈现；二是生成的图像质量低，实现黑白图像上色，通常需要依赖于预训练的模型或算法，这些模型或算法可能不能适应所有场景和需求，导致上色效果不自然或不准确。

为了获得更好的黑白图像上色效果，并使其更好地应用实现，实验采用 Atlas 200DK 开发板[8]实现黑白图像上色。本实验的优势有以下两点：一是实时性更好，开发板可以利用硬件加速和优化的算法，实现对黑白图像的实时上色；二是应用更加灵活，开发板可以根据不同的需求和场景，选择不同的上色模型和参数。实验采用卷积神经网络(convolutional neural network，CNN)，搭载于 Atlas 200DK 设备，使用华为昇腾算子包 CANN，用华为模型转换工具 ATC 命令对 colorization 网络模型进行转换[9]。通过实验，可以让学生更好地理解黑白图像上色的原理，并学会使用 Atlas 200DK 设备进行推理开发。

3 Atlas 200DK 开发平台

Atlas 200DK 开发者套件是以 Atlas 200 AI 加速模块为核心的终端类产品。Atlas 200 AI 加速模块是一款高性能的 AI 智能计算模块，集成了昇腾 310 AI 处理器(Ascend 310 AI 处理器)，可以实现数据分析与推理计算。昇腾 310 AI 处理器是一款华为专门为图像识别、推理计算及机器学习等领域设计的高性能、低功耗 AI 芯片[10]。

其特点主要体现在以下 4 个方面。

1) 高性能。Atlas 200DK 开发板具有 22 TOPS INT8 的 AI 算力，支持 H.264/H.265 的硬件编解码能力，可以实现对黑白图像的实时上色。

2) 高集成度。Atlas 200DK 开发板集成了丰富的外设接口，方便开发者简捷接入各种设备和传感器。

3) 易用性。Atlas 200DK 开发板配套了 MindSpore Studio 开发环境，可以全自动管理离线模型并提供仿真环境。

4) 灵活性。Atlas 200DK 开发板支持多种 AI 应用场景，如图像分类、人脸检测、风格迁移等，可以根据不同的需求和场景，选择不同的模型和参数。

通过案例指导和设备手册，学生就可以实现推理平台的环境设置和平台搭建，进一步实现黑白图片上色的实验。

为了充分体现软硬件结合及工程实现，实验采用 Atlas 200DK 开发板作为硬件推理平台，采用神经网络实现黑白图像上色算法，获取黑白图像上色的原始网络模型及其对应的权重文件，利用 MindSpore Studio 转换模型为 om 格式，构建 AI 应用工程，查看黑白图像上色的效果。通过实验，一方面，学生可以掌握 AI 应用开发的基本流程和方法，如获取和转换模型、构建和编译工程等，提高 AI 技能水平；另一方面，学生可以了解黑白图像上色的原理和算法，探索不同的上色模型和参数，尝试优化和改进效果，增强创新意识和能力。

本实验电脑开发端选取 ubuntu 18.04 系统，开发环境为 x86 架构，运行环境为 arm 架构。采用 Atlas 200DK 开发板搭载推理平台，采用 CANN 6.0.0alpha001 版本，选用 MindStudio5.0.0 版本。Atlas 200DK 系统框图[8]如图 1 所示，Atlas 200DK 的实体图如图 2 所示。

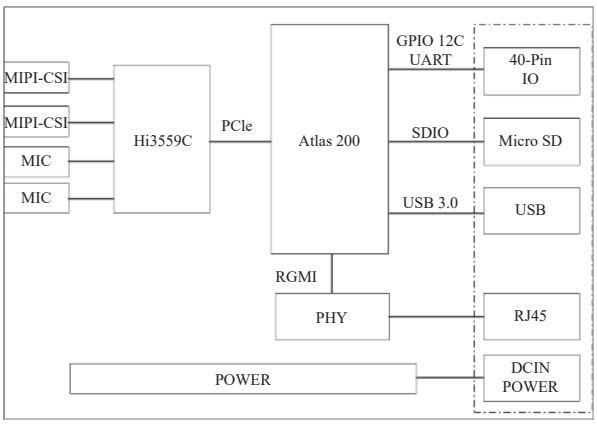

图 1 Atlas 200DK 系统架构

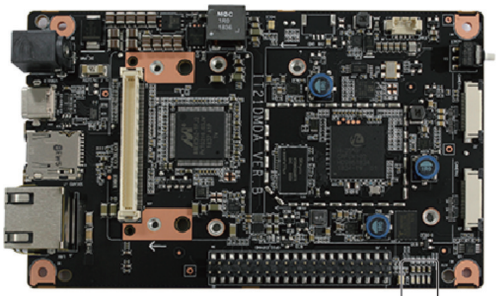

图 2 Atlas 200DK 实体图

4　实验原理

本节介绍相关的实验原理，并说明实验过程。

4.1　卷积神经网络

CNN 是一种在计算机视觉和图像处理领域取得卓越成果的深度学习模型[11]。其设计受到生物视觉系统的启发，通过层次化的特征提取与表示学习，实现了在图像领域的自动特征抽取与分类。

CNN 的核心思想在于卷积层与池化层的结合，以及通过多层堆叠的方式进行逐层抽象。卷积层使用卷积操作对输入数据进行特征提取，通过卷积核（filter）在局部区域上的滑动，实现对不同特征的感知。池化层则通过降采样的方式，减少特征图的尺寸与参数量，同时保留主要特征。

CNN 的网络结构一般由卷积层、激活函数、池化层和全连接层构成，具体结构如图 3 所示[12]。其中，激活函数（如 ReLU）引入非线性因素，增强了网络的表达能力。全连接层在特征抽取后，进行高层次的特征组合与分类。

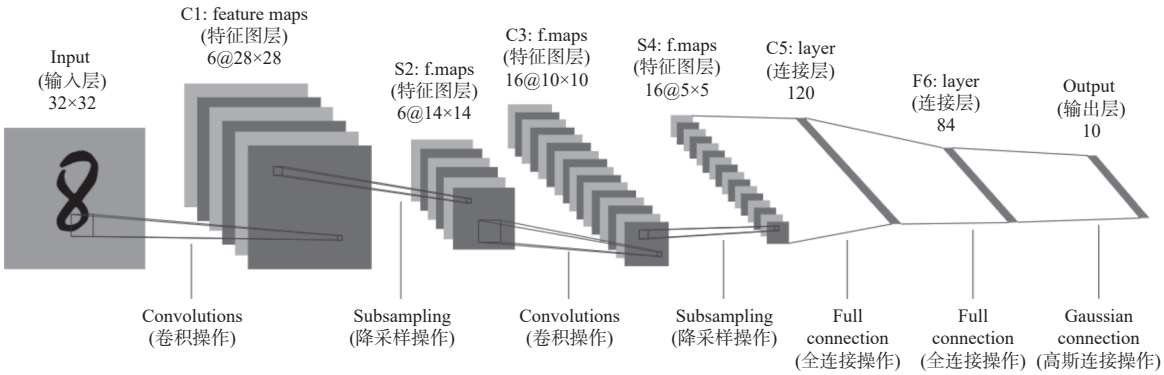

图 3　卷积神经网络结构图

此外，CNN 还广泛使用了参数共享的机制，即同一卷积核在不同区域共享参数，大大减少了网络的参数量，提升了模型的训练效率。同时，通过堆叠多个卷积层，网络能够从低级特征逐步提取更高级的语义特征，实现更精确的分类和识别。

4.2　Colorization 模型

本实验中进行推理的 Colorization 模型由文献[13] 提出，从分类任务的视角出发，设计了一套基于卷积神经网络结构的深度学习网络模型。最终形成可能的概率分布，根据概率分布得到颜色分布情况，将颜色分布叠加到原黑白图像，实现对黑白图像的上色。Colorization 模型的网络结构如图 4 所示。

从黑白图像上色模型原理图中看出，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（a，b 通道）。将 a，b 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 L 通道的数据（224，224，1），输出为 a，b 通道的数据（56，56，2）。

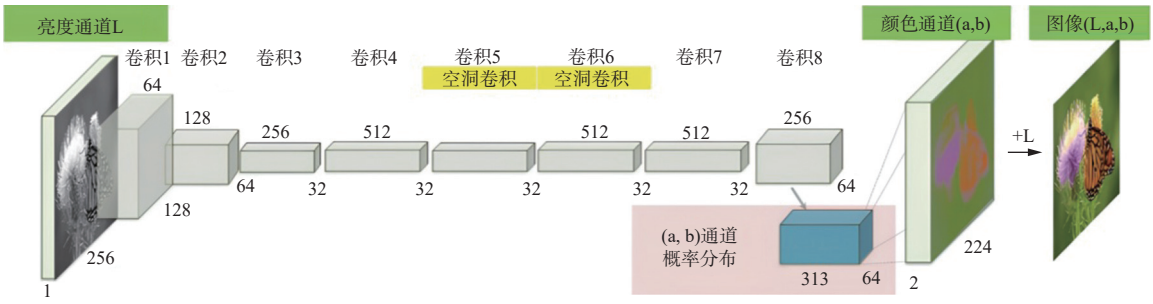

图 4　Colorization 模型结构图

该模型一共有 8 层卷积，每层卷积块由许多卷积 conv2d 块和激活函数 relu 组成，第 5 层和第 6 层用到了空洞卷积。输入图片的 L 通道，使用一个 CNN 预测对应的 a，b 通道取值的概率分布，最后转化为 RGB 图像结果。整个模型最终包含两个部分，即用于预测所有像素颜色分布的卷积神经网络和用于产生最终预测图像的退火平均操作。其中，退火平均操作是一种在模型训练过程中用于平滑概率分布的技术，它通过逐步降低温度参数来减少模型预测的波动性，从而提高模型的稳定性和预测准确性。虽然整个系统不能从严格意义上实现端到端训练，但是可以注意到退火操作是对每个像素单独作用的，只有一个参数，可以作为 CNN 的前向转播的一部分实现。

4.3　神经网络搭建

本实验的网络模型主要涉及的是 CNN，为了方便学生理解，介绍整体网络实现过程。首先，我们设计了一个基于卷积层的编码器–解码器架构，其中编码器负责从输入的灰度图像中提取特征表示，而解码器则负责将这些特征转化为彩色图像。编码器部分采用多层卷积和池化层来逐渐提取图像的语义信息，同时减少图像的分辨率。解码器部分则使用反卷积和上采样操作，逐步重建出高分辨率的彩色图像。

在训练过程中，采用了大量的彩色图像和对应的灰度图像作为训练数据集。实验使用均方误差(mean square error，MSE)作为损失函数，用于衡量模型生成的彩色图像与真实彩色图像之间的差异。为了提高训练稳定性和加速收敛，我们还采用了一种渐进式的训练策略，即先采用低分辨率的输入图片，通过网络结构逐步增加分辨率，以便模型可以逐步学习图像的细节和结构。在训练完成后，保存训练好的模型参数，为后面的运行开发板作准备。

5　实验方案

实验所完成的是基于 Atlas 200DK 的黑白图片上色，因此实验中既包括了深度学习实现黑白图片上色，也包括搭建开发环境和运行环境实现推理，实验整体方案如图 5 所示。

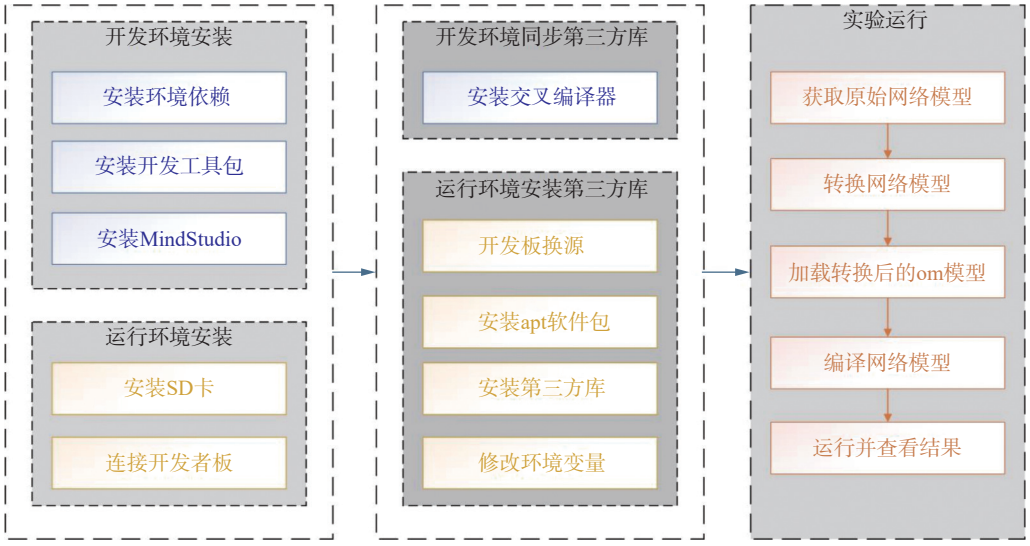

图 5　实验整体方案图

5.1　环境部署

本实验首先要完成环境的搭建，采用开发环境和运行环境分设的方式进行。开发环境即用户的 PC 机，搭配 Ubuntu x86 操作系统，主要用来进行算子开发、应用开发、模型开发及模型转换等功能的开发及编译。运行环境就是 Atlas 200DK 开发板，需要搭载 Ubuntu Aarch64 操作系统，将开发环境编译得到的应用程序可执行文件及依赖文件上传到 Atlas 200DK 侧，进行离线推理的执行[14]。部署场景如图 6 所示。

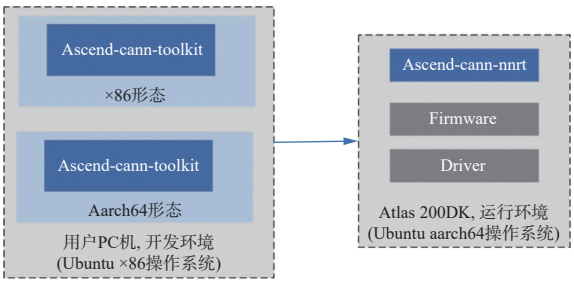

图 6　部署场景架构图

首先安装开发环境，包含以下 6 个步骤：

1）配置用户权限，给普通用户配置 root 权限，在终端中输入 su root 命令，切换 root 用户，然后给 sudoer 文件配置写权限，并打开该文件，添加 ascend ALL=(ALL:ALL) ALL，保存并配置；

2）更新源文件，在 root 用户下输入 vi /etc/apt/sources.list 命令打开 apt 源文件，然后替换源文件后保存，输入 apt-get update 命令，完成源更新；

3）安装相关的 apt 依赖，输入 sudo apt-get install -y gcc make cmake…命令，安装开发环境中套件包所依赖的一些 apt 软件，包括 gcc、 make、cmake 等；

4）返回到普通用户目录，输入 cd Python-3.7.5 命令安装 python 环境，并继续安装相应的 pip 依赖，包括 numpy、scipy、sympy 等；

5）执行./Ascend-Toolkit-20.0.RC1-arm64-linux_gcc7.3.0.run –install 命令安装 toolkit 开发工具包，包括 arm 版本和 x86_64 版本的 Ascend-cann-toolkit 包，如图 7 所示；

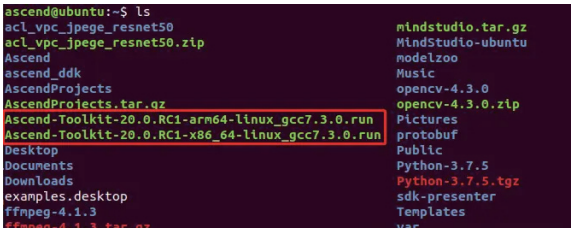

图 7　获取 toolkit 包结果图

6）安装 MindStudio，MindStudio 是针对算子开发的可视化软件，配置并搭载 Ascend-cann-toolkit 包，如图 8 所示。

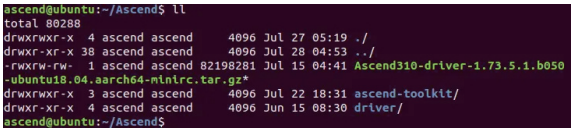

图 8　获取 MindStudio 安装包结果图

然后需要安装运行环境，包含以下 6 个步骤：

1）下载制卡软件包，制卡脚本和 Ubuntu 系统镜像，采用 git clone https://gitee.com/ascend/tools.git 完成，并给脚本和系统镜像增加权；

2）安装制卡需要的 python 依赖，输入 pip3 install pyyaml 命令完成下载；

3）安装制卡需要的 apt 依赖，包括 qemu-user-static、binfmt-support、python3-yaml 等，输入 sudo apt-get install qemu-user-static binfmt-suppor…命令完成下载；

4）利用读卡器和 PC 机制卡，输入 python3 make_sd_card.py local /dev/sdb 命令执行脚本制卡；

5）连接开发板，配置 USB 虚拟网卡 IP；

6）登录到开发板，将动态链接库路径 添加到 ldconfig 文件中，完成运行环境安装，具体结果如图 9 所示。

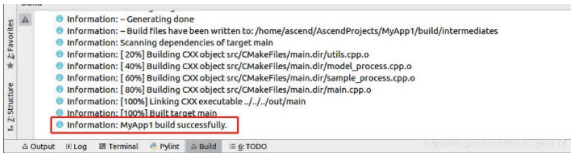

图 9　完成运行环境安装图

在完成环境安装后，可以利用 MindStudio 运行自带的 Classification(resnet50)样例进行验证。选择模型后进行模型转换、编译、添加图片文件后运行。获得“Running workspace_mind_studio_MyApp1 on the remove host finished”的结果，说明环境配置成功。

5.2　库函数安装

在完成环境部署后，还需要安装第三方库才能使用，完成最后的环境准备，包含以下 7 个步骤：

1）设置开发板连接网络，输入 ssh HwHiAiUser@192.168.1.2 命令连接开发板，切换 root 用户，更改 netplan 配置文件，填写配置后将网线连接开发板，开发板可以正常上网；

2）对开发板进行换源，在 root 用户模式下，输入 vi /etc/apt/sources.list 命令打开源，更换后执行 apt-get update 完成更新；

3）运行环境安装 apt 软件包，输入 apt-get install build-essential libgtk2.0-dev…命令安装 python 依赖，执行 pip3 install --upgrade pip 更新 pip 工具包；

4）切换到普通用户模式，安装第三方库，包括 ffmpeg 和 openCV，采用 wget 命令下载到开发板，解压后安装；

5）配置 ffmpeg，输入 vim /etc/ld.so.conf.d/ ffmpeg.conf 命令打开配置文件，增加绝对路径，然后执行 source /etc/profile 使配置文件生效；

6）激活 openCV，切换为 root 用户，使用 cp 命令激活 openCV 库，然后修改环境变量；

7）完成以上步骤，在开发环境的普通用户下将运行环境上导入安装的第三方库开发环境中，以提供编译使用，并安装交叉编译器。

5.3 编写推理应用

在完成环境配置和模型准备后，就可以进行模块构建了，采用模块化设计，具体流程如图 10 所示，构建了各个模块相互协调的推理模型。

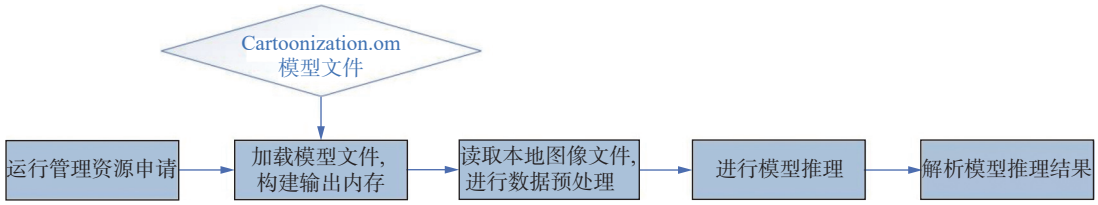

图 10 推理应用架构图

1）运行资源申请。主要用于初始化系统内部资源，固定的调用流程。

2）加载模型文件并构建输出内存。从文件加载离线模型数据，需要由用户自行管理模型运行的内存，根据内存中加载的模型获取模型的基本信息，包含模型输入、输出数据的数据 buffer 大小；由模型的基本信息构建模型输出内存，为接下来的模型推理作好准备。再进行预处理数据。

3）数据预处理。使用 OpenCV 从本地存放有图像数据的目录中循环读取图像数据，对读入的图像数据进行预处理，然后构建模型的输入数据。此过程中，先使用 caffeio.load_image 读取图片，使用该接口读取图像同时实现归一化；然后对读取出的图片进行色域转换成 Lab 格式，分离出 L 通道，将原始图片进行 resize 至模型所需要的宽高；对 resize 之后的图片进行色域转化成 Lab 格式并且分离 L 通道，用 L 分量减均值。在获取到输出结果后，还需要反向处理推理结果。

4）模型推理。根据加载的模型、构建好的模型输入数据和申请好的模型输出内存进行模型推理。

5）解析推理结果并将其应用于原始图像以生成彩色图像。这一步首先对模型的推理输出进行分析，这些输出通常是与图像像素相关的特征值。然后，通过像素级操作，将这些推理结果整合到原始图像中，生成新的彩色图像。

6 实验结果

通过完成环境搭建，根据实验方案，完成实验，获得结果。使用通过整合公开数据集获得的黑白图像数据集测试实验结果，将数据放在 data 文件夹下，然后进行编译运行，得到的结果存储在 out 文件夹下。

选取一些结果图进行展示，并对结果进行分析，展示了一些结果，黑白图像案例如图 11 所示。该图像的主体是一只蝴蝶，根据蝴蝶的纹理，可以对黑白图像进行上色。

图 11 黑白蝴蝶图像

通过对图像的处理，得到黑白图像上色的推理结果，如图 12 所示。通过这个图像可以看到，黑白图像已经转换为彩色图像，整体的背景是绿色的，并且有一定区别。而蝴蝶的颜色是渐变的，由头到翅膀逐渐变浅，图片整体看起来比较自然。与真实的彩色图片很贴近。

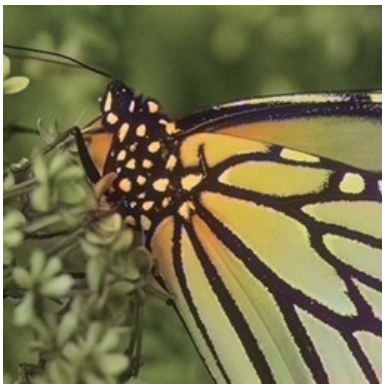

图 12 上色后的彩色蝴蝶图像

为了进一步展现上色效果，选取了景色图进行对比。黑白图像如图 13 所示，这幅图主体是马，背景是草原。

图 13　黑白景色图像

通过对图像的处理，得到黑白图像上色的推理结果，具体的结果如图 14 所示。可见，这幅图的上色效果非常好，对于马的上色，整体上很真实，而且考虑到了光线的影响。而对于整体的背景，以绿色为主，并且增加了渐变的枯黄的感觉，整体的效果更像是彩色图片。

图 14　上色后的彩色景色图像

依托 Atlas 200DK 推理环境的性能，最终可以完成对 Colorization 网络模型的快速部署，快速实现对黑白图像的上色处理。

更多的结果如图 15 所示，可见本实验可以很好地实现对黑白图像的上色处理。对于建筑、事物这种色彩明显的简单图片，可以获得很好的上色效果；对于人像、动物这种有背景的图片，对图像主体也可以很好地实现上色，但是背景色会有些模糊；而对于一些色彩比较复杂的图像，如最后的魔方图，因为色彩复杂，根据黑白图像很难确定是哪些颜色，虽然也将每一个颜色分割开，但是获得的上色结果就可以明显分辨出区别。所以，对于大多数图片，本实验可以较好恢复黑白图像，而对于一些色彩复杂的图片，本实验也可以完成颜色的恢复，但与原彩色图片的差别略大。从观赏角度来说，实现上色使得黑白图片看起来焕然一新，看起来也更真实；从信息角度，上色后的图片也可以很好地展现出图像中的复杂信息。

图 15　图像上色结果对比图

在实验结果的最后，对比 Atlas 200DK 和一台配置为 Intel Core i5 处理器、16 GB 内存、NVIDIA GeForce GTX 1080 显卡的计算机在推理速度上的表现。实验结果表明，通过 Atlas 200DK 实现黑白图像上色可以获得很好的效果，并且推理速度优于计算机。具体来说，通过开发板实现图片推理，每秒可以处理约 25 张图片；通过计算机完成图片推理，每秒可以处理约 16 张图片。相较于采用计算机来实现图片推理，Atlas 200DK 的推理速度提升了 1/3。在资源消耗方面，Atlas 200DK 的资源消耗也远小于计算机，相较于计算机，Atlas 200DK 减少了很多不必要的开销，专注于实现图片推理。通过分析可以看出，在模型推理方面，Atlas 200DK 开发板的优越性。

7　结束语

实验通过基于 Atlas 200DK 的环境搭建完成了黑白图片上色实验，通过理解黑白图像上色的原理和网络结构，并学习开发环境和推理环境的搭建，加深了学生对 Linux 系统的理解，使他们学会了使用开发板进行深度学习推理的过程。

实验方式和结果表明，基于 Atlas 200DK 开发板，能够提升深度学习的推理效率，同时由于开

发板的芯片占用的资源更少，可以满足实时性和可靠性。该实验的灵活与便利为其未来在深度学习推理方面的应用提供良好的支撑，并证明了其在实际应用中的可能性。

参 考 文 献

[1] WANG S, WANG F, ZHU Z, et al. Artificial intelligence in education: A systematic literature review[J]. Expert Systems with Applications, 2024, 252: 124167.

[2] CHEN L, CHEN P, LIN Z. Artificial intelligence in education: A review[J]. IEEE Access, 2020, 8: 75264−75278.

[3] 李德毅, 马楠. 人工智能看教育[J]. 高等工程教育研究, 2023, 3(5): 1−7.

[4] ŽEGER I, GRGIC S, VUKOVIĆ J, et al. Grayscale image colorization methods: Overview and evaluation[J]. IEEE Access, 2021, 9: 113326−113346.

[5] WU Y Z, WANG X T, LI Y, et al. Towards vivid and diverse image colorization with generative color prior[C]//2021 IEEE/CVF International Conference on Computer Vision (ICCV). New York: IEEE, 2021: 14357−14366.

[6] HUANG S S, JIN X, JIANG Q, et al. Deep learning for image colorization: Current and future prospects[J]. Engineering Applications of Artificial Intelligence, 2022, 114: 105006.

[7] DESHPANDE A, LU J J, YEH M C, et al. Learning diverse image colorization[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). New York: IEEE, 2017: 2877−2885.

[8] 华为企业业务. Atlas 200 DK | 开发者套件 | 高性能 AI 应用开发板[EB/OL]. [2023-08-17], http://e.huawei.com/.

[9] Huawei Global. Atlas 200 DK AI Developer Kit [EB/OL]. [2023-08-17], https://consumer.huawei.com/.

[10] 华为. Ascend 开发者社区—华为[EB/OL]. [2023-08-17], https://www.hiascend.com/.

[11] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84−90.

[12] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[Preprint]. arXiv preprint arXiv: 1409.1556, 2014.

[13] ZHANG R, ISOLA P, EFROS A A. Colorful image colorization[M]//LEIBE B, MATAS J, SEBE N, et al, eds. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016: 649−666.

[14] 昇腾社区. 卡通图像生成[EB/OL]. [2023-08-17], https://www.hiascend.com/marketplace/mindsdk/case-studies/93d4d572-cb2a-4cbd-9030-dc475fa567b9.

编辑 葛晋