华南师範大學

《人工智能导论》课程项目

进 展 报 告

项　目　题　目：基于 LRNet 的 AI 换脸识别

所　在　学　院：计算机学院

项　目　组　长：梁玥

小　组　成　员：詹梦纯、何育霏

开　题　时　间：2024 年 9 月 30 日

一、项目任务

本项目的研究目标是通过结合 LRNet 模型和几何特征，(细化模型领域)，**改进人脸预处理模块和校准模块**，从而实现对人脸识别中面部伪造的有效识别。具体方法包括：深入研究 LRNet 模型的预处理模块、结合几何特征和机器学习算法设计新的校准方法、大量实验验证方法可行性和性能表现。突破目前面部预处理模块以及校准模块研究领域上的局限性，实现 AI 伪造人脸的识别方法。

1. **面部预处理模块的优化**

针对人脸识别系统中预处理阶段对关键点提取能力的局限性，深入研究 LRNet 模型**在特征点捕获和几何结构分析**方面的适配性，结合 Dlib 库和改进算法，提升对面部几何特征点的提取精度和数量，为后续检测模块提供更高质量的输入数据。

2. **伪造检测校准模块的优化**

通过引入金字塔-LK 光流算法，提升校准模块对动态特征的捕获能力，特别是在视频序列和跨帧信息对比中的应用，从而增强模型对伪造面部动态纹理和微表情变化的识别能力。该模块将重点解决多尺度信息融合和像素丢失等问题，为 LRNet 模型提供稳定的光流数据支持。

3. **模型鲁棒性与环境适应性**

结合卡尔曼滤波器优化 LRNet 模型的噪声处理能力，减少环境变化和多样化输入（如光线变化、视频质量差异等）对伪造检测精度的影响。该部分研究将聚焦于提高伪造检测模块在实际场景中的鲁棒性和稳定性，并探索其在不同应用场景下的广泛适用性。

二、技术方案

现阶段相关领域的从业者引入了多个用于辨别Deepfake模型制造的伪造人脸的方法（图 1）。

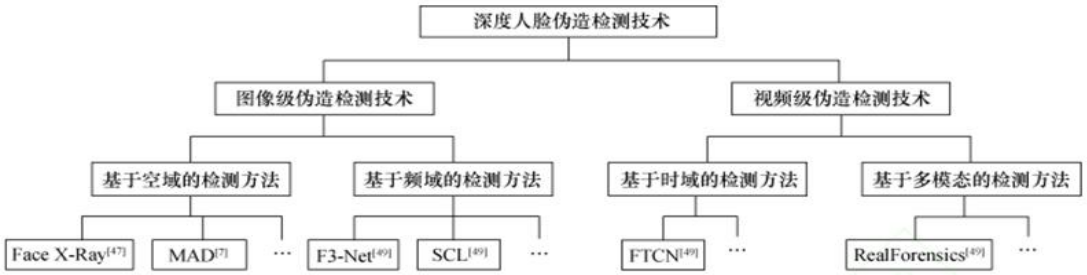


图 1 深度人脸伪造（Deepfake）检测技术总结

目前的Deepfake检测技术融合了图像级和视频级检测方法，前者着眼于**静态特征**，主要包括**基于空间域和频域**的策略。Face X-Ray模型通过检测**人脸融合边界**展现出对未见伪造方法的强泛化能力，而**频域感知分解模块**则在处理**高压缩伪造图像**时表现卓越，二者在检测精度上形成互补。而**视频级检测**则聚焦**动态特征和多模态融合**，以增强对伪造手法复杂性的适应能力。双流分支网络通过**放大伪影细节**并结合时序一致性捕捉伪造痕迹，RealForensics模型则通过**音视频自然对应关系**的学习，实现了对复杂伪造视频的高鲁棒性检测。通过综合利用图像级静态特征与视频级动态信息，不同模型间实现优势互补，为Deepfake检测提供了全面而高效的技术解决方案，展现出对未知伪造方法的强大适应能力和准确性。

在现有的Deepfake检测技术基础上，基于深度学习的LRNet模型能实现对复杂伪造人脸的精确识别和高泛化能力。LRNet模型结合了**空间、频域以及多模态信息**的提取，构建了一种兼具**静态特征分析与动态特征捕捉**的检测框架。

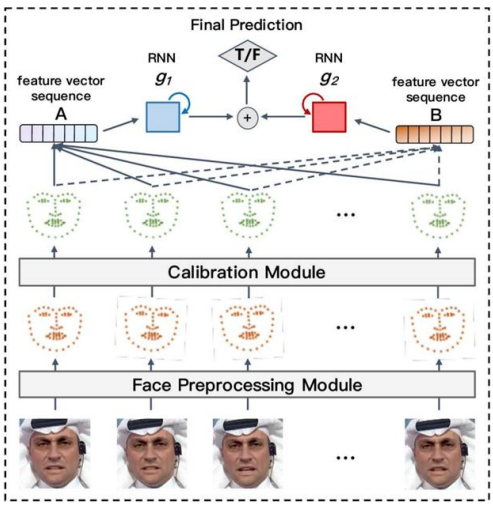


图 2 LRNet模型流程图

在图像级伪造检测中，模型首先通过**多尺度卷积神经网络**提取**伪造图像的局部纹理与频域特征**，以揭示**不同伪造方法中不可避免的微小伪影**。同时，为了增强模型对未知伪造方法的适应性，模型嵌入了**特征自适应模块**，能够动态优化提取的特征向量，提升对跨方法伪造图像的识别能力。

值得注意的是，尽管伪造面部视频在单帧上可能呈现出较高的保真度，但由于伪造技术逐帧生成的固有特性，它们在**细微表情和面部器官运动上**常表现出不自然之处。为捕捉这些“时间伪影”，模型设计了一个**结合几何特征的检测机制**。具体来说，模型通过面部地标的几何形状和位置建模面部动态行为，这些地标是描述面部肌肉运动的重要指标，如动作单元（AU）的强度差异。

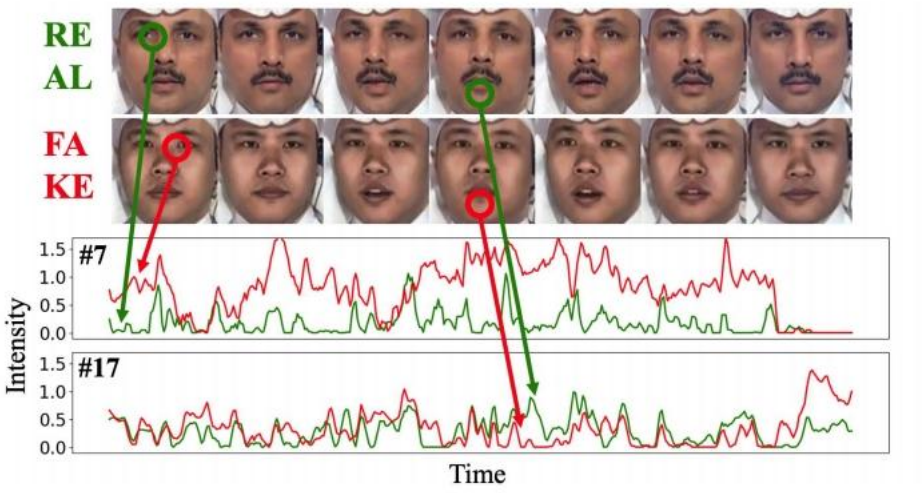


图 3 原始和深度伪造视频序列的动作单元（AU）强度分析

图 3 展示了原始视频序列与深度伪造视频序列的动作单元（AU）强度分析，其中 AU 代表构成面部表情的具体肌肉运动模式。我们特别关注两个最显著的动作单元：7（眶轮匝肌的收缩，用于表现眯眼等动作）和 17（下巴提升器的收缩，用于表现抬下巴等动作）。尽管深度伪造视频在视觉效果上足以迷惑观察者，但通过分析，我们可以发现伪造视频中面部表情的细微差异。例如，即使原始视频和伪造视频中的面部在执行相同的动作，**伪造视频在这些动作单元的表现上仍显得不自然**。这些微小但显著的差异为检测伪造提供了有力的证据。

模型进一步设计了**双流递归神经网络**提取**地标序列的深度时间特征**，并通过地标校准模块减少抖动，从而增强几何特征的鲁棒性和识别能力。利用这种几何特征和深

度时间特征的互补优势，LRNet 能够更有效地区分原始与伪造视频。

在视频级检测方面，LRNet 引入了**时序一致性检测机制**，通过捕捉视频**多帧间的人脸动态特征与伪造细节的不一致性**来区分真实与伪造。模型采用了双分支结构，一方面聚焦伪造过程中常见的几何变形，另一方面通过频域信息分解模块分析高度压缩伪造视频中隐藏的细微差异。此外，为解决伪造视频中音频与图像的不匹配问题，LRNet 融入了**多模态特征对齐**策略，利用残差网络提取音频特征并与视频帧特征进行自监督对齐，进而有效应对伪造视频的复杂性。

通过这些模块的协同作用，LRNet 不仅在静态和动态检测任务中都具备高效性能，还能针对不同伪造方法展现出优异的鲁棒性和泛化能力。实验结果表明，LRNet 在多种公开数据集上的检测精度显著优于现有方法，特别是在应对高压缩伪造视频和跨方法伪造图像时，展现出了领先的表现。

三、实验方案

1. 提出一个高效和鲁棒的框架来分类深度伪造视频，其中我们在精确的几何特征上建模时间特征。

2. 学习一种新的即插即用的地标校准模块 LRNet，以提高几何特征的精度和时间建模的有效性，同时使我们的框架更加灵活和可重复性。

3. 进行广泛的实验来验证方法的有效性和鲁棒性，并探讨影响因素。

四、目前进展

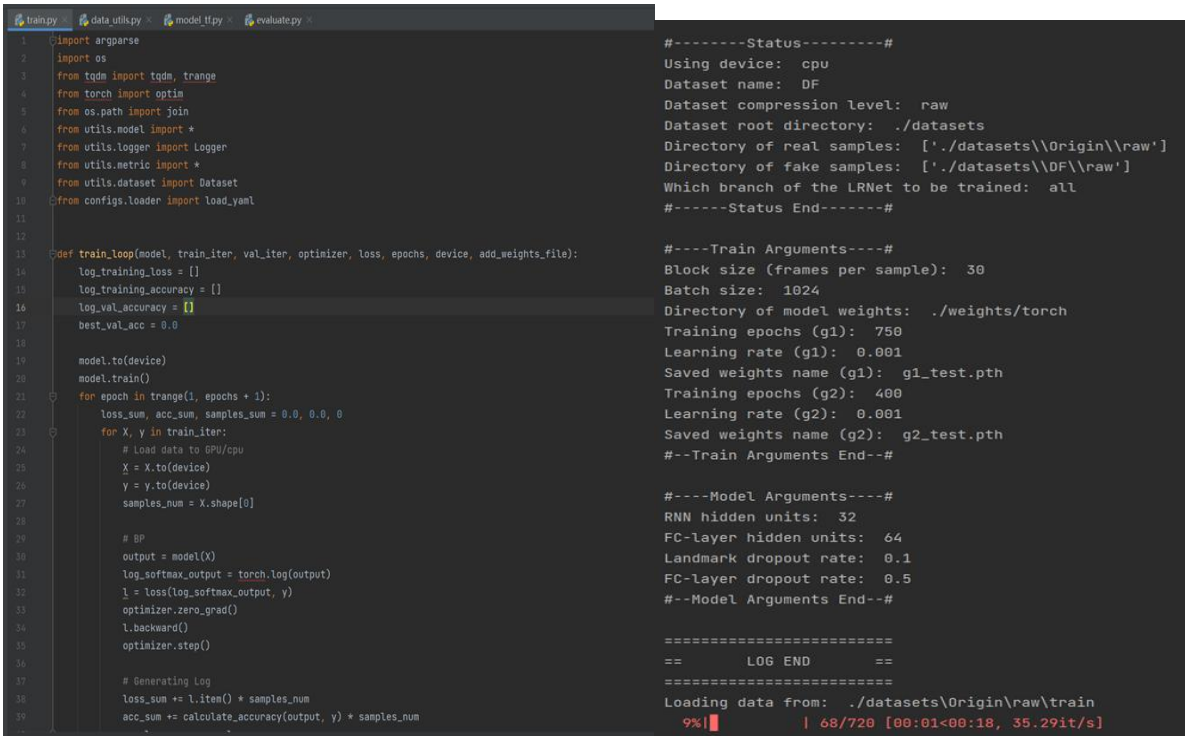
经过自项目立项以来的学习积累，课题组成员已基本掌握深度学习领域的基本概念以及掌握了训练与测试模型的基础算法的代码复现能力，更进一步了解了 DEEPFAKE 人脸识别模型的技术理论。具体完成内容如下：

1. 理解 Python 扩展的 Pytorch 模块运作原理并处理 FaceForensics 数据集。

名称	日期	类型
newtest	2024/10/22 16:22	文件夹
rotated	2024/10/22 16:22	文件夹
test	2024/10/22 16:22	文件夹
test-low	2024/10/22 16:22	文件夹

图 4 经过处理的数据集

2. 基本复现模型代码并按照需求修正特定参数

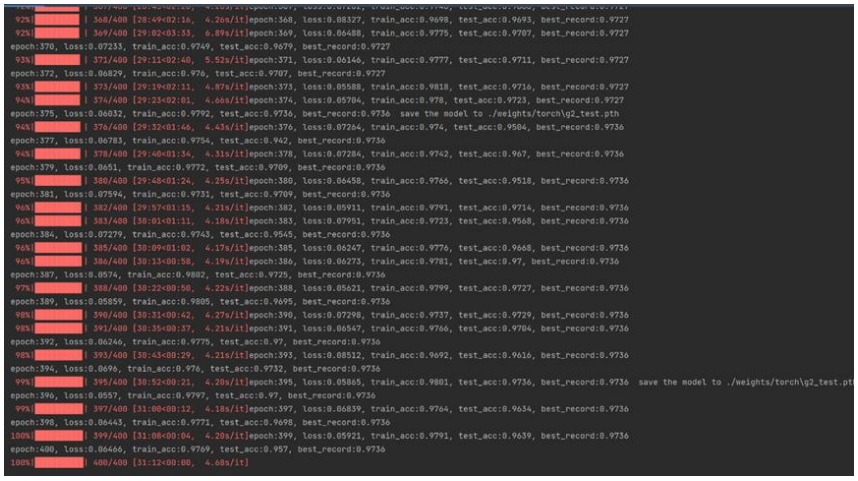


```python
import argparse
import os
from tqdm import tqdm, trange
from torch import optim
from os.path import join
from utils.model import *
from utils.logger import Logger
from utils.metric import *
from utils.dataset import Dataset
from configs.loader import load_yaml


def train_loop(model, train_iter, val_iter, optimizer, loss, epochs, device, add_weights_file):
    log_training_loss = []
    log_training_accuracy = []
    log_val_accuracy = []
    best_val_acc = 0.0

    model.to(device)
    model.train()
    for epoch in trange(1, epochs + 1):
        loss_sum, acc_sum, samples_sum = 0.0, 0.0, 0
        for X, y in train_iter:
            # Load data to GPU/cpu
            X = X.to(device)
            y = y.to(device)
            samples_num = X.shape[0]

            # BP
            output = model(X)
            log_softmax_output = torch.log(output)
            l = loss(log_softmax_output, y)
            optimizer.zero_grad()
            l.backward()
            optimizer.step()

            # Generating Log
            loss_sum += l.item() * samples_num
            acc_sum += calculate_accuracy(output, y) * samples_num
```

```
#--------Status---------#
Using device:  cpu
Dataset name:  DF
Dataset compression level:  raw
Dataset root directory:  ./datasets
Directory of real samples:  ['./datasets\\Origin\\raw']
Directory of fake samples:  ['./datasets\\DF\\raw']
Which branch of the LRNet to be trained:  all
#------Status End-------#

#----Train Arguments----#
Block size (frames per sample):  30
Batch size:  1024
Directory of model weights:  ./weights/torch
Training epochs (g1):  750
Learning rate (g1):  0.001
Saved weights name (g1):  g1_test.pth
Training epochs (g2):  400
Learning rate (g2):  0.001
Saved weights name (g2):  g2_test.pth
#--Train Arguments End--#

#----Model Arguments----#
RNN hidden units:  32
FC-layer hidden units:  64
Landmark dropout rate:  0.1
FC-layer dropout rate:  0.5
#--Model Arguments End--#

=========================
==       LOG END       ==
=========================
Loading data from:  ./datasets\Origin\raw\train
  9%|▉         | 68/720 [00:01<00:18, 35.29it/s]
```

图 5 复现的训练部分函数及其它部分代码

3. 成功训练并测试模型并得到重要数据



```
 92%|▉| 368/400 [28:49<02:16,  4.26s/it]epoch:368, loss:0.08327, train_acc:0.9698, test_acc:0.9693, best_record:0.9727
 92%|▉| 369/400 [29:02<03:33,  6.89s/it]epoch:369, loss:0.06488, train_acc:0.9775, test_acc:0.9707, best_record:0.9727
epoch:370, loss:0.07233, train_acc:0.9749, test_acc:0.9679, best_record:0.9727
 93%|▉| 371/400 [29:11<02:40,  5.52s/it]epoch:371, loss:0.06146, train_acc:0.9777, test_acc:0.9711, best_record:0.9727
epoch:372, loss:0.06829, train_acc:0.976, test_acc:0.9707, best_record:0.9727
 93%|▉| 373/400 [29:19<02:11,  4.87s/it]epoch:373, loss:0.05588, train_acc:0.9818, test_acc:0.9716, best_record:0.9727
 94%|▉| 374/400 [29:23<02:01,  4.66s/it]epoch:374, loss:0.05704, train_acc:0.978, test_acc:0.9723, best_record:0.9727
epoch:375, loss:0.06032, train_acc:0.9792, test_acc:0.9736, best_record:0.9736  save the model to ./weights/torch\g2_test.pth
 94%|▉| 376/400 [29:32<01:46,  4.43s/it]epoch:376, loss:0.07264, train_acc:0.974, test_acc:0.9584, best_record:0.9736
epoch:377, loss:0.0678, train_acc:0.9754, test_acc:0.942, best_record:0.9736
 94%|▉| 378/400 [29:40<01:34,  4.31s/it]epoch:378, loss:0.07284, train_acc:0.9742, test_acc:0.967, best_record:0.9736
epoch:379, loss:0.0651, train_acc:0.9772, test_acc:0.9709, best_record:0.9736
 95%|▉| 380/400 [29:48<01:24,  4.25s/it]epoch:380, loss:0.06458, train_acc:0.9766, test_acc:0.9518, best_record:0.9736
epoch:381, loss:0.07594, train_acc:0.9731, test_acc:0.9709, best_record:0.9736
 96%|▉| 382/400 [29:57<01:15,  4.21s/it]epoch:382, loss:0.05911, train_acc:0.9791, test_acc:0.9714, best_record:0.9736
 96%|▉| 383/400 [30:01<01:11,  4.18s/it]epoch:383, loss:0.07951, train_acc:0.9723, test_acc:0.9568, best_record:0.9736
epoch:384, loss:0.07279, train_acc:0.9743, test_acc:0.9545, best_record:0.9736
 96%|▉| 385/400 [30:09<01:02,  4.17s/it]epoch:385, loss:0.06247, train_acc:0.9776, test_acc:0.9668, best_record:0.9736
 96%|▉| 386/400 [30:13<00:58,  4.19s/it]epoch:386, loss:0.06273, train_acc:0.9781, test_acc:0.97, best_record:0.9736
epoch:387, loss:0.0574, train_acc:0.9802, test_acc:0.9725, best_record:0.9736
 97%|▉| 388/400 [30:22<00:50,  4.22s/it]epoch:388, loss:0.05621, train_acc:0.9799, test_acc:0.9727, best_record:0.9736
epoch:389, loss:0.05859, train_acc:0.9805, test_acc:0.9695, best_record:0.9736
 98%|▉| 390/400 [30:31<00:42,  4.27s/it]epoch:390, loss:0.07298, train_acc:0.9737, test_acc:0.9729, best_record:0.9736
 98%|▉| 391/400 [30:35<00:37,  4.21s/it]epoch:391, loss:0.06547, train_acc:0.9766, test_acc:0.9704, best_record:0.9736
epoch:392, loss:0.06246, train_acc:0.9775, test_acc:0.97, best_record:0.9736
 98%|▉| 393/400 [30:43<00:29,  4.21s/it]epoch:393, loss:0.08512, train_acc:0.9692, test_acc:0.9616, best_record:0.9736
epoch:394, loss:0.0696, train_acc:0.976, test_acc:0.9732, best_record:0.9736
 99%|▉| 395/400 [30:52<00:21,  4.20s/it]epoch:395, loss:0.05865, train_acc:0.9801, test_acc:0.9736, best_record:0.9736  save the model to ./weights/torch\g2_test.pt
epoch:396, loss:0.0557, train_acc:0.9797, test_acc:0.97, best_record:0.9736
 99%|▉| 397/400 [31:00<00:12,  4.18s/it]epoch:397, loss:0.06839, train_acc:0.9764, test_acc:0.9634, best_record:0.9736
epoch:398, loss:0.06443, train_acc:0.9771, test_acc:0.9698, best_record:0.9736
100%|▉| 399/400 [31:08<00:04,  4.20s/it]epoch:399, loss:0.05921, train_acc:0.9791, test_acc:0.9639, best_record:0.9736
epoch:400, loss:0.06466, train_acc:0.9769, test_acc:0.957, best_record:0.9736
100%|▉| 400/400 [31:12<00:00,  4.68s/it]
```

图 6 运行结果截图

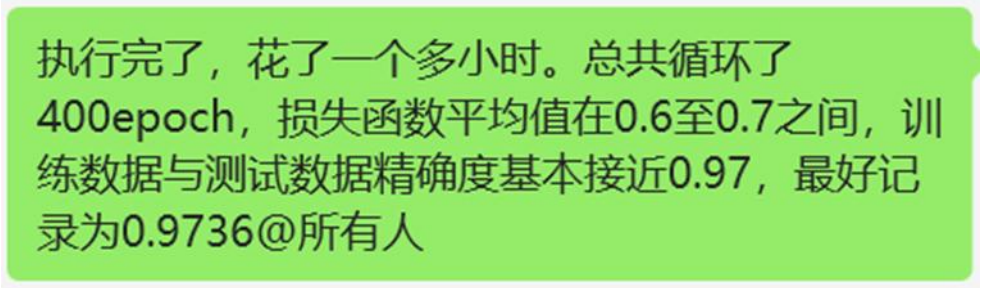


图 7 得到数据总结

五、待解决的问题

1. 测试模型性能，提高模型泛化能力，修改和完善系统的不足与缺陷。
2. 得到改进的模型嵌入到特定软件中，结合软件开发技术开发出具有泛用性的专业软件。
3. 整合子模块，检查系统错误。
4. 逐步优化迭代，整理实验数据并撰写结题报告。

六、参考文献

[1] Sun, Z., Han, Y., Hua, Z., Ruan, B. N., & Jia, W. (2024). Improving the Efficiency and Robustness of Deepfakes Detection through Precise Geometric Features. *In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. Shanghai Jiao Tong University, Beijing Normal University.

[2] Siwei Lyu and Hany Farid. "DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection." IEEE Access 7 (2019): 128611-128636.

[3] Yuezun Li, Ming-Ching Chang, Xiaoming Liu, and Siwei Lyu. "In Ictu Oculi: Exposing AI Generated Fake Face Videos by Detecting Eye Blinking." Proceedings of the IEEE International Conference on Computer Vision. 2019.

[4]David G ¨ uera and Edward J Delp. Deepfake video detection using recurrent neural networks. In IEEE International Con ference on Advanced Video and Signal Based Surveillance, pages 1 – 6, 2018. 2

[5]Yuezun Li and Siwei Lyu. Exposing deepfake videos by de tecting face warping artifacts. In IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 46 – 52, 2019.