HDTR: A Real-Time High-Definition Teeth Restoration Network for Arbitrary Talking Face Generation Methods

项目分：94

黄渤南[[1]](#endnote-0) 李正鑫[[2]](#endnote-1) 唐荣泽[[3]](#endnote-2) 王浩[[4]](#endnote-3)

## 一．组内分工及组内评价

**1.1组长对组员贡献评价（0-1.0）**

|  |  |
| --- | --- |
| 组员姓名 | 个人贡献度 |
| 王浩 | 1.0 |
| 唐荣泽 | 1.0 |
| 李正鑫 | 1.0 |

### 1.2组员对组长评价（0-1.0）

组长个人贡献度为各位组员对组长评价的平均值

|  |  |
| --- | --- |
| 组员姓名 | 个人贡献度 |
| 王浩 | 1.0 |
| 唐荣泽 | 1.0 |
| 李正鑫 | 1.0 |

## 二、实验报告

**2.1模型简介**

**2.1.1 模型解决的问题**

这篇论文提出了一个新的网络模型——HDTR-Net（High-Definition Teeth Restoration Network），主要是解决在生成“说话脸”视频时，牙齿区域的清晰度和自然同步问题，能够对视频中的牙齿和面部图像进行细节修复。论文中讨论的“说话脸生成（Talking Face Generation, TFG）”是一个音视频合成任务，目标是生成与音频同步的、自然的面部表情和口型。现有的TFG方法虽然能够生成高同步的嘴唇动作，但在生成牙齿和牙齿周围区域的清晰度上仍然存在困难。尤其是，当面部图像分辨率较低时，模型难以恢复清晰的牙齿细节，且很容易出现不同帧之间的不一致或像素失真。本篇论文的核心问题是如何在保证嘴唇同步的同时，提升牙齿的清晰度和周围区域的细节，并解决现有方法中的帧间连续性和处理速度问题。

### 2.1.2 模型使用的方法

为了解决上述问题，论文提出了HDTR-Net，一个能够在任意TFG方法中应用的通用高分辨率牙齿恢复网络。HDTR-Net包含两个主要组件：

* Fine-Grained Feature Fusion (FGFF) 模块：该模块通过融合细粒度的特征来恢复牙齿区域及其周围的高频细节。它还利用参考图像来指导模型有效地恢复这些高频细节，并在推理阶段提供像素连续性的提示。
* Decoder 模块：该模块负责从提取的细粒度特征中恢复高频细节，以确保牙齿和口型区域的清晰度。HDTR-Net的一个重要特点是其实时生成能力，能够快速恢复高分辨率的“说话脸”视频，在提高牙齿清晰度的同时保证同步和连贯性。

### 2.1.3 模型的效果

该模型通过大量实验验证了其效果，显示出HDTR-Net能够显著提高牙齿区域的清晰度，并在处理速度上超越了现有的基于超分辨率的面部修复方法。具体来说，HDTR-Net的推理速度比当前的超分辨率图像恢复方法快了300%。此外，实验结果表明，该方法可以在不同的TFG方法中进行有效应用，且能够解决现有方法在视觉质量方面的不足，尤其是在处理牙齿和其周围区域时。HDTR-Net在提高生成图像的视觉质量、尤其是牙齿细节的恢复、以及在实时视频生成中的应用上都取得了显著进展。

**2.2实验困难及解决方案**

配置环境时候出现以下问题报错；

FFmpeg命令可能包含无关或冲突的参数，这些参数导致了路径解析异常和文件加载失败等一系列问题。

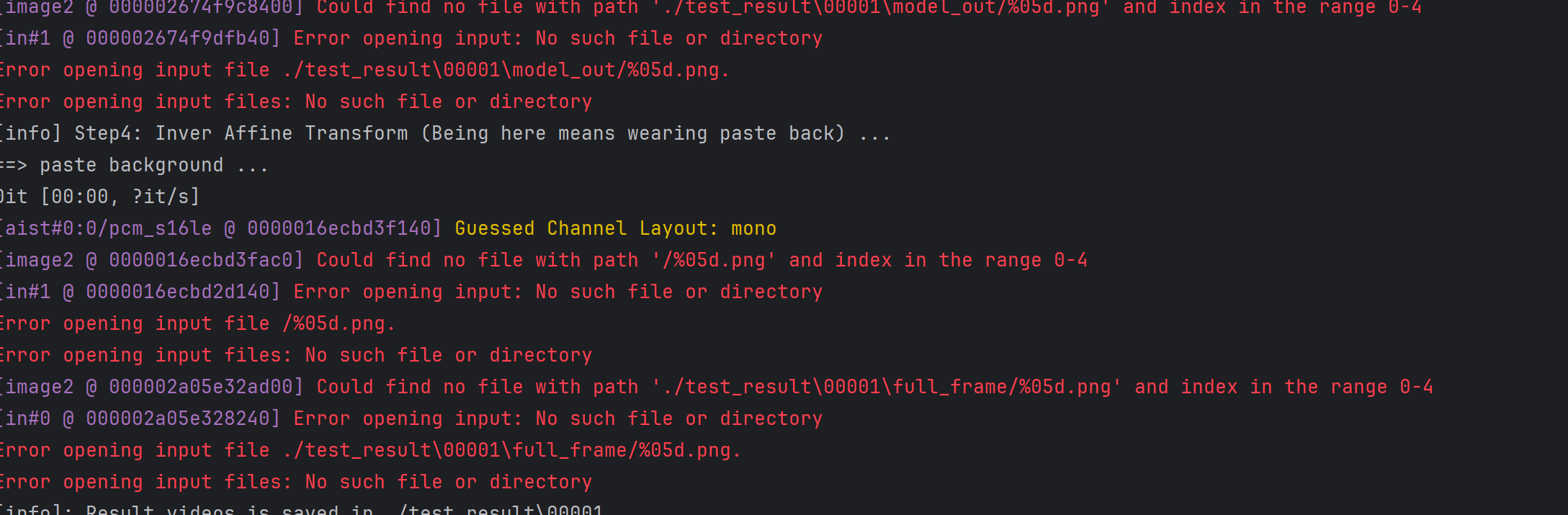


图1：配置实验环境的报错截图

解决方案:如图2所示，只需将FFmpeg命令中 -i 参数后的输入路径部分修改正确，特别是删除不必要的冗长路径字符串或检查路径是否匹配文件格式，问题解决。

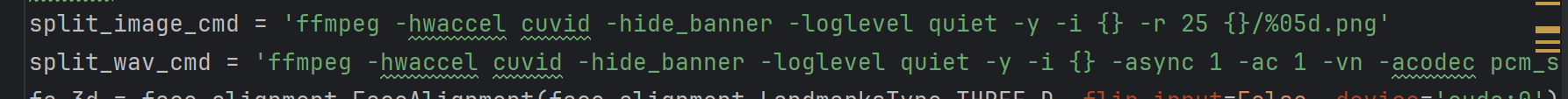


图2：相关实验环境截图

**2.3模型的定性及定量评价结果及可能改进方法**

### 2.3.1定量评估结果

#### 测试数据与指标概述

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Video | NIQE-Org | NIQE-Opt | PSNR | FID | SSIM | LSE-C | LSE-D |
| Jae-in | 17.269 | 17.280 | 44.326 | 1.816 | 0.997 | 4.450 | 8.683 |
| Lieu | 13.043 | 13.086 | 41.896 | 1.350 | 0.995 | 7.996 | 7.087 |
| Macron | 12.122 | 11.324 | 43.086 | 0.628 | 0.994 | 5.314 | 7.864 |
| May | 13.842 | 13.304 | 39.845 | 11.015 | 0.993 | 6.726 | 7.297 |
| Obama | 15.625 | 14.847 | 41.717 | 1.550 | 0.993 | 7.222 | 7.625 |
| Obama1 | 15.125 | 14.611 | 39.607 | 2.277 | 0.988 | 7.127 | 7.629 |
| Obama2 | 14.344 | 14.270 | 42.180 | 1.618 | 0.993 | 6.903 | 7.256 |
| Shaheen | 14.267 | 13.663 | 40.545 | 6.135 | 0.992 | 8.433 | 7.148 |

表一 ：评价数据集的各项指标测试数据

相关指标说明：

* NIQE（Natural Image Quality Evaluator，自然图像质量评价）：通常评分范围在0到100之间，评分的数值和图像质量成反比。评分的数值越低，表示图像质量越好。
* 其公式为：

(1)

其中：

* 是图像的特征向量。
* 是自然图像模型中的均值向量。
* 是自然图像模型中的协方差矩阵。
* 是协方差矩阵的逆矩阵。
* PSNR（Peak Signal-to-Noise Ratio，峰值信噪比）：表示图像信号的峰值（最大可能值）与噪声（失真）之间的比值。PSNR值越大，表示失真图像和原始图像之间的差异越小，图像质量越接近原始图像。

其公式为：

(2)

其中：

* 是图像的动态范围，通常对于8位图像来说，R=255（即像素值范围为0到255）。
* 是均方误差，表示原始图像和失真图像之间的平均平方差。
* FID（Fréchet Inception Distance，弗雷谢不匹配距离）：一种用于衡量生成图像质量的无参考评估指标，广泛应用于生成对抗网络和其他生成模型的评估。FID值越低，表示生成图像与真实图像的质量越接近。

其公式为：

(3)

其中：

* 和分别是真实图像和生成图像特征的均值。
* 和分别是真实图像和生成图像特征的协方差矩阵。
* 表示协方差矩阵的矩阵平方根。
* SSIM（Structural Similarity Index，结构相似度）：是一种衡量两幅图像相似度的标准化方法，旨在模仿人眼对图像质量的感知。SSIM的数值越接近1，图像相似度越高。

其公式为：

(4)

其中：

* 和分别表示两幅图像的像素值。
* 和是图像x和y的平均亮度（均值）。
* **和**是图像x和y的对比度（方差）。
* 是图像x和y之间的结构相似度（协方差）。
* 和是常数，用于防止分母为零。
* LSE-C（Least Squares Error with Constraints，带约束的最小二乘误差）：指的是在最小二乘误差基础上加入约束条件的优化方法。较小的LSE-C值表示图像处理结果与原图更为接近，说明图像质量较好。
* LSE-D（Least Squares Error with Data-driven Approach，数据驱动的最小二乘误差）：指的是在最小二乘误差方法的基础上使用数据驱动的方法来优化模型。较小的LSE-D值表明模型的预测结果与真实图像的差异较小，生成或恢复的图像质量较好。

#### 整体结果概述：

* NIQE：大多数视频在优化后，NIQE-Opt的值都略低于NIQE-Org，只有少数视频的值基本没有变化，这表明优化后的图像质量更接近自然图像。例如，“Macron”从12.122降到11.324，尽管变化不大，但已经表明优化后图像质量有微弱提升。
* PSNR：大部分视频的PSNR在优化后有所提升，显示优化方法有效提高了视频的保真度。特别是“Jae-in”的PSNR 高达44.326，优化后的结果在压缩失真方面表现出色，这表明优化后的图像几乎没有失真，与原图相似度极高。
* FID：优化后的FID值较低，表明生成图像与真实图像之间的分布差异较小。生成图像的质量较高，生成图像在视觉上和原始图像非常相似。
* SSIM：优化后的SSIM值通常很高，几乎接近1，显示出优化后的图像在结构、亮度、对比度等方面与原图非常相似，说明优化算法在保持图像细节和结构方面效果非常好。例如，“Jae-in”的SSIM为0.997，优化后的结构相似性保持得非常好。
* LSE-C：大多数视频在优化后的LSE-C值较低，这意味着优化算法更好地保留了图像的结构和纹理信息，处理后的图像在结构保持方面更为精确。
* LSE-D：优化后的LSE-D值有些较高，而有些视频保持在相对较低的水平。这表明不同视频在数据驱动的优化方面表现有所不同。

### 2.3.2定性评估结果

我们将我们的模型同原始数据集以及其他几个模型处理后的结果进行对比定性评估。

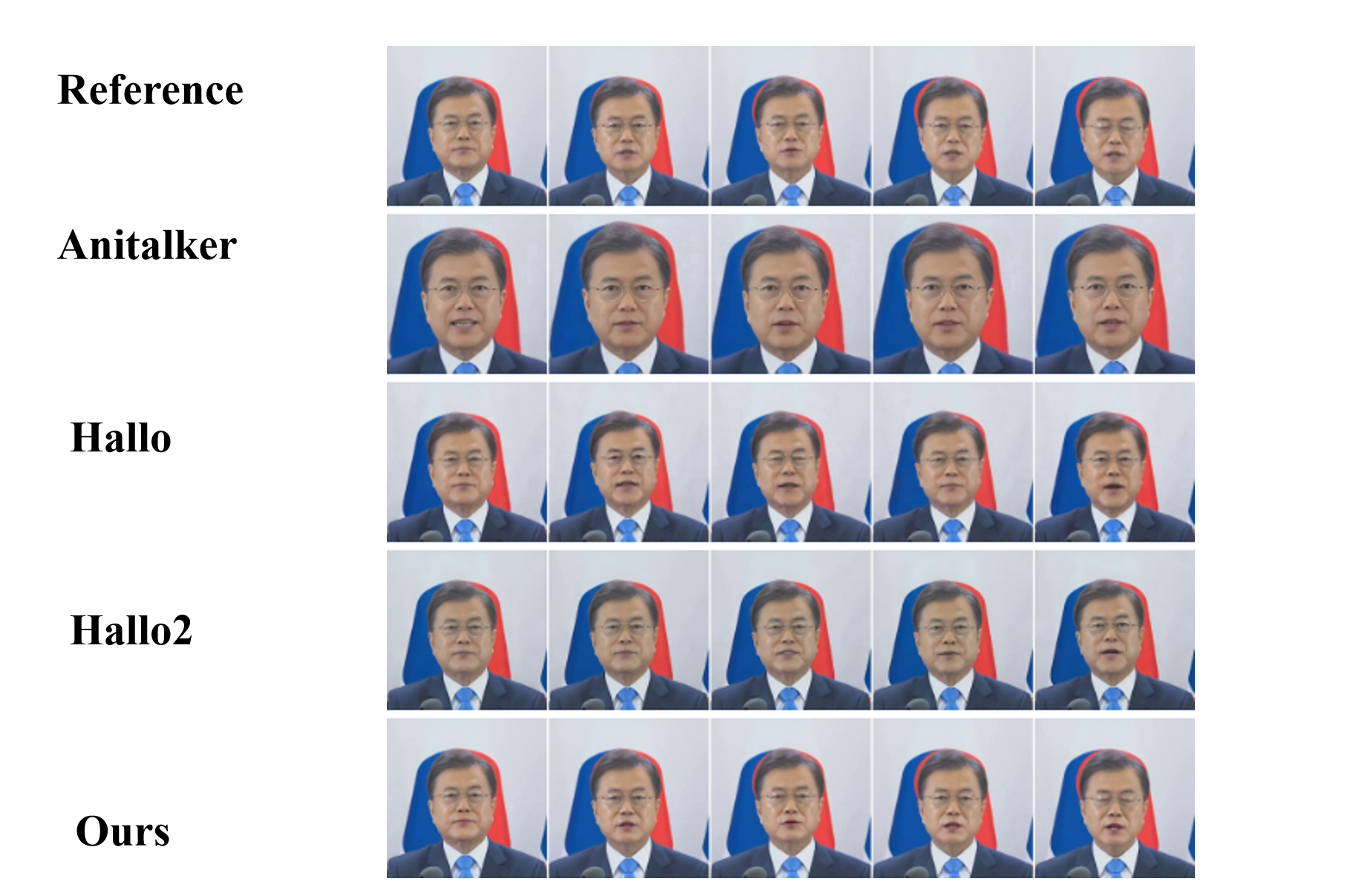


图3：不同模型对比图-Jae-in

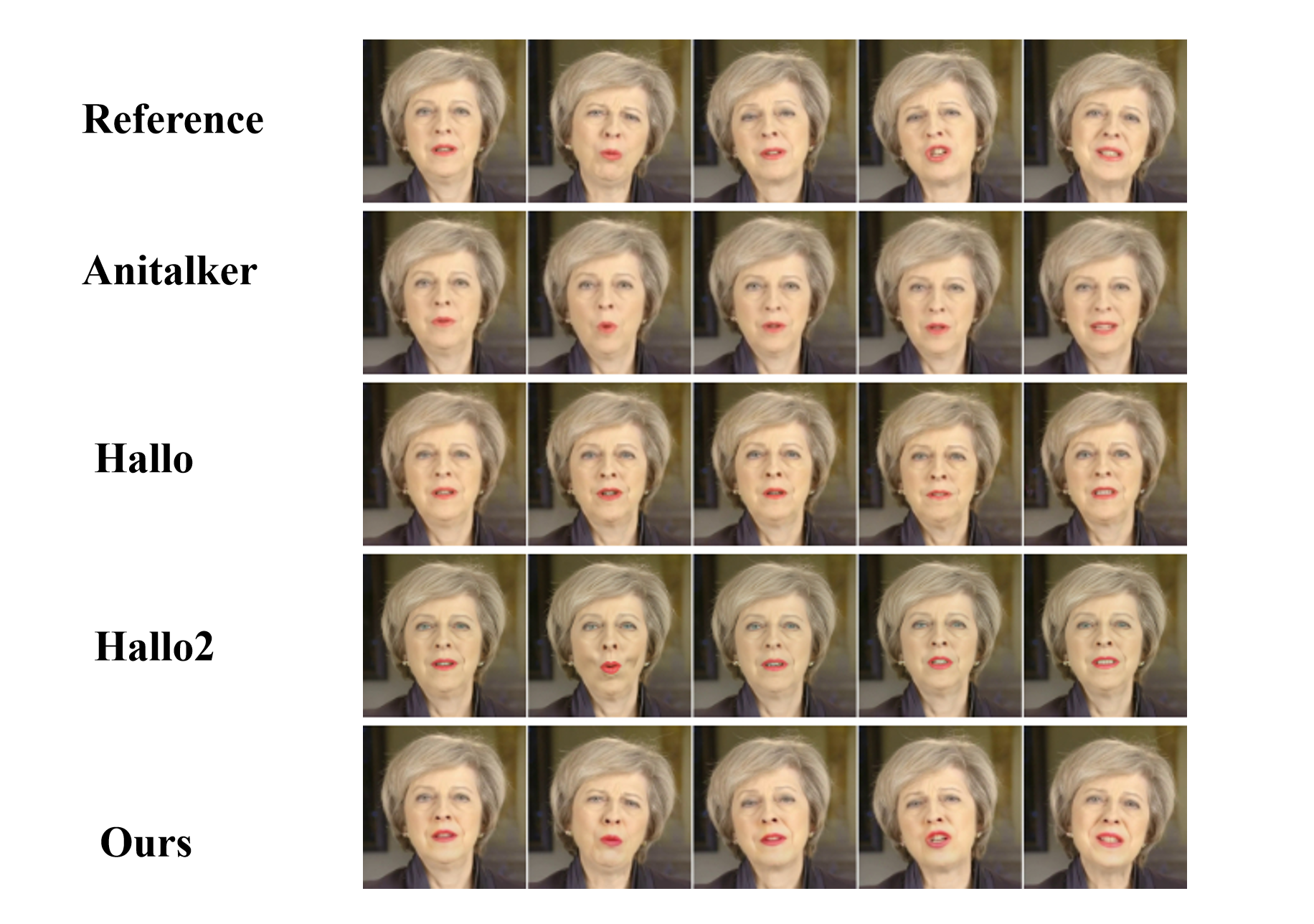


图4：不同模型对比图-May

### obama

图5：不同模型对比图-Obama

通过与其他小组的模型处理结果进行比较，可以发现我们的模型存在以下特点：

#### 细节恢复能力强：

我们的模型通过自注意力机制和生成对抗网络GAN的结合，使得模型在恢复牙齿细节的时候能够处理复杂的纹理和形状，恢复后的牙齿能够比较清晰，而且符合牙齿的自然结构。

#### 面部图像生成逼真：

我们的模型能够将相符后的牙齿图像与面部图像进行严丝合缝的融合，使得面部整体看起来协调逼真。

### 2.3.3模型现存问题

#### 依赖高质量输入图像：

HDTR-Net的性能在很大程度上依赖于输入图像的质量。尤其是在牙齿区域的细节恢复中，如果输入图像的质量较低，模型很难恢复出精细的细节。尤其在面部图像较模糊或噪声较多的情况下，牙齿区域的恢复效果可能会受到显著影响。HDTR-Net对图像中的低质量区域的恢复能力有限，就会导致最终生成的文件的面部图像看起来不够自然或真实。

#### 计算资源需求较高：

HDTR-Net在推理速度方面相较于其他超分辨率面部恢复方法有所提高，需要处理高分辨率的面部图像，并且在牙齿区域进行精细的纹理恢复，模型对计算资源的需求仍然较高。这意味着在推理阶段，尤其是在实时应用场景下，仍需要较强的计算能力。在我们自己的电脑上用GPU跑的话，需要跑十多个小时甚至一天多的时间，和论文中提高的处理时间差距很大。

#### 对其他面部区域的细节恢复有限：

HDTR-Net专注于恢复牙齿区域的细节，面部其他区域（如皮肤、眼睛、嘴唇等）的细节恢复效果可能不如专门针对这些区域设计的恢复方法。因为该模型的核心关注点是牙齿区域的恢复，它可能未能在面部的其他区域（尤其是在复杂表情或动态场景中）提供相同水平的细节恢复。在一些需要全脸细节恢复的应用中，HDTR-Net可能无法提供足够的精度，特别是在面部的动态动作（如笑、张嘴等）和表情变化较大的情况下。对于这些区域的恢复可能不如专门针对眼睛、嘴唇等区域优化的模型精细。

**2.4 总结心得及对课程的建议**

通过复现论文《HDTR-Net: A Real-Time High-Definition Teeth Restoration Network for Arbitrary Talking Face Generation Methods》并完成其Docker封装的实验，我们对该模型的设计原理、性能表现以及实际部署有了更加全面和深入的认识。HDTR-Net的核心在于通过Fine-Grained Feature Fusion (FGFF)模块和Decoder模块的结合，实现了对说话脸生成任务中牙齿区域的高分辨率修复，同时兼顾了帧间的连续性和嘴唇动作的自然同步性。这一设计充分体现了细粒度特征提取和融合技术的重要性。实验中，我们发现在处理牙齿和周围区域时，传统的超分辨率方法无法达到HDTR-Net所展示的细腻效果，尤其是在生成复杂口腔区域的高频细节时，该模型展现出了明显的优势。特别是HDTR-Net推理速度较传统方法快300%的特点，也让我们对深度学习模型在实时性和高效性方面的优化有了更深刻的认识。

实验的最大收获在于让我们对理论与实践的结合有了更全面的理解。从模型复现的角度来看，HDTR-Net的设计充分体现了在说话脸生成领域中对牙齿区域和周围特征的高效处理，这也让我们意识到，在视觉生成任务中，如何选择关键区域并针对性优化是一项非常重要的技术策略。从实际应用的角度来看，Docker封装让我们进一步认识到模型工程化部署的重要性，尤其是在不同硬件和环境下实现稳定运行的必要性。此外，通过比较实验数据，我们发现模型的生成质量虽然在大多数情况下表现优异，但对于高复杂度、多干扰的数据仍存在一定的不足，这也为后续的改进提供了启发。

本次实验让我们更加明确了高分辨率视频生成领域中的技术重点和难点。数据质量和模型适应性是实际应用中的关键问题，高质量的数据不仅能直接提升模型性能，也能有效降低实验调试的复杂度。

进行实验的过程中遇到了一些困难，刚开始部署环境时遇到了一点小困难，通过网上查阅相关资料很快就解决了，然后就是运行测试数据集，在我们电脑上跑的速度实在是很慢，特别是Macron的这个视频跑了好几天，和论文上说到的速度有点区别，应该是设备问题。

至于docker封装的问题，刚开始遇到了网络问题，通过修改代理得以解决。之后在环境配置上又遇到了诸多困难，在我们一一解决之后，项目成功封装。

对于评价指标，NIQE、PSNR、FID和SSIM指标的评价较为简单，基本上都有现成的代码或者直接调用库就能实现，对于LSE-C和LSE-D指标，网上可以找到的资料较少，我们详细学习了github上面一篇文章的测试方法然后进行测试。

对课程的建议：可以尽早尽快发布大作业，让我们尽早开始做，这样可以让我们能够一边学习理论知识，一边加以实践，我们觉得这样学习效果会更好，而且也有更充足的时间来完成我们的大作业，并加以创新改进。

1. 专业：计算机科学与技术 学号：1120223576 [↑](#endnote-ref-0)
2. 专业：计算机科学与技术 学号：1120222909 [↑](#endnote-ref-1)
3. 专业：人工智能 学号：1120221586 [↑](#endnote-ref-2)
4. 专业：软件工程（鸿蒙菁英班） 学号：1120223453 [↑](#endnote-ref-3)