**中英文对照题目**

**基于PYNQ-Z2的图像识别算法集成模型设计**

**Design of image recognition algorithm ensemble model based on PYNQ-Z2**

摘 要

“基于PYNQ-Z2的图像识别算法集成模型设计”旨在探索FPGA技术在提高图像处理效率和实现实时应用方面的潜力。我们的研究重点在于板卡中集成不同的算法模型，一种是通过对已有的算法（CNN、VGA等）进行优化训练，将训练好的模型封装成pytorch并形成权重文件，供后期统一的调用，实现了分类模型以及口罩检测模型。另一种算法模型采用caffe框架，以便可以直观的获得网络结构的图形化表示，实现了快速特征嵌入的卷积结构，在模型中实现年龄和性别的检测。多算法集成提供了一个综合的解决方案，这种多模型协同的策略增强了系统的综合性能和应用范围。

**关键词：FPGA技术pynq-z2，图像处理算法改进（cnn，vga等），pytorch，caffe，算法集成**

**Abstract**

" Design of image recognition algorithm ensemble model based on PYNQ-Z2 " aims to explore the potential of FPGA technology in improving image processing efficiency and realizing real-time applications. Our research focuses on integrating different algorithm models into the board. One is to optimize and train existing algorithms (CNN, VGA, etc.), encapsulate the trained model into pytorch and form a weight file for later unified Called to implement the classification model and mask detection model. Another algorithm model uses the caffe framework to intuitively obtain a graphical representation of the network structure, implement a convolutional structure that implements fast feature embedding, and implement age and gender detection in the model. Multi-algorithm integration provides a comprehensive solution. This multi-model collaboration strategy enhances the comprehensive performance and application scope of the system.

**Keywords: FPGA technology pynq-z2, image processing algorithm improvement (cnn, vga, etc.), pytorch, caffe, algorithm integration**

**目 录**

目录

[项目概述 5](#_Toc164067097)

[项目成员基本情况、指导教师基本情况 5](#_Toc164067098)

[项目的选题背景与意义 5](#_Toc164067099)

[（1）选题背景： 5](#_Toc164067100)

[（2）研究意义： 6](#_Toc164067101)

[项目实施过程的人员工作分配和完成情况 6](#_Toc164067102)

[（1）人员工作分配： 6](#_Toc164067103)

[项目实施过程收获和体会 7](#_Toc164067104)

[项目预期成果完成情况和创新点 8](#_Toc164067105)

[项目成果完成情况： 8](#_Toc164067106)

[创新点： 8](#_Toc164067107)

[项目说明 10](#_Toc164067108)

[实验方法设计及方案 10](#_Toc164067109)

[实施过程： 10](#_Toc164067110)

[项目特色 11](#_Toc164067111)

[最终实现功能展示： 11](#_Toc164067112)

[项目总结 14](#_Toc164067113)

[附录 14](#_Toc164067114)

# 项目概述

## 项目成员基本情况、指导教师基本情况

项目人数：2

项目成员：

刘英琦：大连理工大学软件学院2021级软件工程专业

张希哲：大连理工大学软件学院2021级软件工程专业

指导教师：王洁

职称：副教授

专业领域：可信软件与容错计算，并行计算，信息内容安全

## 项目的选题背景与意义

### **（1）选题背景**：

随着数字多媒体的快速发展，数字图像的处理技术被广泛应用于航空航天，通信，医学及工业生产等领域中，新研发的产品在图像的存储容量，图像清晰程度，图像处理速度等方面有了新的需求。在图像处理过程中，虽然处理算法简单，但参与运算的数据量大，数据需多次重复使用，因此，图像处理往往是图像处理系统中最为耗费时间的一个环节，对整个系统速度影响较大。在当前图像识别算法研究已经很成熟的背景下，提高图像处理的时效性有很大的应用前景。FPGA技术是在PAL、GAL等可编程器上发展出的新型产物，在专用集成电路中有着广泛的运用。FPGA作为一种高密度的集成电路，改善了传统可编程器门电路有限的问题，增强了定制电路的逻辑功能，可以重复进行编程，重构系统。这为配置数据和系统功能的灵活性运用提供了基础。FPGA由嵌入式SRAM、数字时钟、逻辑模块组成，其丰富的避险资源，为可编程逻辑的控制提供了多种算法设计，可用于SRAM的更新升级，而且FPGA所具有的大量门电路能够有效实现加减法等逻辑运算，复杂算法也可以通过编程来实现，这与传统数字信息处理器相比，FPGA的计算能力更强。FPGA在使用时应当采用同步设计，可以防止出现不稳定状态，避免对系统稳定性产生影响，FPGA技术适合实时性要求高、频率快的功能模块，在使用时需要开了存储器和芯片资源占用，合理配置系统结构。在此基础上，就为图像识别算法设计在算法，系统结构上带来了新的思路与方法。由于图像中的所有元素均可实施以同样的操作，存在固有的并行性，非常适合映射到FPGA架构中由硬件算法实现，使得图像处理的速度大大加快。对于图像识别处理，底层的图像处理的数据量很大，要求处理速度快，但运算结果相对比较简单，以FPGA作为主要处理芯片的图像识别系统非常适合对于对图像进行快速识别。

### （2）研究意义：

在如今的图像识别系统中，方案大致有以下几种:1)软件实现，此方案大多用于算法的仿真与验证，较差的灵活性与实时性是其缺点;2)通用单片机实现，此方案大都用于简单的数字处理，缺点就是MCU的运算速度相对较慢;3)通用数字信号处理器实现，此方案使用广泛，为大多数开发人员所利用，其缺点就是外围电路设计相对复杂，开发及调试的难度系数大，开发周期较长，系统的扩展性和升级性较差;4)专用数字信号处理器实现，此方案一般应用于对信号处理速度要求极高的场合，缺点就是灵活性差且成本高。本项目研究目标是实现一种基于 FPGA 的图像识别算法。通过 FPGA 完成对图像的采集和处理，图像中元素的识别归类，识别结果的显示等功能，充分利用FPGA并行性和模块化的设计理念，提升效率，缩短开发周期。利用FPGA技术来实现对各种图像分类识别的研究来检验和验证一些新的算法和理论有着非常重大的意义，同时也为光学字符识别(OCR)、人脸识别、军事卫星图像、医学图像和地貌图像的判别、理解和分析等应用项目开辟出了另一条可行的道路。

## 项目实施过程的人员工作分配和完成情况

### （1）人员工作分配：

刘英琦（项目负责人）：负责统筹规划整个项目组的任务分配、资源调配等工作以及把控项目研究主要方向，跟踪项目进度；对已有的算法（如经典CNN算法）进行优化训练，将训练好的模型封装成pytorch并形成权重文件，供后期统一的调用，实现了分类模型以及口罩检测模型；负责API的部署，将训练好的模型统一集成到板卡之中，同时做好调用硬件资源的分配工作

张希哲：负责采用Caffe框架，实现快速特征嵌入的卷积结构，通过DNN残差网络算法处理摄像头捕捉到的人脸，然后传入模型中实现年龄和性别的检测；负责收集训练数据和测试数据，并对数据进行预处理，对项目进行运行检测和维护

（2）完成情况：

小组成员均按照预期进度进行项目，虽然过程中遇到了很多技术难题，比如在算法的代码调试等等各方面，但我们积极寻求了指导老师的帮助，小组成员任务基本顺利完成预期目标

## 项目实施过程收获和体会

在我们的项目实施过程中，我们小组取得了以下收获和体会：

我们通过对经典CNN算法进行优化训练，提高了模型的准确性和效率。我们首先将训练好的模型封装成PyTorch并生成权重文件，使得模型的调用更加方便，提高了后期开发的灵活性和可复用性。第二种模型我们选择了Caffe框架，主要是因为其Prototxt结构的清晰性，可以直观地展示网络结构，并且能够对每层的权重和偏置设置不同的学习速率。这种灵活性对于我们实现快速特征嵌入的卷积结构至关重要，使得我们能够有效地捕捉人脸特征，实现年龄和性别的检测。我们的项目实施过程中取得了很多宝贵的经验和收获。我们深刻体会到算法优化和模型封装的重要性，以及选择合适的工具和框架对于项目成功的关键作用。我们也意识到了团队合作的力量，每个成员的努力都对项目的成功起到了不可或缺的作用。我们期待将这些经验和收获应用到未来的项目中，不断提高和完善我们的技术能力。

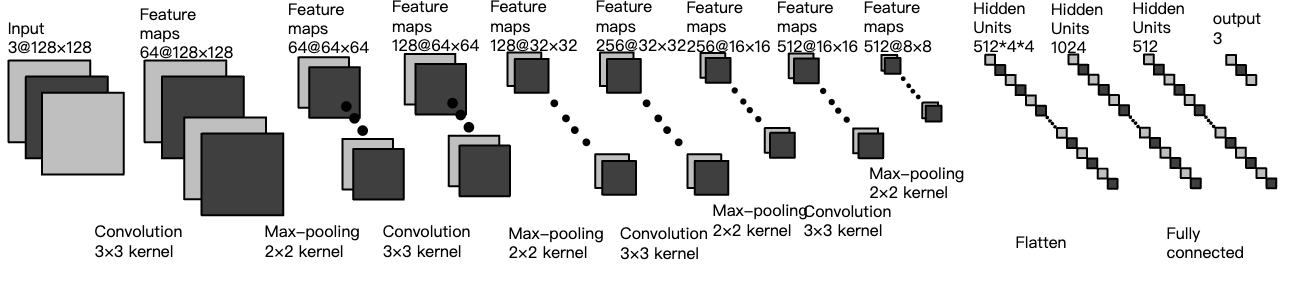
# 项目预期成果完成情况和创新点

## 项目成果完成情况：

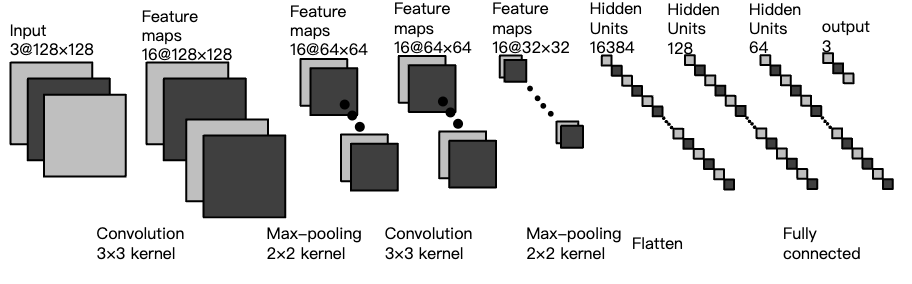
我们成功地对经典算法(CNN,VGA等)进行了优化训练，提升了模型的性能。通过将训练好的模型封装成PyTorch并生成权重文件，我们实现了一个便于调用和复用的模型库，这些模型已经成功应用于分类任务和口罩检测模型，展现出良好的效果和准确性。我们采用了Caffe框架来实现一个快速特征嵌入的卷积结构，这一结构特别适用于捕捉人脸特征。利用Caffe模型的清晰的Prototxt结构和灵活的学习速率设置，我们能够精确地调整网络参数，实现了准确的年龄和性别检测功能

## 创新点：

1. 我们对经典算法（CNN、VGA等）进行了优化训练，优化后的网络结构可以在在不损失准确率的情况，使用更少的资源，这使得模型在处理复杂图像分类任务时更加高效和准确。



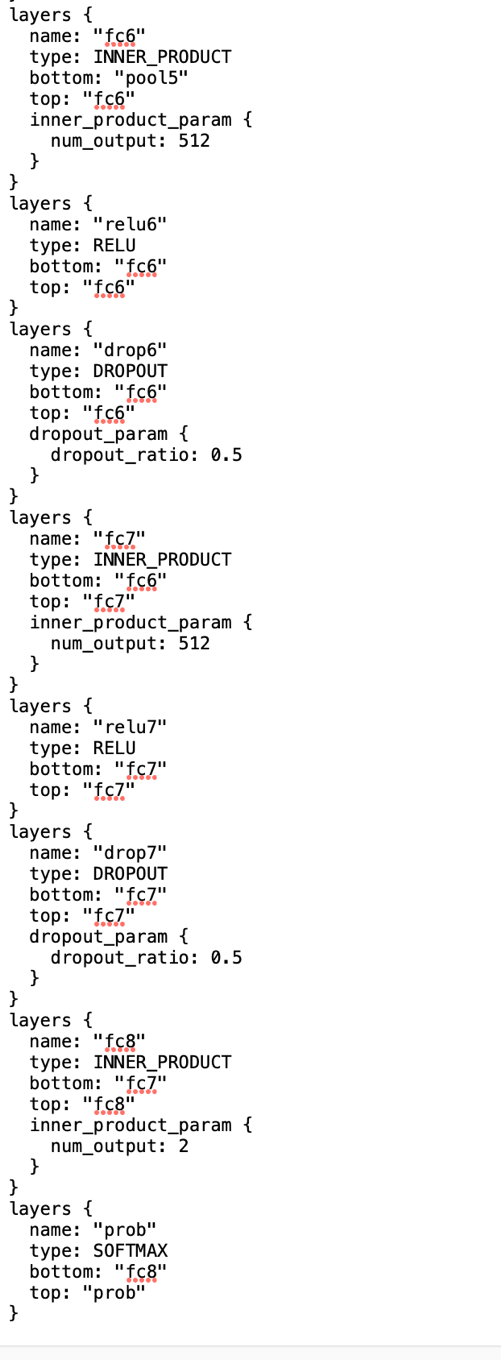
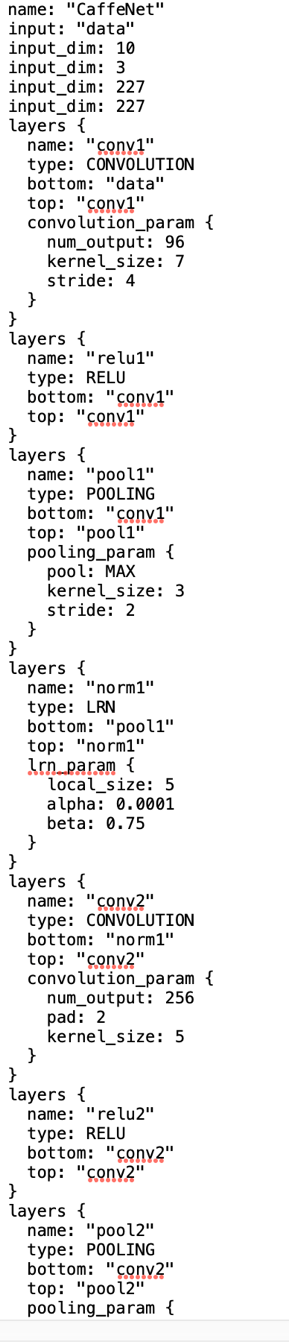
经典CNN网络结构



优化后网络结构

（2）通过将训练好的模型封装成PyTorch并生成权重文件，我们实现了分类任务和口罩检测模型，为后期的开发和应用提供了便利。

（3）我们采用了Caffe框架，并利用其Prototxt结构的清晰性和灵活的学习速率设置，实现了快速特征嵌入的卷积结构，特别适用于捕捉人脸特征，这在年龄和性别检测方面展现出了优异的性能。



Caffe框架展示

（4）项目中不同的模型（如分类模型、口罩检测模型、年龄性别检测模型）协同工作，提供了一个综合的解决方案，这种多模型协同的策略增强了系统的综合性能和应用范围。

# 项目说明

## 实验方法设计及方案

我们的项目旨在探索FPGA技术在提高图像处理效率和实现实时应用方面的潜力。我们的方案是在板卡中集成不同的算法模型，提供一个综合的解决方案。一种是通过对已有的算法（CNN、VGA等）进行优化训练，将训练好的模型封装成pytorch并形成权重文件，供后期统一的调用，实现分类模型以及口罩检测模型。另一种算法模型采用caffe框架，以便可以直观的获得网络结构的图形化表示，实现快速特征嵌入的卷积结构，在模型中实现年龄和性别的检测。

## 实施过程：

准备阶段：

确定实验目标和需求，明确要解决的问题和预期结果。

选择合适的FPGA开发板和相应的硬件资源。

准备所需的软件环境，包括安装PyTorch、Caffe等框架和相关库。

算法优化与训练：

对CNN、VGA等算法进行优化，以提高图像处理的效率和准确性。

使用PyTorch等工具训练模型，并将训练好的模型封装成权重文件。

模型集成与测试：

将优化训练好的模型集成到FPGA开发板中。

开发相应的软件接口，以便在FPGA上调用封装好的模型。

对集成的模型进行测试，验证其在图像分类和口罩检测等方面的能。

实时应用开发：

基于FPGA和集成的模型，开发实时图像处理应用。

调整和优化应用程序，确保其满足实时性能要求。

进行实际场景的测试和验证，评估应用的效果和可用性。

评估与优化：

对实验结果进行分析，评估模型和应用的性能。

根据评估结果进行优化，改进模型和应用的效率和准确性。

进行反复测试和调整，直到达到满意的性能水平。

## 项目特色

FPGA技术应用：项目利用FPGA技术的并行处理能力，提高图像处理的效率，实现实时应用的需求，展示了FPGA在高性能计算领域的应用潜力。

算法优化与集成： 通过对经典的CNN、VGA等算法进行优化和训练，提高了模型的性能。将优化后的模型封装并集成到FPGA板卡中，展示了算法与硬件的紧密结合。

多模型融合： 项目不仅利用pytorch实现了图像分类和口罩检测模型，还集成了基于Caffe框架的年龄和性别检测模型，展示了多模型融合在提升应用功能和性能方面的优势。

实时应用开发： 针对实时图像处理的需求，开发了相应的应用程序，通过优化确保满足实时性能要求，体现了项目在实际应用场景中的可行性和有效性。

## 最终实现功能展示：

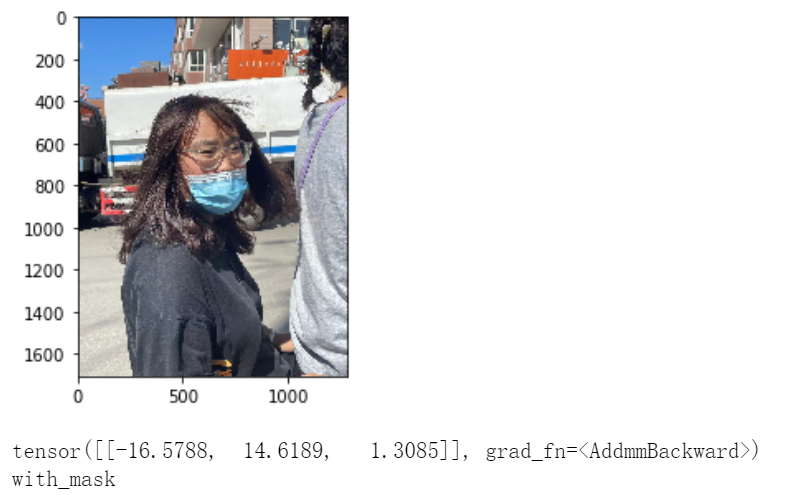


图1 pytorch(口罩检测)

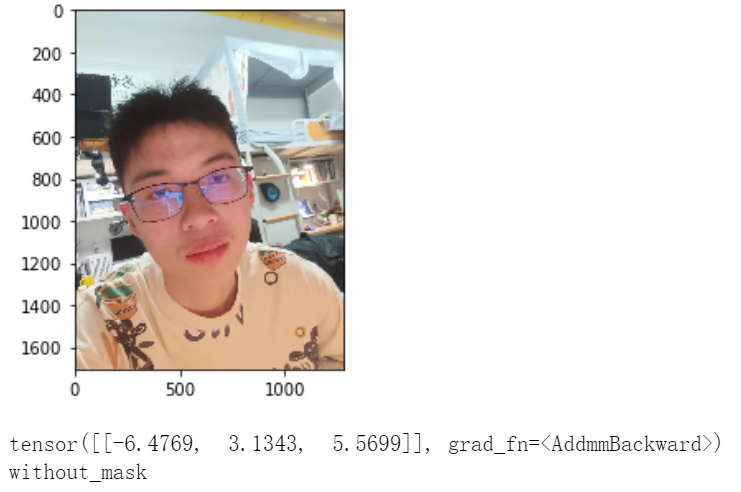


图2 pytorch(口罩检测)

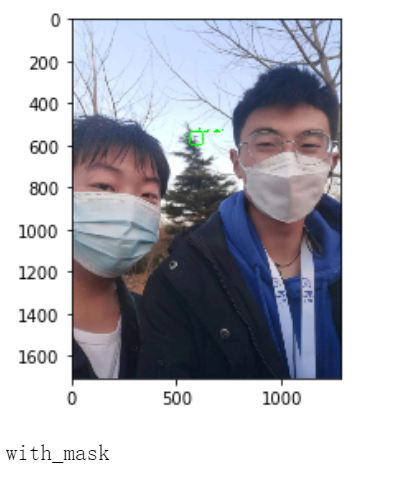


图3 pytorch(口罩检测)

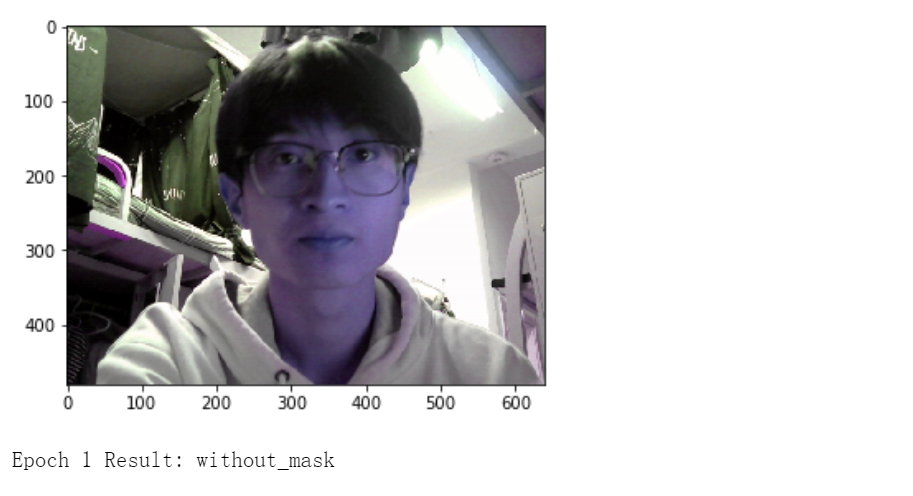


图4 pytorch(口罩检测)

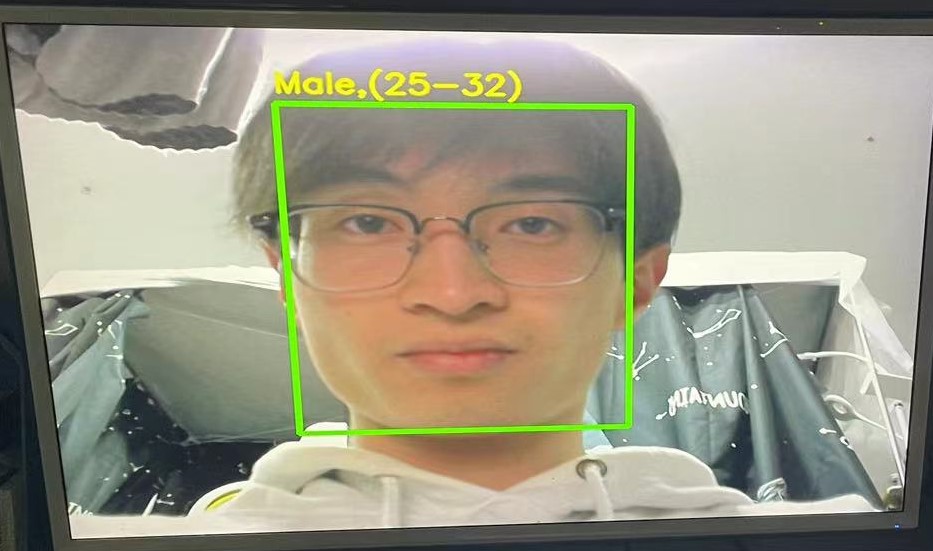


图6 caffe(年龄性别检测)

# 项目总结

本项目致力于探索FPGA技术在图像处理领域的应用潜力，通过对深度学习算法的优化和集成，实现了高效的图像分类、口罩检测以及年龄和性别识别等功能。项目充分利用了FPGA的并行处理能力，加速了图像处理算法的执行，提高了处理效率。通过对经典算法如CNN、VGA等的优化，项目提高了模型的准确性和处理速度。同时，项目将优化后的模型集成到FPGA板卡中，并开发了相应的实时图像处理应用，验证了项目在实际应用场景中的可行性和有效性。此外，项目通过集成不同的模型（如pytorch、caffe），实现了多功能的图像处理应用，展示了多模型融合在提升应用能力方面的潜力。项目还通过持续的性能评估和优化，不断提高了系统的处理效率和准确性，体现了在实际应用中不断迭代和改进的重要性。

# 附录

模型训练代码（以CNN为例）：

import torch  
import torch.nn as nn  
import torch.optim as optim  
import torchvision   
from torchvision import datasets, transforms  
from torch.utils.data import DataLoader  
from torch.optim import lr\_scheduler  
from tqdm import tqdm  
class CNN(nn.Module):  
 def \_\_init\_\_(self):  
 super(CNN, self).\_\_init\_\_()  
 self.Conv1 = nn.Sequential(  
 nn.Conv2d(3, 16, 3, 1, 1),  
 nn.ReLU(),  
 nn.MaxPool2d(2, 2)  
 )  
 self.Conv2 = nn.Sequential(  
 nn.Conv2d(16, 16, 3, 1, 1),  
 nn.ReLU(),  
 nn.MaxPool2d(2, 2)  
 )  
 self.fc = nn.Sequential(  
 nn.Linear(16384, 128),  
 nn.ReLU(),  
 nn.Linear(128, 64),  
 nn.ReLU(),  
 nn.Linear(64, 3)  
 )  
 def forward(self, x):  
 x = self.Conv1(x)  
 x = self.Conv2(x)  
 x = x.view(x.size()[0], -1)  
 x = self.fc(x)  
 return x

def train(model, train\_loader, criterion, optimizer, scheduler, epoch, num\_epochs):  
 model.train()  
 total\_loss = 0  
 total\_correct = 0  
 total\_data = 0  
 train\_bar = tqdm(train\_loader)  
 for data in train\_bar:  
 images, labels = data  
 images, labels = images.to(device), labels.to(device)  
 #梯度清零  
 optimizer.zero\_grad()  
 #正向传播  
 outputs = model(images)  
 \_, predicted = torch.max(outputs.data, dim=1)  
 total\_correct += torch.eq(predicted, labels).sum().item()  
 #计算损失  
 loss = criterion(outputs, labels)  
 total\_loss += loss.item()  
 #反向传播  
 loss.backward()  
 #权重更新  
 optimizer.step()  
  
 total\_data += labels.size(0)  
 train\_bar.desc = "train epoch[{}/{}] loss:{:.3f}".format(epoch+1, num\_epochs, loss)  
   
 scheduler.step()  
 train\_loss = total\_loss / len(train\_loader)  
 train\_acc = total\_correct / total\_data  
 print('Train epoch{}: Loss:{:.4f} Acc:{:.4f}'.format(epoch+1, train\_loss, train\_acc))

Caffe框架部分核心代码展示：

#DNN残差网络

def getFaceBox(net, frame, conf\_threshold=0.7):

frameOpencvDnn = frame.copy()

frameHeight = frameOpencvDnn.shape[0] # 高就是矩阵有多少行

frameWidth = frameOpencvDnn.shape[1] # 宽就是矩阵有多少列

blob = cv2.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123], True, False)

# blobFromImage(image[, scalefactor[, size[, mean[, swapRB[, crop[, ddepth]]]]]]) -> retval 返回值 # swapRB是交换第一个和最后一个通道 返回按NCHW尺寸顺序排列的4 Mat值

net.setInput(blob)

detections = net.forward() # 网络进行前向传播，检测人脸

bboxes = []

for i in range(detections.shape[2]):

confidence = detections[0, 0, i, 2]

if confidence > conf\_threshold:

x1 = int(detections[0, 0, i, 3] \* frameWidth)

y1 = int(detections[0, 0, i, 4] \* frameHeight)

x2 = int(detections[0, 0, i, 5] \* frameWidth)

y2 = int(detections[0, 0, i, 6] \* frameHeight)

bboxes.append([x1, y1, x2, y2]) # bounding box 的坐标

cv2.rectangle(frameOpencvDnn, (x1, y1), (x2, y2), (0, 255, 0), int(round(frameHeight / 150)),

8) # rectangle(img, pt1, pt2, color[, thickness[, lineType[, shift]]]) -> img

return frameOpencvDnn, bboxes

# 网络模型 和 预训练模型  
faceProto = "./AgeGender/opencv\_face\_detector.pbtxt"  
faceModel = "./AgeGender/opencv\_face\_detector\_uint8.pb"  
ageProto = "./AgeGender/age\_deploy.prototxt"  
ageModel = "./AgeGender/age\_net.caffemodel"  
genderProto = "./AgeGender/gender\_deploy.prototxt"  
genderModel = "./AgeGender/gender\_net.caffemodel"

Pytorch部分核心代码展示：

#人脸识别

        #若没有人脸，输出空屏

        if len(faces)==0:

            print("Epoch {} Empty window!".format(i+1))

            videoIn.release()

        #若只有一张人脸，则只输出一次信息

        elif len(faces)==1:

            #图像预处理

            for (x,y,w,h) in faces:

                item = np\_frame[y:y+h, x:x+w]

                cv2.imwrite("temp1.png", item)

                image = Image.open("temp1.png")

                image = image.convert('RGB')

                image\_torch = transform(image)

                image\_torch = image\_torch.view(-1, 3, 128, 128)

            #传入模型进行预测

                output = model(image\_torch)

                \_, predicted = torch.max(output.data, dim=1)

                pred = predicted[0].numpy()

            #输出对应检测结果

                if pred==0:

                    print("Epoch {} Result: mask\_weared\_incorrect".format(i+1))

                    videoIn.release()

                elif pred==1:

                    print("Epoch {} Result: with\_mask".format(i+1))

                    videoIn.release()

                elif pred==2:

                    print("Epoch {} Result: without\_mask".format(i+1))

                    videoIn.release()

        else:

        #若有多张人脸，需在同一行输出多条信息，分别对应画面中从左至右的人脸

            flag = 0

        #设置标志，以控制输出信息

        #此处对检测到的人脸信息进行预处理，由于人脸级联分类器精度较低，存在将面部器官一并识别为人脸的问题，在此删去被重复识别的人脸

其他已优化模型权重文件简略图：

