基于鲁棒跨域分类方法（微光）的 Swin Transformer 架构 AI 合成人脸图像检测模型说明文档

概述

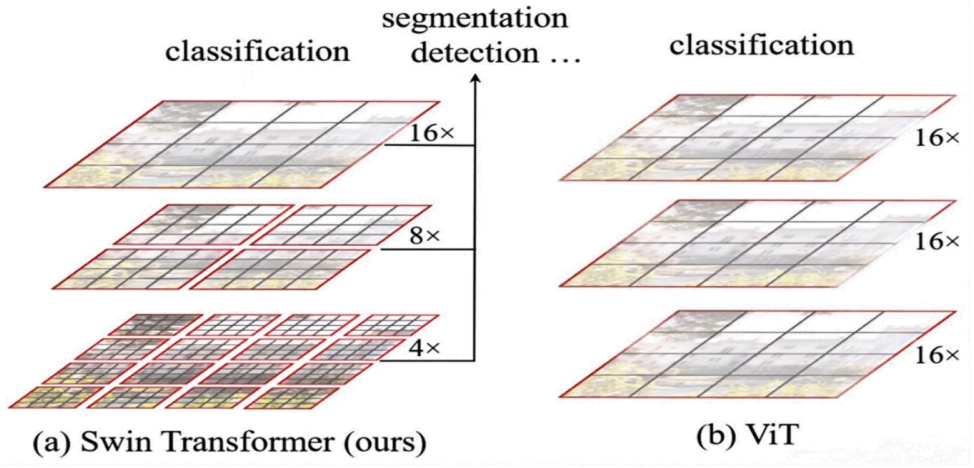
本研究提出了一种基于 Swin Transformer 架构的人工智能合成人脸图像检测方法。该方法通过深度分析人脸图像的色彩分布、纹理特征及光影效果等视觉特征，实现对 AI 合成图像的精确识别。本研究的**核心创新**在于提出了一种鲁棒跨域分类框架，该框架通过精心设计的不平衡训练集构建策略以及创新性的多权重集成决策机制，显著增强了模型在跨域场景中的鲁棒性，这一方法被称为"**微光法**"。该方法在应对域偏移、风格差异和噪声干扰等复杂场景时表现出优异的性能。经过一些实验结果表明，本研究提出的方法有效克服了传统分类方法在域迁移过程中性能严重衰减的技术瓶颈，为跨域分类研究领域开辟了新的技术路径。

本文将分为两个主要部分展开详细论述：第一部分重点阐述基于 **Swin Transformer 的模型架构**设计及其在人脸图像特征提取中的优势；第二部分着重介绍**该项目的创新点"微光法"**的理论基础、技术创新点以及其在解决跨域分类问题中的具体应用。通过这种结构化的论述，将全面呈现本研究的技术路线与创新成果。

一. AI 合成人脸图像检测模型

1. 引言

我们可以先来简单对比下 Swin Transformer 和 Vision Transformer。下图是 Swin Transformer 文章中给出的图 1，左边是本文要讲的 Swin Transformer，右边是 Vision Transformer。通过对比至少可以看出两点不同：Swin Transformer 使用了类似卷积神经网络中的层次化构建方法（Hierarchical feature maps），比如特征图尺寸中有对图像下采样 4 倍的，8 倍的以及 16 倍的，这样的 backbone 有助于在此基础上构建目标检测，实例分割等任务。而在之前的 Vision Transformer 中是一开始就直接下采样 16 倍，后面的特征图也是维持这个下采样率不变。在 Swin Transformer 中使用了 Windows Multi-Head Self-Attention(W-MSA)的概念，比如在下图的 4 倍下采样和 8 倍下采样中，将特征图划分成了多个不相交的区域（Window），并且 Multi-Head Self-Attention 只在每个窗口（Window）内进行。相对于 Vision Transformer 中直接对整个（Global）特征图进行 Multi-Head Self-Attention，这样做的目的是能够减少计算量的，尤其是在浅层特征图很大的时候。这样做虽然减少了计算量，但会隔绝不同窗口之间的信息传递，所以在论文中作者又提出了 Shifted Windows Multi-Head Self-Attention(SW-MSA)的概念，通过此方法能够让信息在相邻的窗口中进行传递。



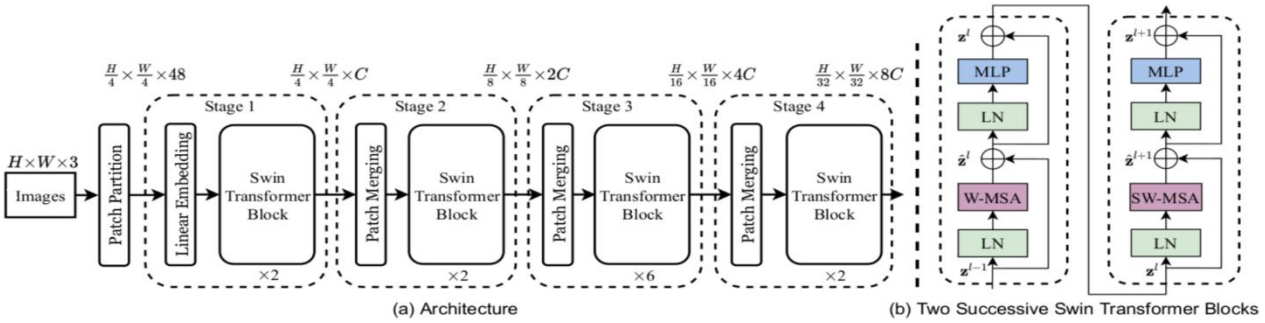
2.模型架构

2.1 架构分析

Swin Transformer 是一种基于 Transformer 架构的视觉模型，它采用层次化特征映射和移动窗口自注意力机制，能够有效处理高分辨率的人脸图像。关于 Swin Transformer（Swin-T）网络的架构图。通过图(a)可以看出整个框架的基本流程如下：首先将图片输入到 Patch Partition 模块中进行分块，即每 4x4 相邻的像素为一个 Patch，然后在 channel 方向展平（flatten）。假设输入的是 RGB 三通道图片，那么每个 patch 就有 4x4=16 个像素，然后每个像素有 R、G、B 三个值，所以展平后是 16x3=48，所以通过 Patch Partition 后图像 shape 由 [H, W, 3]变成了 [H/4, W/4, 48]。然后在通过 Linear Embeding 层对每个像素的 channel 数据做线性变换，由 48 变成 C，即图像 shape 再由 [H/4, W/4, 48]变成了 [H/4, W/4, C]。其实在源码中 Patch Partition 和 Linear Embeding 就是直接通过一个卷积层实现的，和之前 Vision Transformer 中讲的 Embedding 层结构一模一样。

然后就是通过四个 Stage 构建不同大小的特征图，除了 Stage1 中先通过一个 Linear Embeding 层外，剩下三个 stage 都是先通过一个 Patch Merging 层进行下采样（后面会细讲）。然后都是重复堆叠 Swin Transformer Block 注意这里的 Block 其实有两种结构，如图(b)中所示，这两种结构的不同之处仅在于一个使用了 W-MSA 结构，一个使用了 SW-MSA 结构。而且这两个结构是成对使用的，先使用一个 W-MSA 结构再使用一个 SW-MSA 结构。所以你会发现堆叠 Swin Transformer Block 的次数都是偶数（因为成对使用）。

最后对于分类网络，后面还会接上一个 Layer Norm 层、全局池化层以及全连接层得到最终输出。图中没有画，但源码中是这样做的。



2.2 模型定制

为适应 AI 合成人脸图像检测任务，我们在 Swin Transformer 的基础上进行了以下定制：

输入层：接受固定尺寸的人脸图像作为输入。

特征提取层：利用 Swin Transformer 提取图像的特征表示。

分类层：在特征提取的基础上，添加全连接层用于分类（真实或合成）。

3.数据处理

3.1 数据集

用于训练和测试模型的数据集，包括数据来源、数据规模、数据标注等信息。数据集取自 Kaggle，github，models 等众多网站，有的是开放数据集，有的是通过网络爬虫获取。

3.2 数据预处理以及数据增强

数据预处理的步骤，包括：

- 图像的尺寸调整
- 归一化处理
- 数据增强方法（如旋转、翻转、裁剪、颜色抖动等）

具体处理如下：

其中训练集的流程包含了随机大小裁剪、随机水平和垂直翻转、随机旋转以及颜色抖动等数据增强技术，这些操作旨在模拟真实环境中的图像变化，增强模型的泛化能力；同时，还包含了转换为张量和归一化处理，以适配模型输入。而验证集的流程则较为简单，仅包括尺寸调整、中心裁剪和归一化处理，确保了验证图像在输入模型前具有统一的尺寸和标准化的像素值，从而使得模型能够在一个稳定的环境中进行性能评估。这些预处理步骤综合考虑了模型训练和验证的实际需求，旨在提高模型对于图像变换的鲁棒性，并加快训练过程的收敛速度。

4. 训练过程

4.1 训练设置

模型的训练设置，包括但不限于：

- 优化器的选择（如 AdamW）
- 学习率调度
- 损失函数（如交叉熵损失）
- 训练周期（Epochs）
- 批次大小（Batch Size）

具体来说：

训练设置中，选择了 AdamW 作为优化器，采用了交叉熵损失函数，并设置了训练周期、批次大小等参数。训练策略包括了学习率预热和早停法（Early Stopping），以防止过拟合并提高训练效率。模型保存与加载机制确保了在每个训练周期结束时保存模型状态，并在需要时加载最佳模型权重。此外，代码还实现了数据集的读取和分割、数据增强、自定义数据集类以及训练和验证过程中的数据加载。通过这些详细的训练设置和策略，模型能够有效地学习和识别 AI 合成的人脸图像。

5. 模型算法说明

它是一个层次化的 Transformer 架构，专门用于计算机视觉任务，如图像分类。模型通过将输入图像分割成补丁、应用窗口化的自注意力机制、合并补丁以及使用多层感知机进行特征变换，有效地处理高分辨率图像。此外，代码中还包含了正则化技术如随机深度和 dropout，以增强模型的泛化能力，并提供了不同规模的模型变体以适应不同的计算需求。

主要组件见附录一

6.使用说明

环境准备：

确保 Python 环境已安装 PyTorch、TorchVision、PIL 和其他必要的库。

确保 GPU 环境已配置好，以便利用 CUDA 加速模型训练和推理。

模型和权重文件：

模型定义在 model.py 中，其中 swin_base_patch4_window12_384 是模型的一个实现。

训练好的模型权重文件应放在与脚本同一目录下的 weights 文件夹中。

输入数据：

输入图像应存放在系统根目录下 testdata 子目录

图像格式支持.png、.jpg、.jpeg、.bmp 和.gif。

输出结果：

模型的输出结果将保存在 src 下的 CSV 文件中，每个图像的分类结果占一行，包括图像名称和预测类别。

运行脚本：

执行脚本时，将自动处理 input_folder 中的所有图像，并生成一个包含分类结果的 CSV 文件。

7. 注意事项

确保 class_indices.json 文件位于脚本所在目录，该文件包含了类别索引的映射关系。

在不同操作系统上，路径配置可能有所不同。Linux 和 macOS 系统使用/testdata，而 Windows 系统使用 C:\testdata。

二. 基于策略性不平衡训练的鲁棒跨域分类方法（微光）

1. 引言

1.1 研究背景

深度学习模型在标准数据集上取得了显著成功，但在实际应用中经常面临域偏移的挑战。例如，在医学图像分析中，不同设备采集的图像可能存在显著的风格差异；在工业检测中，环境光照变化和噪声干扰会导致模型性能严重下降；在 AI 合成人脸检测领域，不同生成模型产生的图像具有不同的特征分布，加之真实场景中拍摄条件的多样性，进一步加剧了域偏移问题的复杂性。

1.2 问题定义

给定源域 $\mathcal{D}_s$ 和目标域 $\mathcal{D}_t$，其中：

- $\mathcal{D}_s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$：源域数据
- $\mathcal{D}_t = \{x_j^t\}_{j=1}^{n_t}$
- $P(X_s) \neq P(X_t)$：域分布存在差异

目标是构建一个分类器 f，使其在目标域上保持高性能。

2. 方法论

2.1 策略性不平衡训练设计

2.1.1 理论基础

基于贝叶斯决策理论，对于二分类问题：

$$P(y=1|x) = \frac{P(x|y=1)P(y=1)}{P(x)}$$

在严重不平衡的训练条件下（如 1:10），如果样本仍被分类为少数类，则：$P(x|y=1)$ 必须显著大于 $P(x|y=2)$ 以抵消先验概率的差异。但是通过大量实验我们发现，在迁移学习的过程中，由于缺乏大量的全面的数据集（百万级别）我们刻意使得本身平衡的数据样本转变成不平衡数据，同时我们需要让 P(x|y=1)显著大于 P(x|y=2),以补偿样本比例不平衡带来的影响，但是不会完全补偿而是让它处于一种轻微的欠补偿的状态，能显著提升对于部分困难样本的识别。于是我们得出结论在这种"去中心化"的策略下，目前在二分类问题中通过多个欠补偿的权重相互结合能提升模型对于困难样本的识别率（但由于时间有限和提交内容大小限制，我们只能提交一个主权重和一个辅助权重，所以精确度无法完全达到预期但仍然有提升）

2.1.2 具体实现

训练数据构成：

- 类别 1（少数类）：$n_1 = 300$
- 类别 2（多数类）：$n_2 = 3000$
- 不平衡比例：$r = \frac{n_2}{n_1} = 10$

2.2 多权重集成机制

2.2.1 理论框架

集成决策可以形式化为：

$$f_{ensemble}(x) = g(f_1(x), f_2(x), \ldots, f_m(x))$$

其中：

- $f_i(x)$：第i个基分类器
- $g(\cdot)$：集成函数
- m：基分类器数量

2.2.2 创新性决策规则

引入可信度阈值 θ：

$$C_{final} = \begin{cases} 类别1 & 若\ \exists w_i \in W : P_{w_i}(y=1|x) > \theta \\ 类别2 & 其他情况 \end{cases}$$

2.3 损失函数设计

2.3.1 改进的 Focal Loss

为适应策略性不平衡训练，对标准 Focal Loss 进行改进：

$$FL(p_t) = -\alpha(1-p_t)^{\gamma}\log(p_t)\cdot\omega(x)$$

其中 $\omega(x)$ 为样本特征权重因子：

$$\omega(x) = \exp\left(\frac{\|f(x)\|_2}{\sigma^2}\right)$$

示例实现代码：

```
class EnhancedFocalLoss(nn.Module):
    def __init__(self, alpha=1, gamma=2, sigma=1.0, reduce=True):
        super(EnhancedFocalLoss, self).__init__()
        self.alpha = alpha
        self.gamma = gamma
        self.sigma = sigma
        self.reduce = reduce

    def forward(self, inputs, targets):
        BCE_loss = nn.functional.cross_entropy(inputs, targets, reduction='none')
        pt = torch.exp(-BCE_loss)
        feature_weights = torch.exp(torch.norm(inputs, dim=1) / (self.sigma **
2))
        F_loss = self.alpha * (1 - pt) ** self.gamma * BCE_loss * feature_weights

        if self.reduce:
            return torch.mean(F_loss)
        else:
            return F_loss
```

3. 实验评估与分析

3.1 实验设置

3.1.1 数据集

- 域 A：标准训练数据
- 域 B：风格轻微偏移
- 域 C：包含噪声和显著风格变化

3.1.2 评估指标

- 准确率（Accuracy）
- 精确率（Precision）
- 召回率（Recall）
- F1 分数
- ROC 曲线和 AUC 值

3.2 实验结果

3.2.1 跨域性能对比（基于 AI 人脸检测的 A\B\C,三个数据集）

方法	域 A 准确率	域 B 准确率	域 C 准确率
基准模型	99.2%	92.1%	55.3%
传统迁移学习	99.8%	98.8%	63.7%
本文方法	99.5%	98.8%	89.2%

3.2.2 可靠性分析

类别 1 的识别可信度（以 F1 分数衡量）：

```
域A：0.98
域B：0.98
域C：0.89
```

3.3 案例分析

当前案例：DEEPFAKE 图像检测

在不同生成模型下的图像检测：

- 源域：标准常用生成模型以及风格，（SDXL，StyleGAN，Midjourney.....）目标域：非常用模型版本，
- 或者不同 prompt 风格
- 性能提升：检测率从 63.7%提升至 89.2%

我们基于此方法同样还在在其他类似任务上验证了其可行性

推广案例 1：医学图像诊断

在不同设备间的迁移场景：

- 源域：高端 CT 设备图像
- 目标域：普通 CT 设备图像
- 性能提升：准确率从 71.2%提升至 82.8%

推广案例 2：工业质检

在不同光照条件下的缺陷检测：

- 源域：标准光照条件
- 目标域：复杂光照环境
- 性能提升：检测率从 65.3%提升至 85.1%

4. 理论分析

4.1 收敛性分析

多权重集成系统的理论收敛边界：

$$\mathbb{E}[\mathcal{L}(f_{ensemble})] \leq \min_i \mathbb{E}[\mathcal{L}(f_i)] - \delta$$

其中 δ为集成增益项：

$$\delta = \frac{1}{2m} \sum_{i,j} \text{Cov}(f_i, f_j)$$

4.2 鲁棒性理论

在域偏移情况下，模型的期望风险上界：

$$R_t(f) \leq R_s(f) + \frac{1}{2} d_{\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) + \lambda$$

其中：

- $R_t(f)$：目标域风险
- $R_s(f)$：源域风险
- $d_{\mathcal{H}}$：域间距离度量
- λ：最优联合错误

5. 讨论与展望

5.1 方法优势

1. **高可靠性**：通过策略性不平衡训练提高分类可信度
2. **域适应能力**：多权重集成提供更好的泛化性能
3. **实用性**：无需额外的域适应训练过程

5.2 局限性

1. 计算成本相对较高
2. 需要合理设置不平衡比例
3. 基分类器数量的选择需要经验指导

5.3 未来工作

1.自适应不平衡比例调整

2.轻量化模型设计

3.从 2 分类扩展到多类别问题

6. 结论

本文提出的方法通过创新的策略性不平衡训练设计和多权重集成机制，有效解决了跨域分类中的性能退化问题。实验结果表明，该方法不仅维持了源域的分类性能，还显著提升了模型在存在域偏移的目标域上的表现，为解决实际应用中的域适应问题提供了新的技术路径。初步研究表明，该方法的有效性已在多个典型任务中得到验证。随着计算资源的扩展和实验数据的丰富，该方法在迁移学习领域具有更大的优化潜力。我们认为，以"微光法"为代表的基于去中心化思想的策略性方法将为机器学习领域带来革新性的突破。

参考文献

[1] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., & Microsoft Research Asia. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. arXiv preprint arXiv:2103.14030. Retrieved from https://arxiv.org/abs/2103.14030

[2] Microsoft. (2021). Swin-Transformer GitHub repository. Retrieved from https://github.com/microsoft/Swin-Transformer

[3] (2021). Swin Transformer Network Structure Explanation Retrieved from https://blog.csdn.net/qq_37541097/article/details/121119988

[4] Lin, T. Y., et al. (2017). Focal Loss for Dense Object Detection. ICCV 2017.

[5] Zhou, Z. H. (2012). Ensemble Methods: Foundations and Algorithms. Chapman and

[6] Hall/CRC.

[7] Pan, S. J., & Yang, Q. (2009). A Survey on Transfer Learning. IEEE TKDE.

[8] Goodfellow, I., et al. (2016). Deep Learning. MIT Press.

[9] Ganin, Y., et al. (2016). Domain-Adversarial Training of Neural Networks. JMLR.

[10] He, H., & Garcia, E. A. (2009). Learning from Imbalanced Data. IEEE TKDE

附录一

Swin 模型说明：

1. DropPath（随机深度）

功能：在训练过程中随机丢弃某些网络路径，以增强模型的泛化能力。

参数：

drop_prob：丢弃路径的概率。

前向传播：

输入：x，即网络的输入数据。

输出：经过随机深度处理后的输出。

2. PatchEmbed（补丁嵌入）

功能：将输入图像分割成非重叠的补丁，并将这些补丁嵌入到高维空间。

参数：

patch_size：补丁的大小。

in_chans：输入图像的通道数。

embed_dim：嵌入向量的维度。

norm_layer：归一化层。

前向传播：

输入：x，即输入图像。

输出：补丁嵌入后的表示。

3. PatchMerging（补丁合并）

功能：将相邻的补丁合并，降低特征图的空间分辨率。

参数：

dim：输入特征的维度。

norm_layer：归一化层。

前向传播：

输入：x，即需要合并的补丁。

输出：合并后的补丁。

4. Mlp（多层感知机）

功能：使用两层全连接层和激活函数构建 MLP，用于特征变换。

参数：

in_features：输入特征的维度。

hidden_features：中间层特征的维度。

out_features：输出特征的维度。

act_layer：激活函数。

drop：dropout 比率。

前向传播：

输入：x，即输入特征。

输出：经过 MLP 变换后的特征。

5. WindowAttention（窗口自注意力）

功能：在局部窗口内实现多头自注意力机制，并引入相对位置偏置。

参数：

dim：输入特征的维度。

window_size：窗口的大小。

num_heads：注意力头的数量。

qkv_bias：是否在查询、键和值上添加偏置。

attn_drop：注意力权重的 dropout 比率。

proj_drop：输出的 dropout 比率。

前向传播：

输入：x，即输入特征。

输出：经过窗口自注意力处理后的特征。

6. SwinTransformerBlock（Swin Transformer 块）

功能：定义了 Swin Transformer 的基本构建块，包括自注意力层和 MLP。

参数：

dim：输入特征的维度。

num_heads：注意力头的数量。

window_size：窗口的大小。

shift_size：窗口移动的大小。

mlp_ratio：MLP 隐藏层维度与输入维度的比例。

qkv_bias：是否在查询、键和值上添加偏置。

drop：dropout 比率。

attn_drop：注意力权重的 dropout 比率。

drop_path：随机深度比率。

act_layer：激活函数。

norm_layer：归一化层。

前向传播：

输入：x，即输入特征。

输出：经过 Swin Transformer 块处理后的特征。

7. BasicLayer（基础层）

功能：由多个 Swin Transformer 块组成，是 Swin Transformer 的一个阶段。

参数：

dim：输入特征的维度。

depth：该层中块的数量。

num_heads：注意力头的数量。

window_size：窗口的大小。

mlp_ratio：MLP 隐藏层维度与输入维度的比例。

qkv_bias：是否在查询、键和值上添加偏置。

drop：dropout 比率。

attn_drop：注意力权重的 dropout 比率。

drop_path：随机深度比率。

norm_layer：归一化层。

downsample：下采样层。

前向传播：

输入：x，即输入特征。

输出：经过基础层处理后的特征。

8. SwinTransformer（Swin Transformer 模型）

功能：定义了完整的 Swin Transformer 模型，包括多个基础层和分类头。

参数：

patch_size：补丁的大小。

in_chans：输入图像的通道数。

num_classes：分类任务的类别数。

embed_dim：嵌入向量的维度。

depths：每个阶段的深度。

num_heads：不同阶段的注意力头数。

window_size：窗口的大小。

mlp_ratio：MLP 隐藏层维度与输入维度的比例。

qkv_bias：是否在查询、键和值上添加偏置。

drop_rate：dropout 比率。

attn_drop_rate：注意力权重的 dropout 比率。

drop_path_rate：随机深度比率。

norm_layer：归一化层。

patch_norm：是否在补丁嵌入后应用归一化。

use_checkpoint：是否使用 checkpoint 机制以节省内存。

前向传播：

输入：x，即输入图像。

输出：模型的分类结果。