

程卉

✉ chengh76@mail2.sysu.edu.cn

☎ (+86) 139-2216-3662

🌐 github.com/chenghui-666

教育背景

中山大学，广州

2020.9 至今

在读本科生，计算机学院，信息与计算科学专业，预计 2023.7 毕业

中山大学，深圳

2019.9-2020.7

在读本科生，电子与通信工程学院，电子信息类



相关课程：

- 【数学】数学分析(98)、高等代数(98)、概率论与数理统计(100)、运筹学与最优化(98)、离散数学、复变函数、解析几何 (97)、常微分方程 (98)
- 【计算机体系结构】计算机组成原理、计算机网络、操作系统原理、数据结构与算法、高性能计算 (97)
- 【人工智能】模式识别、机器学习与数据挖掘、神经网络原理、人工智能、强化学习
- 【其它】信息论与编码 (95)、科技论文写作、数值计算方法、数字电路

GPA: 92.2/100(4.22/5.00) Rank: 1/35 CET-4: 561 CET-6: 516

获奖情况

MICCAI 2022 旅行奖(获奖率约3.7%)、校级二等奖学金、校级三等奖学金等

科研经历

基于条件生成对抗网络的头发图像生成

2022.1 至今

导师：李冠彬副教授，中山大学人机物智能融合实验室

- 针对每新增一种发色需要大量数据标注的问题，提出了图像生成的两阶段学习策略，通过“归一化-特征化”的思路，使用 50 张标注数据的生成效果能达到标注量为 500 张的水平。

缺失源数据的无监督域迁移 (source-free unsupervised domain adaptation)

2022.3-2022.5

导师：李冠彬副教授，中山大学人机物智能融合实验室

- 基于源模型将目标域数据分为 source like 和 target specific 两类，从而将 source-free unsupervised domain adaptation(SFUDA) 的问题转化为 unsupervised domain adaptation(UDA) 的问题来求解。
- 提出并证明了 target domain error 的一个上界，把减小 error 的问题转换为减小上界的问题，并以此指导我们的方法。
- 提出了基于上述问题转化设定的对比学习方法，同时利用了 source model 带来的全局和局部特征。
- 提出了一种新的 domain discrepancy 的度量方法，解决传统 Maximum Mean Discrepancy(MMD) 在采样下带来的距离负值情况以及减小了 batch bias。
- 以学生第二作者的身份将该工作投稿至NeurIPS 2022，目前在审。

基于课程学习的甲状腺超声图像良恶性二分类

2022.2-2022.3

导师：李冠彬副教授，中山大学人机物智能融合实验室

- 针对医生对于甲状腺超声图像良恶性的判断常与活检标签不一致的困难，提出了一种自适应的课程学习方法 ACL，通过网络学习自适应的阈值来划分奇异样本（即困难样本）与简单样本，使网络能逐步学习图片特征。
- 利用了先进先出队列平衡自适应阈值的准确性和时效性。
- 使用了自适应系数评估网络判断能力的稳定性，用于校准阈值。
- 实验结果显示 ACL 在甲状腺超声图像的分类任务上达到 SOTA，并能成功通过阈值划分简单样本和奇异值样本。
- 以共同第一作者身份将该工作投稿至 MICCAI 2022，论文已经被接收为oral，项目和数据集将开源至 Github。

基于深度学习的 T2DM 湿热证组面象特征二分类

2021.7-2021.8

导师：陆遥教授，中山大学计算医学成像实验室

- 针对训练数据量少、类别不平衡的问题，建立了基于 ResNet50 的面象图像分类模型，通过迁移学习微调模型。
- 使用了 focal loss 加强对难分样本的学习，改善面象图像数据不平衡问题（正负样本比例约为 1:20）。
- 使用五折交叉验证评估模型准确度，最终结果在测试集上平均 auc 为 0.8。
- 该项目属于**科技部重点研发课题**：中医证候辨识共性技术方法学研究（2018YFC1704206）。

课程项目

高性能程序设计

2021 年秋季

- 通过 CUDA 实现 CNN 的 2 维卷积操作：包括滑窗法和 im2col，并与 cuDNN 库的性能进行对比。
- 分别基于算法分析和软件优化的优化方法，对矩阵乘法进行优化。
- 运用 MPI、Pthread 和 OpenMP 等工具优化矩阵乘法，并改造成矩阵乘法库函数。
- 实现基于 Pthread 的 parallel_for 函数，并将通用矩阵乘法的 for 循环改造成基于 parallel_for 的并行化矩阵乘法。
- 使用 MPI 和 Pthread，完成傅里叶变换 (fft) 的并行化实现。
- CUDA 编程：通过 CUDA 实现通用矩阵乘法的并行版本，与 CUBLAS 库的性能进行对比。

凸优化

2021 年秋季

- 设计分布式算法，仿真在 10 节点的分布式系统下，使用邻近点梯度法、交替方向乘法、次梯度法得到的实验结果与真实值和最优值的距离。
- 使用梯度下降法和随机梯度法求解 MNIST 数据集上的分类问题。
- 课程期末成绩99分，单项成绩专业排名 1/35。

2021 年春季

基于 Linux 内核的操作系统设计

- 编写了一个二级文件系统，可以实现文件/目录的创建和删除、用户登录和权限设置等功能。
- 编写了字符设备驱动程序，可用多个 terminal 对设备进行读/写操作。
- 实现进程的管道通信和软中断通信，使用互斥锁来实现进程的同步与互斥，并解决了死锁问题。
- 实现了 ls 和 cp 指令，并能直接从用户程序中调用。
- 在操作系统实验课中获得**满分 (100/100)**。

2020 年秋季

MIPS 架构下的多周期 CPU 设计

- 使用 Verilog 语言设计了一个基于 MIPS 指令集的多周期 CPU，并在 FPGA 上实际运行。
- 用 Python 实现了一个汇编器，支持 MIPS 指令自动转化为二进制指令。

2020-2021 学年

其它

- 在国产计算平台和 Matlab 上求解矩阵的 LU 分解和用列主元消去法求 $PA=LU$ ，对比分析在不同计算平台上运行相同算法的耗时和精度差别。
- 用 C++ 实现：校园电子卡管理系统，具有存钱、转账、查询、增删卡的功能；多项式计算器，能完成多项式加减乘、求导、求值；基于邻接多重表的图的存储，并将遍历获得的生成树在 Matlab 中可视化。
- 数据结构与算法课程期末成绩接近满分，单项成绩专业排名 1/39。

基本技能

- 编程语言：C、C++、Python、M 语言、Verilog、Octave
- 工具：Pytorch、Matlab、Vivado HLS、MPI、OpenMP、cuDNN、AutoCAD、LaTeX、Markdown

Hui Cheng



✉ chengh76@mail2.sysu.edu.cn

☎ (+86) 139-2216-3662

🌐 github.com/chenghui-666

Information

Sun Yat-sen University, Guangzhou

2020.9-now

Undergraduate student, School of Computer Science and Engineering, Information and Computer Science

GPA: 92.2/100(4.22/5.00) Rank: 1/35 CET-4: 561 CET-6: 516

Honors

MICCAI 2022 student travel reward, The Second Prize Scholarship, The Third Prize Scholarship, etc

Research

Virtual Hair Dying With cGAN

2022.1-now

Advisor: Prof. Guanbin Li, Human Cyber Physical Intelligence Integration Lab, Sun Yat-sen University

- It requires a large amount of labeled data to train a virtual hair dying model with cGAN. To address this challenge, I propose a two-stage training method, utilizing the normalization-featurization idea.
- Experiment shows that the an-notation amount can be reduced **from 500 paired images to 50 paired images**.

Source-free Unsupervised Domain Adaptation

2022.3-2022.5

Advisor: Prof. Guanbin Li, Human Cyber Physical Intelligence Integration Lab, Sun Yat-sen University

- We propose to separate target domain data into two categories, source-like data, and target-specific data. By doing so, we convert the Source-free unsupervised domain adaptation(SFUDA) problem into an unsupervised domain adaptation (UDA) problem.
- We propose an upper bound of the target domain error with proof. Hence our method can be guided by minimizing the upper bound.
- We propose a contrastive learning method to utilize both the global and local features of the source model.
- We propose a novel domain discrepancy metric to reduce the batch bias of Maximum Mean Discrepancy (MMD) and align the source-like and target-specific domain.
- **This work is submitted to NeurIPS 2022. I am the second student author in this work.**

Adaptive Curriculum Learning for Thyroid Nodule Diagnosis

2022.2-2022.3

Advisor: Prof. Guanbin Li, Human Cyber Physical Intelligence Integration Lab, Sun Yat-sen University

- Clinical diagnosis of thyroid nodules is not always consistent with biopsy results. To address this challenge, we propose an adaptive curriculum learning method ACL to adaptively learn samples from easy to hard during the training process.
- We use an FIFO queue to make our threshold accurate and time-efficient.
- We adopted the adaptive coefficients to evaluate the robustness of the model to calibrate the threshold.
- The experiment shows that ACL achieves SOTA performance on the classification task of thyroid ultrasound images, and successfully selects and discards the inconsistent samples.
- **This work is accepted by MICCAI 2022(oral). I am the co-first author of this work.**

T2DM Damp-heat Syndrome Classification

2021.7-2021.8

Advisor: Prof. Yao Lu, Computational Medical Imaging Lab, Sun Yat-sen University

- I propose a classification model based on transfer learning to address the challenge of lack of training data in the T2DM damp-heat syndrome classification problem.
- I use FocalLoss to address the data imbalance challenge. The ratio of positive to negative samples is 1:20.
- The experiment result achieves an average AUC of 0.8 in the five-fold cross-validation.
- This project belongs to the key research and development project of the Ministry of Science and Technology: Research on the Common Technology and Methodology of TCM Syndrome Identification (2018YFC1704206).

Projects

High-Performance Computing Programming

Fall 2021

- I implemented the 2D convolution operations of CNN with CUDA programming, including the sliding window and im2col method. I also compare the performance with the cuDNN library and analyze it.
- I implemented matrix multiplication based on algorithm analysis and software optimization.
- I implemented the matrix multiplication library functions with MPI, Pthread, and OpenMP.
- I implemented the parallel_for function with Pthread, and use the parallel function to implement the parallelized matrix multiplication with a general for loop.
- I implemented the parallelized Fast Fourier Transform (FFT) with MPI and Pthread.
- CUDA Programming: I implemented a parallel version of generic matrix multiplication via CUDA, and compared its performance with that of the CUBLAS library.

Convex Optimization Programming

Fall 2021

- I implemented distributed PGD, ADMM, and subGD algorithms to fit the ground truth under a 10-node distributed system.
- I complete a classification task on the MNIST dataset using gradient descent and stochastic gradient descent.

Operating System Based on Linux Kernel

Spring 2021

- I implemented a secondary file system, which has the following functions: file/directory creation and deletion, user-login and permission setting.
- I implemented a character device driver so that the device can be read/written by multiple terminals.
- I implemented the pipe communication and soft interrupt communication of the process and solved the deadlock problem. A mutex is used for synchronization and mutual exclusion.
- I implemented ls and cp instructions which can be called directly from the user program.
- I got **full marks (100/100)** in the operating system laboratory class.

Multicycle CPU Design under MIPS Architecture

Fall 2020

- I implemented a multi-cycle CPU based on the MIPS instruction set using Verilog language. It can run on FPGA.
- I implemented an assembler in Python, which converses MIPS instructions into binary instructions automatically.

Skills

- Programming Language: C, C++, Python, M Language, Verilog, Octave
- Tools: Pytorch, Matlab, Vivado HLS, MPI, OpenMP, cuDNN, AutoCAD, LaTeX, Markdown