图像相似性检查工具文档

**1. 功能说明**

**①图像相似度计算**

本工具的核心功能是计算图像之间的相似度。用户可以通过选择不同的图像对，并通过深度学习模型（ResNet-18）提取图像特征，从而计算出每对图像之间的相似度。该功能的主要步骤如下：

1、加载图片数据集：用户可以选择一个文件夹，其中包含多张图像。工具会自动加载该文件夹中的所有图像，并进行预处理，以便后续处理。

2、计算特征向量：对于每一张图像，工具使用 ResNet-18 模型提取图像的特征向量。通过这种方式，将每张图像转化为高维度的向量表示。

3、计算相似度：通过计算图像特征向量之间的余弦相似度，工具得出每一对图像之间的相似度。余弦相似度的值范围为 [-1, 1]，值越接近 1 表示两张图像越相似。

4、相似度排序：计算完成后，用户可以按相似度高低对图像进行排序，选择排序方式（高到低或低到高）。

**②图像显示与高亮**

查看图像：在计算完相似度后，工具会列出所有相似度对，用户可以点击任意一对，工具将显示两张图像。

高亮显示：当用户点击某张图像时，图像会被高亮显示。高亮显示通过更改图片的边框样式来实现，提供了更清晰的视觉反馈，帮助用户明确当前选中的图像。

**③筛选功能**

相似度筛选：用户可以输入最小和最大相似度区间，例如 0.5 到 0.6，工具会根据这个区间过滤出所有符合条件的图像对。

显示过滤结果：根据用户输入的筛选条件，工具会在结果列表中显示符合条件的图像对。

**④删除与重命名**

删除选中的图片：用户可以选择一张图像并删除。删除操作会更新相似度计算结果，并从列表中移除对应的相似度对。

重命名选中的图片：用户可以对选中的图像进行重命名操作。工具会自动处理图像的扩展名，确保新名字不冲突。

**⑤排序功能**

排序方式切换：用户可以选择将相似度对按从高到低或从低到高的顺序进行排序。这使得用户可以更灵活地查看最相似或最不相似的图像对。

**⑥图像展示与交互**

多图显示：在界面右侧显示两张图像，每张图像的下方显示其文件名。用户可以点击任意一张图像进行选中，选中后，图像的边框会被高亮显示，帮助用户确认当前选中的图片。

图像缩略图：显示时，图像会缩小到固定尺寸（如 224x224），以便于展示更多的图像。

**2. 如何运行工具**

下载项目文件：下载工具的源代码并解压到本地目录。

运行工具： 在命令行进入到项目的根目录，并运行以下命令启动工具：

python image\_similarity\_tool.py

①使用图形界面： 启动后，工具的图形界面将弹出，用户可以根据界面提供的功能按钮操作：

1、加载图像文件夹：点击“Load Image Folder”按钮，选择包含图像的文件夹。

2、筛选相似度：输入最小和最大相似度值，并点击“Filter”按钮，筛选符合条件的图像对。

3、查看相似度列表：相似度计算完成后，工具会列出所有相似度对，用户可以点击任意一对进行查看。

4、删除或重命名图像：在右侧界面中，用户可以选择要删除或重命名的图像，并点击相应按钮进行操作。

②交互方式

1、加载图像： 点击“Load Image Folder”按钮，选择一个包含图像的文件夹。工具会自动加载该文件夹中的所有图像，提取特征并计算相似度。

2、排序： 工具提供了“Sort Order: High → Low”按钮，用户可以通过点击切换排序顺序（从高到低或从低到高）。

3、查看图像对： 用户可以在结果列表中点击任意图像对，界面会在右侧显示这两张图像，且支持点击图像进行高亮选择。

4、删除或重命名图像： 用户选中某个图像后，可以点击“Delete Selected Image”按钮进行删除，或者点击“Rename Selected Image”按钮进行重命名。

**3. 设计思路**

**工具整体架构**

本工具的核心目标是实现图像相似度计算，并为用户提供一个易于交互的图形界面，以便他们能够浏览、筛选、查看相似图像对，并对图像进行管理（如删除和重命名）。

工具整体设计分为以下几个模块：

①图像加载与预处理模块：

该模块负责从用户选择的文件夹中加载图像，并进行预处理。图像预处理包括图像的尺寸调整和格式转换，以适应深度学习模型的输入要求。

本工具采用 ResNet-18 模型，该模型已经经过预训练，因此在加载图像后，我们会使用预训练的模型提取图像特征，并将其存储在内存中，以便后续的相似度计算。

②特征提取与相似度计算模块：

特征提取是图像相似度计算的核心步骤，采用 ResNet-18 模型从每张图像中提取出一个高维特征向量。该向量将图像的高阶特征信息（如纹理、形状、颜色等）编码在内。通过提取所有图像的特征向量，我们可以计算出两张图像之间的余弦相似度，该相似度衡量的是它们的特征向量在高维空间中的角度差异。

余弦相似度的值范围为 -1 到 1，其中 1 表示两张图像完全相同，-1 表示它们完全不同，0 表示它们之间没有相关性。

③图像展示与界面交互模块：

在图形界面中，用户可以看到所有相似度计算结果，并以图像对的形式进行展示。图像对会按照相似度值排序，用户可以查看排序结果、选择图像对，并高亮显示选中的图像。

图像展示支持点击交互，当用户点击某张图像时，该图像会被高亮显示，通过改变图像的边框样式（如添加黄色边框）来实现。

④筛选与排序模块：

用户可以通过设置最小和最大相似度值来过滤结果，只显示符合条件的图像对。筛选功能利用用户输入的相似度区间，通过比较相似度值筛选出符合要求的图像对，并实时更新显示的图像列表。

排序功能允许用户切换排序方式，选择将结果按相似度从高到低或从低到高排列。该功能使得用户可以根据需求查看最相似或最不相似的图像对。

⑤删除与重命名功能模块：

用户可以选择删除某张图像，当图像被删除后，工具会自动更新相似度计算结果，并移除相应的图像对。这使得删除图像变得方便快捷，不需要手动重新计算其他图像的相似度。

重命名功能使得用户可以根据需要修改图像的文件名，工具会在用户输入新名称时自动检查文件名是否存在冲突，并进行重命名操作。

**模块设计与实现**

**1. 图像加载与预处理模块**

**目的：**加载用户指定文件夹中的图像文件，并对图像进行适当的预处理，以便输入到模型中进行特征提取。

**设计：**

使用 OpenCV 库读取图像文件，通过文件扩展名过滤出支持的图像格式（如 .jpg、.png）。

使用 PIL 库将 OpenCV 读取的图像转化为 PIL 图像对象，以便进一步处理。

对图像进行 resize（大小调整为 224x224），并进行 归一化（将图像像素值缩放到 [0, 1] 范围内）以匹配模型的输入要求。

**2. 特征提取与相似度计算模块**

**目的：**利用深度学习模型（ResNet-18）提取图像的特征，并计算图像之间的相似度。

**设计：**

使用 torchvision 提供的预训练 ResNet-18 模型，并去除最后的全连接层，使得模型输出的是一个图像的特征向量。

使用 cosine\_similarity（来自 scikit-learn 库）计算图像特征向量之间的余弦相似度，得出每对图像的相似度。

特征提取是一个计算密集型过程，因此需要优化性能。例如，可以使用 批量处理，一次提取多个图像的特征，而不是每次只处理单张图像。

**3. 图像展示与界面交互模块**

**目的：**提供一个用户友好的图形界面，展示图像对的相似度计算结果，并允许用户进行交互操作。

**设计：**

使用 PyQt5 设计图形界面，其中包括加载文件夹、显示图像对、显示图像名称、选择和高亮图像等组件。

图像显示组件是 QLabel，通过修改其样式表来实现图像的高亮显示。每当用户点击某张图像时，图像的边框会变为黄色，以便用户确认选中的图像。

用户与界面的交互通过事件绑定来实现，如点击图像、选择排序方式、输入筛选条件等。

**4. 筛选与排序模块**

**目的：**根据用户输入的相似度范围对结果进行筛选，并根据相似度排序显示图像对。

**设计：**

筛选功能：用户输入最小和最大相似度，工具将根据输入的区间筛选出所有符合条件的图像对。

排序功能：通过切换排序按钮，用户可以选择将相似度从高到低或从低到高排列图像对。排序后，界面会更新以显示新的排序结果。

**5. 删除与重命名功能模块**

目的：允许用户删除或重命名选中的图像，并在操作后更新相似度列表。

设计：

删除功能会从图像库中移除图像文件，并自动更新相似度计算结果，确保删除的图像不会出现在相似度计算中。

重命名功能会弹出对话框，允许用户输入新的文件名，并确保文件名的唯一性。工具会检查新文件名是否与现有文件冲突，避免重命名失败。

**交互设计**

用户与工具的交互是通过简单而直观的图形界面完成的。所有操作都通过界面上的按钮和输入框进行，用户无需接触代码即可完成图像的加载、筛选、查看和管理等操作。交互设计的关键点包括：

**清晰的功能区分：**界面分为左侧和右侧，左侧用于显示结果列表、排序和筛选功能，右侧则用于显示图像和操作按钮（删除、重命名）。

**即时反馈：**当用户执行某个操作（如筛选、排序、点击图像）时，工具会即时更新界面，提供清晰的反馈，帮助用户更好地理解操作结果。

**高亮显示：**用户点击某张图像时，该图像会通过改变边框样式进行高亮显示，增强视觉反馈，帮助用户识别当前操作对象。

**4. 用到的匹配算法原理概述**

在本工具中，图像相似度匹配的核心算法是基于 深度学习模型提取图像特征，并利用 余弦相似度 来计算图像之间的相似度。具体来说，我们使用 ResNet-18 这一预训练的卷积神经网络（CNN）模型来提取图像的特征向量，然后通过 余弦相似度 来计算图像对之间的相似性。

**1. 图像特征提取：ResNet-18**

ResNet-18 模型概述

ResNet（Residual Networks）是一种深度卷积神经网络（CNN）架构，它通过引入“残差连接”（Residual Connections）解决了深层网络训练过程中梯度消失的问题，允许构建非常深的网络结构。在 ResNet-18 中，网络包含 18 层深度，设计上相对较浅，但依然能够从大量数据中学习到强大的特征表示。

ResNet-18 采用了如下架构：

残差块（Residual Block）：每个残差块由两个卷积层组成，卷积层之间存在跳跃连接，跳过某些层，这样可以更高效地训练深层网络。

全局平均池化（Global Average Pooling）：最后的全连接层之前，使用了全局平均池化层来降低空间维度，输出一个 512 维的特征向量。

特征提取的过程

通过加载一个预训练的 ResNet-18 模型（通常是使用 ImageNet 数据集进行预训练的），我们能够使用该网络提取图像的特征。预训练模型在处理各种图像时已经能够自动学习到图像的低层次到高层次的特征（如边缘、纹理、物体形状等）。当一张图像被输入到该模型时，它经过卷积层、池化层等层的处理后，最终会输出一个 512 维的特征向量。

这个特征向量代表了图像的高层次信息，如颜色、形状、纹理等，从而可以用于与其他图像进行相似度比较。

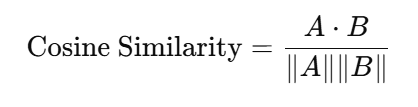
**2. 相似度计算：余弦相似度**

余弦相似度概述

余弦相似度是一种衡量两个向量之间相似度的度量方法，广泛用于文本挖掘和图像检索中。在余弦相似度的计算中，我们假设图像特征已经被转化为一个高维向量，余弦相似度的公式为：

余弦相似度

余弦相似度是一种常用的度量两个向量在角度上的相似度的方法。其公式为：



其中：A 和 B 是两个图像的特征向量（由 ResNet-18 提取）。

·表示向量的点积操作。

∥A∥ 和 ∥B∥ 分别是向量 A 和 B 的模长（即各自元素的平方和的平方根）。

余弦相似度的值范围从 -1 到 1：

1 表示两个向量完全相同，即两张图像非常相似。

0 表示两个向量正交，即两张图像之间没有相似性。

-1 表示两个向量完全不同，即两张图像极为不同。

**3. 选择 ResNet-18 及余弦相似度的原因**

1、ResNet-18 是高效且强大的图像特征提取模型：

ResNet-18 作为一个深度神经网络模型，已经在图像识别领域取得了显著的成果，尤其在处理各种类型的图像时，能够提取到有效的特征。

它的预训练版本已在大型数据集（如 ImageNet）上训练过，能够从大规模的图像中学习到丰富的通用特征，这使得它在很多实际应用中表现得非常好。

2、余弦相似度适合高维特征向量的比较：

在处理从深度学习模型中提取的高维特征时，余弦相似度是一种简单高效的计算方式。它能够通过向量的方向来衡量图像的相似度，不受图像尺度的影响，能够很好地适应图像特征的表示。

3、计算速度与准确性之间的平衡：

ResNet-18 作为一个相对较浅的模型（比 ResNet-50、ResNet-101 等深度模型浅），在保证特征提取准确性的同时，计算效率较高，适合在实际应用中进行快速的相似度计算。4. 图片仓库概要及处理耗时

**图片仓库概要**

在本工具的开发过程中，图像数据集的构建与管理至关重要。为了确保图像的多样性和质量，我们采用了 ComfyUI 生成图像，并利用 MATLAB 对生成的图像进行了后期处理，以增强图像的特征并增加数据集的复杂性。以下是图片仓库的构建过程和相关处理操作的详细描述。

1. 图像的获取与生成

1.1 使用 ComfyUI 生成图像 我们首先通过 ComfyUI 生成基础图像。ComfyUI 是一款基于文本提示生成图像的工具，它通过用户提供的文本描述（如“可爱的狗狗在草地上”）生成相应的图像。生成的图像种类多样，能够充分满足相似性分析所需的图像特征。

**·文生图**（Text-to-Image）：根据文本提示生成图像，例如输入“a cute puppy in a park”，ComfyUI 会生成符合该描述的图像。

**·图生图**（Image-to-Image）：通过输入一张已生成的图像，并结合文本描述进一步细化图像内容，生成更符合要求的图像。例如，增加光影效果或调整图像的颜色和细节。

通过这种方式，我们获得了大量具有多样化特征的图像数据，为后续的图像相似度计算提供了丰富的资源。

1.2 图片数据集的存储 生成的图像被保存在一个指定的目录中，这些图像成为工具中的图像仓库，供相似度计算和其他分析任务使用。每张图像都包含详细的描述和标注信息，便于后续的分类与管理。

2. 图像的后期处理

为了增加图像的复杂性和多样性，我们利用 MATLAB 对生成的图像进行了进一步处理，包括添加噪声、调整图像大小和颜色等操作。

2.1 添加噪声 MATLAB 提供了多种噪声生成方法，包括高斯噪声和椒盐噪声。我们通过添加噪声来模拟真实环境中的图像干扰，如摄像头噪声或低光环境下的图像失真。这一操作帮助增强图像的特征，提升相似度计算的鲁棒性。

噪声类型：使用 MATLAB 的 imnoise() 函数，添加高斯噪声。

2.2 **调整图像大小**所有图像都会通过 MATLAB 的imresize() 函数调整为统一的尺寸（224x224 像素），以符合深度学习模型的输入要求。调整大小后的图像确保了数据的一致性。

2.3 颜色与对比度调整 MATLAB 还被用于调整图像的颜色与对比度。我们通过 imadjust() 或 histeq() 函数增强图像的视觉效果，使图像更加生动鲜明。这不仅帮助增强图像的细节，也为后续的相似度计算提供了更多可识别的特征。

**调整颜色与对比度**：使用 imadjust() 函数对图像进行对比度调整。

3. 图像仓库的管理与处理耗时

所有生成并处理过的图像都会存储在指定的图像仓库中，该仓库包含了大量的图像数据。在进行图像相似性计算时，工具会加载这些图像并进行批量处理，提取每张图像的特征向量以供后续匹配与分析。

处理耗时：

图像生成时间：使用 ComfyUI 生成512\*512分辨率的图像的时间通常在 10秒之内，具体时间取决于图像的复杂度。

图像处理时间：MATLAB 进行图像处理（如添加噪声、调整大小和颜色）的时间相对较短，一般在 1-2 秒每张图像，但当处理大量图像时，整体处理时间可能会有所增加。

特征提取时间：特征提取是计算相似度时的关键步骤，使用预训练的 ResNet18 模型提取每张图像的特征向量的时间大约为 0.1-0.3 秒每张图像。

整体来说，图像获取与处理的时间相对较短，但当处理的图像数据量非常庞大时，可能需要进行优化以提高效率。

4. 总结

图像仓库包含了由 ComfyUI 生成并通过 MATLAB 后期处理的多样化图像数据集。通过这些图像，我们能够进行高效的图像相似性计算，并为后续分析提供了丰富的图像特征。这些图像数据经过多种增强操作，包括添加噪声、调整大小和颜色等，确保了图像的多样性与复杂性。在实际使用过程中，我们可以根据需要加载这些图像并快速计算其相似度，为用户提供精准的图像匹配结果。

**辅助编程工具**

**1.OpenCV (cv2)**

用于图像加载和处理，例如读取图像文件 (cv2.imread())、图像颜色转换（cv2.cvtColor()）以及调整图像大小（cv2.resize()）等。

**2.PyTorch 和 torchvision**

PyTorch：作为深度学习框架，使用了 PyTorch 的 torch 和 torchvision 库来处理图像特征提取，加载预训练的 ResNet-18 模型，并提取图像特征。

ResNet-18：预训练的卷积神经网络模型，用于提取图像的特征。在代码中，模型被加载并进行剪枝（去掉最后的分类层），只保留特征提取部分。

torchvision.transforms：用于对图像进行预处理，如调整图像大小并转换为 Tensor 格式，以便输入到模型中进行特征提取。

**3.scikit-learn (cosine\_similarity)**

用于计算图像之间的相似度，具体通过 余弦相似度 来衡量两个图像特征向量的相似性。余弦相似度是判断两个向量方向相似性的常用方法，值越接近 1 表示越相似，值越接近 0 表示越不相似。

**4.PyQt5**

用于构建图形用户界面 (GUI)，让用户可以与工具进行交互。PyQt5 提供了丰富的控件和布局系统，可以方便地创建按钮、文本框、列表框、标签等界面元素。

**5.PIL (Python Imaging Library)**

用于图像处理，将 OpenCV 读取的图像转换为 PIL.Image 对象，方便进行转换操作（如 Image.fromarray()）。

**6.OS**

用于与操作系统交互，主要用于获取文件路径、遍历文件夹、删除文件等操作。

**7.ChatGPT (OpenAI)**

作为智能辅助编程工具，ChatGPT 可以帮助开发者在编程过程中解决问题、生成代码、优化算法、调试错误并提供技术文档。通过自然语言处理能力，ChatGPT 能够理解开发者的需求并提供相关建议或示例代码，极大提高开发效率。

**尚未实现的功能及实现思路**

1. 批量图像处理

批量删除

思路：可以通过在GUI中提供一个多选框，让用户选择多个相似度对。然后，批量删除所选中的图片。

实现：

在相似度对列表中添加多选框或复选框。

在删除时，获取所有选中项的路径，逐一删除并更新界面。

批量重命名

思路：在GUI中增加一个批量重命名选项，允许用户输入统一的命名规则，并自动为选中的图像命名。

实现：

让用户输入重命名的规则（例如：基础名 + 编号）。

获取选中的图片路径，按顺序重命名，确保不会与现有文件冲突。