

专利说明书

技术领域

本发明属于人工智能与计算机视觉领域，特别是涉及一种基于深度学习技术的 CAD 审图匹配系统。该系统通过分析任意两幅或多幅图像或以 PDF 格式存储的 CAD 文件，判断其是否描述相同或相似的设计实体，应用于设计审核、工程复查及制造领域的图纸核对与验证。

背景技术

在现代工程设计中，CAD（计算机辅助设计）文件已成为各类设计项目中的标准化设计工具，被广泛应用于建筑、机械制造、电子电路设计等多个行业。随着项目的复杂性增加和全球化协作的深入，大型工程项目中设计图纸的数量和复杂程度急剧增加，导致审核与比较 CAD 图纸的工作量大幅上升。

传统的 CAD 图纸审核主要依赖于人工对比，这种方法不仅耗时耗力，还易出现人为错误。此外，设计团队往往位于不同的地理位置，使用不同的 CAD 软件，文件格式和版本也各不相同，这进一步增加了审核的复杂性和难度。

自动化的 CAD 图纸匹配和审核技术为此问题提供了一种解决方案。然而，现有的自动化系统普遍存在以下不足之处：

- 处理格式局限性：**现有系统难以处理多种格式的 CAD 文件，尤其是在不同软件之间的兼容性问题，系统往往不能识别不同格式文件中的相同设计。
- 尺度不一致：**不同 CAD 文件可能存在不同的尺度比例，传统算法无法有效对齐和比较尺度不一致的图纸。
- 低检测精度：**现有的检测算法在处理复杂结构和细节密集的 CAD 图纸时精度较低，容易遗漏或误判设计实体。
- 高计算开销：**许多现有系统计算开销较高，处理大型 CAD 图纸时速度较慢，无法满足实时性要求。

因此，开发一种能够高效处理多种格式、多种尺度的 CAD 图纸，且具备高精度和低计算开销的自动化审图匹配系统具有重要意义。

发明内容

本发明提供了一种高效、准确的 CAD 审图匹配系统，旨在克服现有技术的局限性。该系统通过多阶段处理流程，结合深度学习技术，实现对任意两幅或多幅 CAD 图纸的自动化对比与匹配，判定其是否描述相同的设计实体。系统主要包括以下创新点：

- 预训练模型的集成应用：**采用预训练的 YOLOv5 模型进行目标检测，通过对 CAD 图纸中目标实体的精准检测，提取关键设计元素进行后续分析。
- 双头输出模型架构：**使用共享特征提取骨干网络的双头输出模型进行图像匹配和分类，通过双头模型同时进行特征匹配和实体分类，提高系统的检测精度和效率。
- 教师-学生网络知识蒸馏策略：**系统通过知识蒸馏技术，将复杂的教师网络知识传递给轻量级学生网络，降低推理过程中的计算开销，同时保持模型的高精度。
- 滑窗检测与局域坐标转换：**针对超大尺寸 CAD 图纸，采用滑窗检测方法结合局域坐标到全局坐标的转换，实现大图像的高效目标检测，并进行全局非极大值抑制（NMS）以优化检测结果。
- 图像尺寸对齐预处理：**设计图纸预处理模块，通过调整图像尺寸对齐目标位置，确保不同尺寸图纸的匹配精度。
- 特征匹配与阈值优化：**基于余弦距离的特征匹配算法，结合 F1 分数进行阈值优化，实现对图纸中设计实体的高精度匹配。

本发明提供的系统能够在多种复杂场景下实现高效、准确的 CAD 图纸审核和匹配，广泛应用于设计、审核和生产制造领域，提高设计审核效率，降低错误风险。

附图说明

附图用于帮助理解本发明的具体实施例，并不作为对本发明范围的限制。附图中相同的标号表示相同或相似的部件，附图说明如下：

- 图1：**系统整体架构图，展示了系统的主要模块及其相互关系。
- 图2：**目标检测子模块的流程图，详细说明了 YOLOv5 模型在 CAD 图纸中进行目标检测的过程。

- 图3：图像匹配和分类子模块的流程图，描述了双头输出模型进行特征提取和匹配的具体步骤。
- 图4：教师-学生网络知识蒸馏训练策略示意图，展示了知识蒸馏过程中的数据流和优化策略。
- 图5：滑窗检测和局域坐标-全局坐标转换的流程图，详细描述了大尺寸图纸的检测和坐标转换过程。
- 图6：图像尺寸对齐预处理示意图，说明了对不同尺寸图纸进行预处理以实现尺寸对齐的过程。
- 图7：特征匹配子模块的流程图，展示了特征提取、余弦距离计算和阈值优化的步骤。

具体实施方式

实施例1：系统整体架构

本实施例详细描述了 CAD 审图匹配系统的整体架构。该系统由前端模块、后端模块和模型侧服务模块组成，各个模块之间通过标准化接口进行通信，实现 CAD 图纸的自动化匹配和审核。

前端模块

前端模块负责用户界面展示和交互功能，用户通过前端界面上传 CAD 文件，并查看匹配结果。前端模块的设计旨在提高用户体验，确保交互的流畅性和易用性。

- 文件上传与解析：**用户可以通过拖拽或选择文件的方式上传 CAD 图纸。系统支持多种文件格式，包括 PDF、DXF 和 DWG 等。上传后的文件经过解析器处理，提取出图纸的几何和文本信息。
- 结果展示与反馈：**前端界面提供匹配结果的可视化展示，包括检测出的目标实体、匹配相似度评分和分类结果。用户可以查看详细信息并提供反馈，以帮助优化系统性能。

后端模块

后端模块负责协调前端与模型侧服务模块之间的数据交互，并提供文件管理和处理结果的反馈功能。

- **请求处理器**：接收并解析前端的请求，调用相应的模型侧服务模块进行处理。请求处理器采用异步处理机制，支持高并发请求，提高系统响应速度。
- **数据管理器**：存储和管理上传的 CAD 文件及其相关信息。使用分布式文件系统（如 HDFS）确保数据的高可用性，并通过元数据数据库管理文件的版本信息。
- **结果反馈器**：将处理结果返回前端，包括匹配结果和置信度评分。结果反馈器还负责记录用户反馈，并存储在数据库中供后续分析。

模型侧服务模块

模型侧服务模块是系统的核心，负责执行 CAD 图纸的匹配和分析任务。该模块整合了多个深度学习模型和算法，以实现高效准确的匹配功能。

- **目标检测子模块**：使用预训练的 YOLOv5 模型对 CAD 图纸进行目标检测，提取出关键设计元素。此过程包括数据预处理、模型训练与优化，以及实时推理。
- **图像匹配和分类子模块**：基于共享特征提取骨干网络的双头输出模型进行图像匹配和分类。该子模块结合复杂损失函数进行优化，提高模型的精度和泛化能力。
- **知识蒸馏子模块**：采用教师-学生网络知识蒸馏策略，将教师网络的知识传递给学生网络，降低推理的计算开销。
- **滑窗检测与坐标转换子模块**：针对超大尺寸图纸，进行滑窗检测和局域坐标到全局坐标的转换，确保在大尺寸图纸上的准确检测。
- **图像尺寸对齐子模块**：通过调整图像尺寸对齐目标位置，确保不同尺寸图纸的匹配精度。
- **特征匹配子模块**：使用余弦距离和阈值优化方法进行特征匹配，确保设计实体的高精度匹配。

实施例2：目标检测子模块

本实施例详细描述了目标检测子模块的处理流程，主要基于 YOLOv5 模型进行 CAD 图纸中的目标检测。该子模块包括数据预处理、模型训练与优化、以及实时推理。

数据预处理

数据预处理是提高模型检测精度的关键步骤，通过一系列数据增强技术优化输入图像。

1. **自适应直方图均衡 (CLAHE)**：对输入图像进行对比度增强，公式如下：

$$\text{CLAHE}(I) = \sum_{i=0}^{L-1} \frac{n(i)}{N} \cdot (L - 1)$$

其中， I 是输入图像， $n(i)$ 是灰度级 i 的像素数， N 是总像素数， L 是灰度级数。

2. **随机水平翻转**：以概率 p 随机翻转图像，增强数据多样性：

$$P(\text{flip}) = p$$

3. **随机旋转**：对图像进行随机角度旋转，角度 θ 满足均匀分布：

$$\theta \sim \mathcal{U}(-\theta_{\max}, \theta_{\max})$$

4. **归一化处理**：将图像像素值归一化到 $[0, 1]$ 范围，提高模型训练稳定性：

$$I_{\text{norm}} = \frac{I - I_{\min}}{I_{\max} - I_{\min}}$$

其中， I_{\min} 和 I_{\max} 分别为图像的最小和最大像素值。

模型训练与优化

目标检测模型的训练与优化过程采用了多种策略，以提高检测的精度和稳定性。

1. **训练集与验证集划分**：按照 80% : 20% 的比例划分数数据集，确保模型在不同数据上的泛化能力。

2. **优化算法**：采用 Lookahead 和 AdaBelief 组合优化器进行模型训练：

$$\theta_{t+1} = \theta_t - \alpha \cdot \frac{m_t}{\sqrt{v_t} + \epsilon}$$

其中， θ_t 为模型参数， α 为学习率， m_t 和 v_t 为一阶和二阶动量估计， ϵ 为稳定项。

3. **学习率衰减策略**：使用 ReduceLROnPlateau 策略自动调整学习率：

$$\alpha_{\text{new}} = \begin{cases} \alpha & \text{if validation loss decreases} \\ \alpha \times \text{factor} & \text{otherwise} \end{cases}$$

4. **混合精度训练**：结合使用 32 位和 16 位浮点数进行训练，减少内存占用和提高计算效率。

实时推理

目标检测子模块在实际应用中通过 onnxruntime 实现高效的实时推理。

1. **模型加载**：通过 onnxruntime 加载 YOLOv5 模型，确保推理过程的高效性。
2. **非极大值抑制 (NMS)**：去除冗余的检测框，保留最具代表性的检测：

$$\text{IoU}(B_1, B_2) = \frac{|B_1 \cap B_2|}{|B_1 \cup B_2|}$$

其中， $|B_1 \cap B_2|$ 为两个检测框的交集面积， $|B_1 \cup B_2|$ 为并集面积。

3. **推理结果输出**：输出包括目标的位置、类别和置信度评分：

$$\text{Confidence} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

其中，TP 和 FP 分别为真实阳性和假阳性数量。

实施例3：图像匹配和分类子模块

本实施例详细描述了图像匹配和分类子模块的处理流程。该子模块使用共享特征提取骨干网络的双头输出模型实现图像匹配和分类。模型设计旨在提升处理效率和检测精度，通过精心设计的网络架构和优化算法实现。

模型构建

图像匹配和分类子模块的核心是一个双头输出模型，利用共享特征提取骨干网络来同时进行特征匹配和分类任务。

1. **特征提取骨干网络**：
 - 使用 ResNet50 作为特征提取骨干网络，其卷积结构被裁剪以适应 CAD 图纸的特性。
 - 卷积神经网络（CNN）通过一系列卷积层和池化层提取图像的层次化特征，特征向量通过全局平均池化层进行降维。

$$F(x) = W_k * x + b_k$$

其中， $F(x)$ 是提取的特征， W_k 和 b_k 分别为卷积核和偏置， $*$ 表示卷积操作。

2. 双头输出设计：

- **特征匹配头：**输出特征向量用于计算图像之间的相似度。
- **分类头：**输出每个目标的类别概率分布，用于识别图纸中不同类别的目标。

$$P(y|x) = \text{softmax}(W_f F(x) + b_f)$$

其中， $P(y|x)$ 是类别概率分布， W_f 和 b_f 是分类头的权重和偏置。

数据集构建

为了提高模型的泛化能力，数据集的构建和处理是关键所在。我们采用以下策略构建数据集：

1. 样本均衡与增广：

- 使用过采样和欠采样技术平衡数据集的类别分布，确保每个类别的数据量相对均匀。
- 数据增广技术包括随机裁剪、旋转、翻转和颜色抖动等，以提高模型在不同环境下的鲁棒性。

2. 复杂样本挖掘：

- 挖掘具有相似特征但不同标签的困难样本，构建三元组数据集以提升模型的判别能力。
- 三元组由锚点（Anchor）、正样本（Positive）和负样本（Negative）组成，用于优化模型的特征空间。

模型训练与优化

模型训练过程中采用多种复杂损失函数和优化策略，以提高模型的精度和鲁棒性。

1. 损失函数设计：

- **三元组损失：**用于优化正样本和负样本之间的距离。

$$\mathcal{L}_{\text{triplet}} = \max(0, d(f_a, f_p) - d(f_a, f_n) + \alpha)$$

其中， $d(f_a, f_p)$ 和 $d(f_a, f_n)$ 分别为锚点与正样本、负样本之间的欧氏距离， α 是边界阈值。

- **Arcface 损失**：增强同类样本的内聚性。

$$\mathcal{L}_{\text{arcface}} = -\log \frac{e^{s(\cos(\theta_{y_i} + m))}}{e^{s(\cos(\theta_{y_i} + m))} + \sum_{j \neq y_i} e^{s \cos(\theta_j)}}$$

其中， θ_{y_i} 是正确类别的角度， m 是角度边界， s 是缩放因子。

- **类别敏感交叉熵损失**：处理类别不平衡。

$$\mathcal{L}_{\text{ce}} = -\sum_i w_i \cdot y_i \log(\hat{y}_i)$$

其中， w_i 是类别权重， y_i 是真实标签， \hat{y}_i 是预测概率。

2. 训练策略：

- **Lookahead 和 AdaBelief 组合优化器**：提升模型收敛速度。

$$\theta_{t+1} = \theta_t - \alpha \cdot \frac{m_t}{\sqrt{v_t} + \epsilon}$$

其中， m_t 和 v_t 分别为一阶和二阶动量估计， α 是学习率。

- **混合精度训练**：结合使用 32 位和 16 位浮点数，提升计算效率和训练速度。

推理过程

在推理阶段，模型利用已训练的参数对输入图像进行特征提取和分类。

1. 特征提取：

- 通过共享特征提取骨干网络生成高维特征向量，随后用于匹配和分类。

$$F_{\text{inference}}(x) = W_k * x + b_k$$

2. 特征匹配：

- 使用余弦相似度度量特征向量之间的相似性。

$$\text{cosine_similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|}$$

3. 分类结果生成:

- 分类头输出目标的类别概率分布。

$$P(y|x) = \text{softmax}(W_f F(x) + b_f)$$

4. 结果输出:

- 输出匹配结果和分类结果，包括相似度评分、目标的分类信息。

实施例4：教师-学生网络知识蒸馏策略

本实施例详细描述了教师-学生网络知识蒸馏策略。该策略通过知识蒸馏技术将复杂的教师网络知识传递给轻量级学生网络，以降低计算开销，同时保持高推理精度。

教师网络训练

教师网络的训练采用多种损失函数组合，以学习复杂特征表示和提高泛化能力。

1. 复杂损失函数组合:

- BYOL 损失**: 通过自监督学习提升特征表示。

$$\mathcal{L}_{\text{BYOL}} = \|h(\text{proj}(\text{enc}(x_1))) - \text{stopgrad}(h(\text{proj}(\text{enc}(x_2))))\|_2^2$$

其中，enc 是编码器，proj 是投影器，h 是预测器， x_1 和 x_2 是不同视角的图像。

- Arcface 损失**: 提高同类样本的内聚性。

$$\mathcal{L}_{\text{arcface}} = -\log \frac{e^{s(\cos(\theta_{y_i} + m))}}{e^{s(\cos(\theta_{y_i} + m))} + \sum_{j \neq y_i} e^{s \cos(\theta_j)}}$$

- 类别敏感交叉熵损失**: 处理类别不平衡。

$$\mathcal{L}_{\text{ce}} = -\sum_i w_i \cdot y_i \log(\hat{y}_i)$$

2. 训练策略:

- 数据增广与复杂样本挖掘：提高对困难样本的判别能力。
- 优化器选择：使用组合优化器提升收敛速度和性能。

知识蒸馏过程

知识蒸馏过程中，教师网络的知识通过蒸馏损失传递给学生网络。

1. 蒸馏损失设计：

- 特征蒸馏损失：最小化教师网络和学生网络中间层特征之间的差异。

$$\mathcal{L}_{\text{feature_distill}} = \|F_{\text{teacher}} - F_{\text{student}}\|_2^2$$

- 对数几率蒸馏损失：最小化教师网络和学生网络输出的对数几率差异。

$$\mathcal{L}_{\text{logits_distill}} = \sum_i \text{KL}(\sigma(z_i^{\text{teacher}}/T), \sigma(z_i^{\text{student}}/T))$$

其中，KL 是 KL 散度， σ 是 softmax 函数， T 是温度参数， z_i 是对数几率。

2. 学生网络训练：

- 数据输入与特征对比：优化学生网络参数，使其逐渐学习和逼近教师网络的特征表示。
- 参数更新：结合特征蒸馏和对数几率蒸馏损失，更新学生网络的参数。

3. 推理性能优化：

- 模型裁剪与量化：通过裁剪冗余参数和量化模型权重，进一步降低计算复杂度。
- 实时推理效率提升：结合 onnxruntime 和硬件加速技术，优化推理速度。

实施例5：滑窗检测与局域坐标-全局坐标转换子模块

本实施例详细描述了滑窗检测与局域坐标-全局坐标转换子模块的处理流程。该模块专为处理超大尺寸的 CAD 图纸设计，旨在实现高效的目标检测和精确的坐标转换。

滑窗检测

对于超大尺寸的 CAD 图纸，滑窗检测方法通过将图纸分割为多个重叠的窗口，逐个进行目标检测，确保检测精度的同时降低了计算开销。

1. 窗口划分：

- 将大尺寸图像划分为若干个固定大小的窗口，每个窗口大小为 $W \times H$ ，步长为 Δx 和 Δy 。
- 重叠窗口策略： $\Delta x < W$ 和 $\Delta y < H$ ，确保每个目标实体至少被一个窗口完整覆盖。

$$\begin{aligned}x_{i+1} &= x_i + \Delta x, \\y_{j+1} &= y_j + \Delta y\end{aligned}$$

其中， (x_i, y_j) 为窗口的左上角坐标。

2. 逐窗口检测：

- 对每个窗口单独执行 YOLOv5 目标检测，提取关键目标实体。模型在较小的图像片段上运行，提高检测速度和精度。

3. 边界处理：

- 对于窗口边界区域可能导致的检测错误，采用特殊策略进行缓冲和调整，确保边缘目标不被截断。
- 使用权重调整策略对窗口边缘的检测结果进行加权平均，以缓解边缘效应。

局域坐标-全局坐标转换

在滑窗检测完成后，将局域坐标系中的检测结果转换为图纸的全局坐标系，确保结果的一致性。

1. 坐标转换公式：

- 计算局域坐标到全局坐标的映射，通过窗口的起始坐标 (x_i, y_j) 和检测框的局域坐标 $(x_{\text{local}}, y_{\text{local}})$ ，转换为全局坐标 $(x_{\text{global}}, y_{\text{global}})$ ：

$$\begin{aligned}x_{\text{global}} &= x_{\text{local}} + x_i, \\y_{\text{global}} &= y_{\text{local}} + y_j\end{aligned}$$

2. 全局坐标调整：

- 在全局坐标系中，消除窗口重叠区域的重复检测，通过非极大值抑制（NMS）算法去除冗余框。

$$\text{IoU}(B_1, B_2) = \frac{|B_1 \cap B_2|}{|B_1 \cup B_2|}$$

当 $\text{IoU}(B_1, B_2) > \text{threshold}$ 时，移除置信度较低的框。

3. 检测结果输出：

- 输出全局坐标系中的检测结果，包括目标位置、类别和置信度评分。

$$\text{Confidence} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

性能优化

为了确保滑窗检测与坐标转换模块的高效性，进行了多项性能优化。

1. 多线程并行处理：

- 采用多线程技术对滑窗检测进行并行加速。每个线程独立处理一个窗口，利用多核 CPU 和 GPU 提升计算效率。

2. 资源优化与负载均衡：

- 动态调整窗口参数和处理线程数，根据系统负载自动分配计算资源，确保资源的高效利用。

实施例6：图像尺寸对齐预处理子模块

本实施例详细描述了图像尺寸对齐预处理子模块的处理流程。该模块用于对不同尺寸的 CAD 图纸进行预处理，确保图纸的目标实体在匹配过程中位置对齐，从而提高匹配精度。

图像尺寸调整

图像尺寸对齐是图纸匹配的基础步骤，通过调整图像的大小，确保不同图纸间的比例一致。

1. 比例缩放：

- 对输入图像进行比例缩放，确保所有图纸以相同的分辨率进行处理。采用双线性插值算法实现平滑缩放。

$$I'_{x,y} = \sum_{i=0}^1 \sum_{j=0}^1 w(i,j) \cdot I(x+i, y+j)$$

其中， $w(i,j)$ 是插值权重， I 和 I' 分别为原始和缩放后的图像。

2. 长宽比保持：

- 确保图纸在缩放过程中保持其原始长宽比，避免几何变形。

$$\text{aspect_ratio} = \frac{\text{width}}{\text{height}}$$

使用填充（padding）技术调整边界，使图纸符合目标尺寸。

3. 裁剪与填充：

- 对超过目标尺寸的图纸进行裁剪，仅保留核心区域。对于小于目标尺寸的图纸，采用零填充（zero-padding）进行补齐。

$$I_{\text{padded}} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & I & \vdots \\ 0 & \cdots & 0 \end{bmatrix}$$

位置匹配

在尺寸对齐的基础上，对图纸中目标实体进行位置匹配，确保在不同图纸中目标的准确对齐。

1. 关键点检测：

- 使用特征点检测算法（如 ORB）提取图纸中的关键点。

$$\text{ORB_features}(I) = \sum_{p \in \text{keypoints}} \delta(p)$$

其中， $\delta(p)$ 表示图像关键点。

2. 特征描述符生成：

- 为每个关键点生成特征描述符，用于匹配过程。

$$D = \text{describe}(p)$$

其中, D 为描述符, describe 是描述符生成函数。

3. 几何变换与对齐:

- 计算图纸之间的几何变换矩阵 T , 包括平移、旋转和缩放:

$$T = \begin{bmatrix} s \cos(\theta) & -s \sin(\theta) & t_x \\ s \sin(\theta) & s \cos(\theta) & t_y \end{bmatrix}$$

其中, s 是缩放系数, θ 是旋转角度, t_x, t_y 是平移量。

4. 误差评估与调整:

- 计算对齐后的匹配误差, 调整对齐参数以提高准确性。

$$\text{error} = \frac{1}{N} \sum_{i=1}^N \|T(x_i) - x'_i\|^2$$

其中, x_i 和 x'_i 分别为原始和对齐后的坐标, N 是匹配点对数量。

性能与鲁棒性优化

为了确保图像尺寸对齐和位置匹配过程的高效性和鲁棒性, 进行了多项优化。

1. 并行处理:

- 使用并行处理技术对图纸进行批量预处理, 提升处理效率。

2. 鲁棒性提升:

- 采用多种特征提取和匹配算法组合, 提高对噪声和不完整图纸的鲁棒性。

3. 动态参数调节:

- 根据图纸的复杂程度和系统负载, 动态调整预处理参数和匹配策略。

实施例7: 特征匹配子模块

本实施例详细描述了特征匹配子模块的处理流程。该模块使用基于余弦距离的特征匹配算法, 并结合阈值优化策略, 确保 CAD 图纸中设计实体的高精度匹配。

特征提取

特征匹配的第一步是从输入图纸中提取高维特征向量，这些特征向量代表图纸中每个目标实体的视觉信息。

1. 共享特征提取骨干网络：

- 使用深度卷积神经网络（CNN）作为特征提取骨干网络。模型基于 ResNet50 等架构，经过调整以适应 CAD 图纸的特点。

$$F(x) = W_k * x + b_k$$

其中， $F(x)$ 是提取的特征， W_k 和 b_k 分别为卷积核和偏置， $*$ 表示卷积操作。

2. 特征标准化：

- 提取出的特征向量进行归一化处理，确保在不同尺度和强度下的稳定性。

$$f_i = \frac{f_i}{\|f_i\|}$$

其中， f_i 是特征向量， $\|\cdot\|$ 表示欧几里得范数。

3. 特征降维：

- 通过主成分分析（PCA）或线性判别分析（LDA）等降维技术，减少特征向量的维度，降低计算复杂度。

$$\hat{f}_i = W_{pca}^T f_i$$

其中， \hat{f}_i 是降维后的特征向量， W_{pca} 是 PCA 投影矩阵。

余弦距离计算

在特征提取完成后，计算特征向量之间的余弦距离，以判断不同图纸间目标实体的相似程度。

1. 距离计算公式：

- 使用余弦相似度度量两个特征向量的相似性：

$$\text{cosine_similarity}(f_i, f_j) = \frac{f_i \cdot f_j}{\|f_i\| \|f_j\|}$$

其中, f_i 和 f_j 是两个特征向量。

2. 批量计算:

- 通过向量化运算对所有特征向量进行批量计算, 提升计算效率。

$$S = F \cdot F^T$$

其中, S 是相似度矩阵, F 是特征矩阵。

3. 相似度矩阵生成:

- 将计算得到的相似度值存储在相似度矩阵中。矩阵中的每个元素表示一对特征向量之间的相似度。

阈值优化

在获得相似度矩阵后, 通过阈值优化策略确定最终的匹配结果。

1. 初始阈值设定:

- 根据先验知识和训练数据设定初始匹配阈值。初始阈值用于划分匹配和不匹配的特征对。

2. 阈值优化算法:

- 使用动态阈值优化算法, 通过调整阈值提高匹配的精度和召回率。算法根据 F1 分数进行调优, 以达到最佳匹配效果。

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

其中, $\text{Precision} = \frac{TP}{TP+FP}$, $\text{Recall} = \frac{TP}{TP+FN}$ 。

3. 误差评估与修正:

- 对匹配结果进行误差评估, 分析未匹配和错误匹配的特征对, 通过评估结果修正阈值参数, 进一步提高匹配精度。

匹配结果输出

在完成特征匹配和阈值优化后, 输出最终的匹配结果供后续处理和分析。

1. 结果格式化:

- 将匹配结果以标准化格式输出，包含匹配对的特征索引、相似度值和置信度评分。

$$\text{output} = \{(i, j, s_{ij}, c_{ij})\}$$

其中， i 和 j 为特征索引， s_{ij} 为相似度， c_{ij} 为置信度。

2. 匹配结果筛选：

- 结合置信度评分对匹配结果进行筛选和排序，保留最具代表性的匹配对。

3. 结果反馈与存储：

- 将匹配结果返回至后端模块进行展示和分析，匹配结果存储在数据库中，以供后续数据挖掘和模型优化使用。

性能与可靠性优化

为了确保特征匹配子模块的高效性和可靠性，进行了以下优化措施：

1. 加速技术应用：

- 使用 GPU 加速特征提取和余弦距离计算。通过 CUDA 和 cuDNN 等加速库，优化深度学习模型的推理速度。

2. 鲁棒性增强：

- 在特征提取和匹配过程中，结合多种特征描述符和匹配算法，提升系统在复杂场景下的鲁棒性。

3. 动态调整与适应：

- 实施动态调整策略，根据不同场景和输入数据自动调节匹配参数和策略。

实施例8：onnxruntime 实时推理模块**

本实施例详细描述了 onnxruntime 实时推理模块的工作流程。该模块为系统提供在线推理服务，确保 CAD 图纸匹配的实时性和高效性。

模型加载与初始化

onnxruntime 实时推理模块首先进行模型加载和初始化，确保推理过程的高效性和准确性。

1. 模型格式转换：

- 将深度学习模型从 PyTorch 或 TensorFlow 格式转换为 ONNX 格式，提高模型的跨平台兼容性。

$$\text{model}_{\text{ONNX}} = \text{convert}(\text{model}_{\text{PyTorch}})$$

2. 模型加载：

- 使用 onnxruntime 加载经过优化的 ONNX 模型。onnxruntime 提供了高效的推理引擎，支持多种硬件加速。

$$\text{session} = \text{onnxruntime.InferenceSession}(\text{model}_{\text{ONNX}})$$

3. 硬件加速配置：

- 根据系统硬件配置选择合适的加速方案，支持使用 CPU、GPU 和 TPU 等多种加速硬件。

实时推理流程

在完成模型加载和初始化后，模块进入实时推理阶段，对输入数据进行快速处理和结果输出。

1. 数据预处理：

- 对输入图纸数据进行标准化和归一化处理，确保数据在模型输入时符合预期格式。

$$x_{\text{norm}} = \frac{x - \mu}{\sigma}$$

其中， μ 和 σ 为数据的均值和标准差。

2. 模型推理：

- 利用 onnxruntime 提供的高效推理引擎对输入数据进行实时推理。

$$\hat{y} = \text{session.run}([x_{\text{norm}}])$$

3. 推理结果处理：

- 对模型输出的推理结果进行后处理，包括非极大值抑制（NMS）和结果筛选。

$$\text{output} = \text{NMS}(\hat{y})$$

结果输出与反馈

在推理完成后，模块将结果输出至后端系统，供用户查看和进一步分析。

1. 结果格式化与输出：

- 将推理结果以标准化格式输出，包含目标位置、类别和置信度评分等信息。

2. 用户反馈与系统优化：

- 收集用户对推理结果的反馈，结合用户意见对模型进行持续优化。

3. 推理日志记录：

- 记录推理过程中的重要信息，包括输入数据、推理时间和输出结果等，用于性能分析和故障诊断。

性能与可靠性优化

为确保 onnxruntime 实时推理模块的高效性和可靠性，进行了多项性能优化和可靠性增强。

1. 推理加速与优化：

- 利用图计算优化和算子融合技术，提高推理过程中的计算效率。

2. 动态负载均衡：

- 实施动态负载均衡策略，根据系统负载自动调整资源分配和任务调度。

3. 故障容错与恢复：

- 引入故障检测和自动恢复机制，确保系统在高负载下的稳定性和高效性。

结论

以上所述仅为本发明的优选实施例，应当指出，对于本领域的普通技术人员来说，在不脱离本发明原理的前提下，还可以做出若干改进和润饰，这些改进和润饰也应视为本发明的保护范围