非密★

编号：

**面向图像处理软件的蜕变测试技术研究报告**

|  |
| --- |
| **项目名称： 基于属性的智能算法模型确定性测试技术** |
| **单位名称： 北京科技大学** |

2022年11月

**面向图像处理软件的蜕变测试技术研究报告**

拟 制: 

审 核: 

会 签: 

标准化:

批 准:

目 录

[1 引 言 1](#_Toc120482177)

[2 背景介绍 2](#_Toc120482178)

[2.1 课题背景 2](#_Toc120482179)

[2.2 研究现状 3](#_Toc120482180)

[3 研究需求 6](#_Toc120482181)

[3.1 研究意义 6](#_Toc120482182)

[3.2 问题识别 6](#_Toc120482183)

[4 相关技术 7](#_Toc120482184)

[4.1 软件测试 7](#_Toc120482185)

[4.2 蜕变测试 7](#_Toc120482186)

[4.3 CycleGAN基本原理 9](#_Toc120482187)

[4.4 小结 10](#_Toc120482188)

[5 研究总体思路 11](#_Toc120482189)

[6 图像处理软件的蜕变测试支持工具设计与实现 15](#_Toc120482190)

[6.1 需求分析 15](#_Toc120482191)

[6.2 总体设计 16](#_Toc120482192)

[6.3 详细设计 17](#_Toc120482193)

[6.3.1 前端设计 17](#_Toc120482194)

[6.3.2 后端设计 19](#_Toc120482195)

[6.4 工具实现与演示 23](#_Toc120482196)

[6.5 小结 27](#_Toc120482197)

[参考文献 29](#_Toc120482198)

1. 引 言

计算机软件不断发展，实用性越来越强，广泛应用于各行各业。与此同时软件的规模不断壮大，复杂程度上升，软件的质量保障工作越来越艰难。如果无法对软件的质量进行充分有效的评估，很可能导致无法挽回的损失。例如，由于飞机的机动特性增强系统存在软件故障，导致了波音737MAX型客机失事。软件的质量直接影响了人们的日常生活与社会经济运作，保障软件的质量具有重要的现实意义。

软件测试是一种重要的软件质量保证技术，传统的软件测试技术要求在构造测试用例时给出预期输出，通过比较执行软件后的实际输出和预期输出是否相等来判断待测软件是否存在缺陷[1]。

日常生活中许多关键系统都依靠图像处理应用程序，例如医学成像、生物识别、自动驾驶导航等。测试图像处理应用程序是一项非常重要的任务，也是一项具有挑战性的工作。图像处理应用程序通常利用测试图像进行人工测试，测试人员视觉检查测试结果，决定测试是否通过。这种测试方法存在两个主要问题：

1. 测试不是自动化的；
2. 存在测试预期问题。

蜕变测试（Metamorphic Testing）[2]是一种黑盒测试方法，该方法依据待测程序的特性构造蜕变关系，即待测程序的多次执行结果之间应满足的特殊关系，通过判断程序是否违反蜕变关系评估其正确性。该方法不需要提前构造预期输出，解决了测试判定问题，可以实现测试过程自动化与测试结果评估。

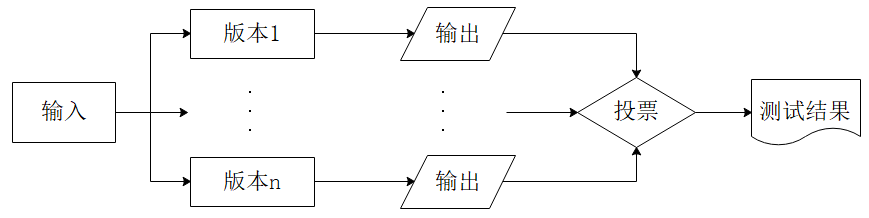
1. 背景介绍
   1. 课题背景

图像处理应用程序在社会中变得越来越普遍，人们对图像处理软件的依赖性逐渐增强，确保它们的质量越来越重要。图像处理应用程序通常由测试人员人工检查测试结果，决定测试是否通过。这种测试方法存在两个主要问题：

1. 测试不是自动化的。为了使得测试高效、低成本，测试过程与测试结果评估应尽可能自动化。
2. 存在测试预期问题[3]。程序的预期输出很难构造，测试结果评估复杂，传统的软件测试技术并不适用。

针对测试预期问题，研究人员已经提出以下一些方法：

1. N版本编程[4]是以规格说明为基础，为待测程序编写多个不同的实现，使用相同的测试用例执行这些不同的实现，最后以投票的方式判定占比多的运行结果为正确输出，N版本编程原理如图1所示。N版本编程方法应用的前提是有多个对同一规格说明的不同实现，实际测试时很难实现这一要求。
2. 断言[5]是指以正确的程序状态、变量初始化、变量值、程序输出的上界或下界等属性作为断言条件，这些断言条件必须都满足才能正确地执行程序。即使待测程序的预取输出是未知的，只要程序有任何违反断言条件的行为都表示待测程序中存在错误。因此断言检查可以用来测试存在预期问题的软件。断言技术人工成本较高，难以实现自动化，并且无法在没有待测程序源码的情况下应用。



**图1 N版本编程原理图**

1. 统计测试[6]是一种根据输出统计分布判断测试是否通过的测试方法。Meyer等人[7]采用统计测试的自动测试评价方法作为测试是否通过的决策手段，已经成功应用于图像处理和图像分析等多个领域。该方法只有在测试结果能够分类统计的情况下才有效，因此适用范围较为有限。
2. 蜕变测试[2]是一种黑盒测试方法，该方法依据待测程序的属性构造蜕变关系，即待测程序的多次执行结果之间应满足的特殊关系，通过判断程序是否违反蜕变关系评估其正确性。Chen等人针对数值型程序的测试问题提出蜕变测试方法[7]。该方法不需要提前构造预期输出，能够有效地解决测试预期问题，可以实现测试过程自动化与测试结果评估。与其他能够解决测试预期问题的方法相比，蜕变测试存在以下几点优势[1]：
3. 概念简单；
4. 容易实现；
5. 能够实现自动化；
6. 低成本。

蜕变测试方法应用于图像处理软件能够缓解测试预期问题，有效提高测试过程自动化程度。

* 1. 研究现状

蜕变测试方法可以高效地解决测试预期问题，已经应用于多种图像处理程序的测试研究中。Mayer等人[7]以二值图像的欧几里得距离变换程序为研究对象，使用随机二值图像模型和布尔模型两种原始测试用例生成技术，采用7种蜕变关系进行蜕变测试，最后使用变异分析评估两种测试用例生成模型的适用性和7种蜕变关系的缺陷检测能力。实验结果表明，使用旋转操作生成衍生测试用例的蜕变关系检测缺陷最有效。

Jameel等人[8]以二值图像膨胀算法作为研究对象，提出膨胀算法的8种蜕变关系，采用变异分析评估蜕变关系的有效性。在实验的8种蜕变关系中，两种是一般的图像处理操作，即左右翻转与上下翻转，六种是膨胀算法的特殊蜕变关系。文献[9]实现了二值图像膨胀程序自动测试框架，使得测试过程与测试结果评估自动化。

图像分类与目标检测是计算机视觉领域的两个基本问题[10]。在解决图像分类问题中常用到各种机器学习方法，但是实验人员几乎总是将不正确的分类结果归因于训练数据不足，督促开发人员添加多样化的训练数据，而忽略了对程序进行测试的重要性。

**表1 图像处理软件蜕变关系**

|  |  |
| --- | --- |
| **研究对象** | **使用的蜕变关系** |
| 二值图像的欧几里得距离变换程序 | 1. 二值图像逆时针旋转90°后距离变换与距离变换后逆时针旋转90°的结果一致； 2. 二值图像转置后距离变换与距离变换后转置的结果一致； 3. 二值图像左右（上下）翻转后距离变换与距离变换后左右（上下）翻转的结果一致。 |
| 二值图像膨胀算法 | 1. 二值图像膨胀后左右（上下）翻转与左右（上下）翻转后膨胀的结果一致； 2. 二值图像膨胀后的补码与补码图像的侵蚀一致；二值图像侵蚀的补码与补码图像的膨胀一致； 3. 二值图像与结构化元素位置互换，膨胀结果一致。 |
| k-最近邻分类器（KNN）和Naïve贝叶斯分类器（NBC） | 1. 对所有训练集和测试集中特征的值做仿射变换后，预测结果保持一致； 2. 对所有训练集和测试集中的类别标签以相同的顺序打乱，预测结果保持一致； 3. 对所有训练集和测试集中的部分特征以相同的顺序打乱，预测结果一致； 4. 在测试集的特征中添加与类别无关的特征值，预测结果一致； 5. 在测试集数据中添加与类别相关度高的属性，预测结果一致； 6. 将分类器对某测试集数据的分类结果添加到训练集，重新训练出的分类器预测结果一致； 7. 假设分类器对某数据的预测类别为a，将训练集中类别与其相同的数据复制，重新训练出的分类器预测结果一致； 8. 假设分类器对某数据的预测类别为a，将训练集中部分类别与其相同的数据改为类别b后添加到训练集，重新训练出的分类器对该数据分类仍为a； 9. 假设分类器对某数据的预测类别为a，将训练集中部分类别与其不同的数据改为类别b，重新训练出的分类器对该数据的分类结果仍为a； 10. 假设分类器对某数据的预测类别为a，将训练集中所有类别为b的数据删除，重新训练出的分类器对该数据的分类结果仍为a； 11. 假设分类器对某数据的预测类别为a，将训练集中部分类别为b的数据删除，重新训练出的分类器对该数据的分类结果仍为a。 |

**表1 图像处理软件蜕变关系（续）**

|  |  |
| --- | --- |
| **研究对象** | **使用的蜕变关系** |
| 支持向量机（SVM）分类器与残差网络（ResNet）分类器 | 1. 对所有训练集和测试集的类别标签以相同的顺序打乱，预测结果一致； 2. 打乱训练集，预测结果一致； 3. 图像RGB通道交换，预测结果一致； 4. 训练集与测试集的卷积运算顺序做一致性改变后，预测结果一致； 5. 测试集标准化，预测结果一致。 |
| 深度学习图像分类器 | 1. 改变图像对比度不影响预测结果； 2. 改变图像亮度不影响预测结果； 3. 图像模糊处理不影响预测结果； 4. 改变图像锐度不影响预测结果； 5. 图像添加微小的噪声扰乱不影响预测结果； 6. 图像仿射变换不影响预测结果。 |

目前蜕变测试方法已经广泛应用在机器学习分类算法相关研究上。Xie等人[11]以开源机器学习包Weka中的k-最近邻分类器（KNN）和Naïve贝叶斯分类器（NBC）作为实验对象，通过研究蜕变测试在机器学习程序上的应用，提出期望机器学习分类算法满足的11种蜕变关系。在实验中根据一般机器学习算法提出的蜕变关系，不一定是所有机器学习算法的必要属性，但是仍可作为验证所选算法是否适合求解问题的依据。

Dwarakanath等人[12]使用蜕变测试方法对一个残差网络（ResNet）实现的分类器进行研究，采用4种蜕变关系对分类器进行蜕变测试，最后用变异分析评估蜕变关系的有效性，实验结果表明此方法能够有效地检测出分类器的缺陷。

Braiek等人[13]对深度学习在计算机视觉领域的应用进行研究，使用蜕变测试方法检测程序的缺陷。研究人员提出一些图像的基本操作可作为蜕变关系，包括改变图像对比度与亮度、模糊处理、改变锐度、图像缩放、图像旋转、图像翻转等，这些基本操作可以模拟光学仪器的自然输入差异，测试程序对图像变形的敏感度，敏感度越低说明待测程序的性能越好。

表1所示为总结图像处理软件中研究对象及其使用的蜕变关系[7,8,11-14]。

1. 研究需求
   1. 研究意义

图像处理技术在自动驾驶导航等安全攸关的领域中应用广泛，图像处理应用程序的质量问题日益受到重视。本项目的主要目的是基于蜕变测试的基本原理，设计实现一个面向图像处理软件的蜕变测试支持工具，以可视化的方式呈现蜕变关系与测试结果。主要研究意义如下：

1. 使用蜕变测试方法缓解测试判定问题、实现测试过程自动化，避免复杂的人工任务，提高效率；
2. 通过可视化的方式展示蜕变关系与测试结果，增强该工具的易用性。
   1. 问题识别

蜕变测试广泛应用于多类图像处理软件的测试，包括二值图像处理程序、手写数字识别类程序、目标识别软件、自动驾驶系统的道路检测软件等。一方面，蜕变测试在图像处理软件测试方面取得了丰富的成果；另一方面，现有研究工作存在如下不足之处：

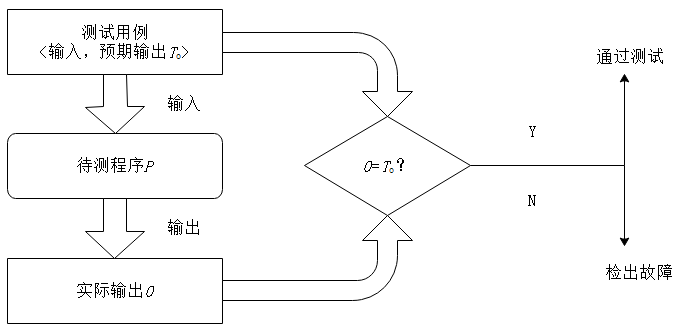
1. 缺少一个面向不同类型的图像处理软件的统一蜕变测试框架：现有工作局限于特定类型的图像处理软件，如二值图像处理程序、图像分类程序、目标识别软件；
2. 难以产生符合真实应用场景的测试图片：自动驾驶与导航等系统的实际应用场景存在雨天、雪天等复杂的天气环境，现有蜕变关系主要对测试图片进行简单的变换操作，难以在这类简单变换操作的基础上有效产生符合真实应用场景的测试图片。
3. 相关技术

介绍软件测试、蜕变测试以及CycleGAN等相关技术概念。

* 1. 软件测试

软件测试是一种重要的软件质量保障技术，目的是发现软件中潜藏的缺陷。传统的软件测试技术需要在构造测试用例时给出预期输出，通过比较与实际输出是否相等来得出测试结论，主要过程如图2所示，具体步骤如下：

1. 选择执行的测试用例，测试用例应包括输入与预期输出*T*o；
2. 以测试用例中的输入执行程序*P*，获得实际输出*O*；
3. 对比预期输出*T*o与实际输出*O*，如果两者不相等，则证明程序中存在错误；反之如果两者相等，则测试通过。测试通过仅代表没有检测出错误，不能断定程序中没有错误。



**图2 传统软件测试过程**

很多情况下，待测软件的预期输出很难甚至无法构造，即存在测试预期问题。对于存在测试预期问题的软件，传统软件测试技术无法应用，软件的质量无法得到保障。

* 1. 蜕变测试

蜕变测试依据待测程序的属性构造蜕变关系，以原始测试用例为基础生成衍生测试用例，通过比较对应输出结果之间是否满足蜕变关系来判断测试是否通过[15]。在一些情况下，传统的软件测试技术存在测试预期问题，即预期输出很难构造，测试结果无法或者很难评估。蜕变测试方法不需要构造预期输出，有效地缓解了测试预期问题，因此蜕变测试方法被广泛应用于科学计算、交互式系统、CPU调度程序[16]、网络安全[17]、计算机图形[18]等领域中。

**定义1** 蜕变关系[2]

假设存在待测程序*P*，程序实现了函数*f*，*x*1, *x*2, …, *x*n (n>1)是函数*f*的n次输入值，并且*f*(*x*1), *f*(*x*2), …, *f*(*x*n)是这n次输入所对应的函数返回值。假设*x*1, *x*2, …, *x*n满足关系*r*的同时，它们的函数返回值*f*(*x*1), *f*(*x*2), …, *f*(*x*n)满足关系*rf*，则称(*r*, *rf*)为程序*P*的**蜕变关系**。

**定义2** 原始/衍生测试用例[2]

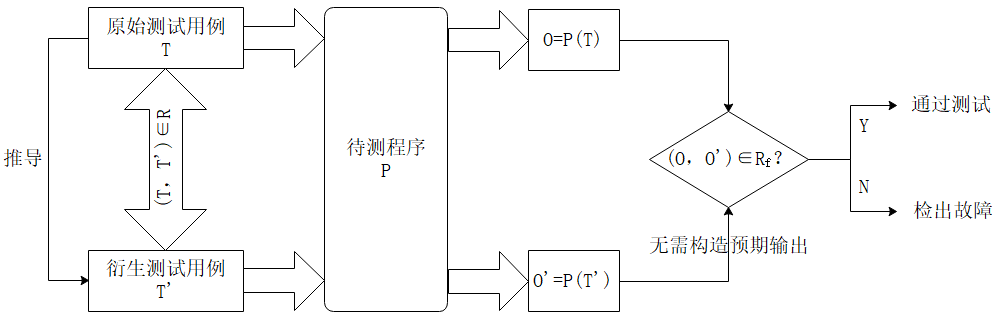
在蜕变测试方法中，起始的测试用例被称为**原始测试用例**。原始测试用例可以使用符号执行、特殊值法、随机值法、CPM等方法生成。以原始测试用例为基础，根据构造的蜕变关系生成的新测试用例被称为**衍生测试用例**。

蜕变关系是蜕变测试方法的核心依据[19]，也是解决测试预期问题的独特之处，构造蜕变关系是蜕变测试方法不可或缺的一个步骤。不同程序的蜕变关系通常没有通用性，为了构造更有效的蜕变关系，需要对待测程序进行具体分析。

在蜕变关系构造正确的情况下，蜕变测试的结果应包含以下3种可能性：

1. 待测程序没有缺陷，多次执行结果满足蜕变关系，测试通过；
2. 待测程序有缺陷，多次执行结果不满足蜕变关系，测试不通过；
3. 待测程序有缺陷，多次执行结果满足蜕变关系，没有检测出缺陷，测试通过。

有效的蜕变关系应使得蜕变测试结果更多地出现第一、二种情况，更少地出现第三种情况。



**图3 蜕变测试原理图**

蜕变测试原理如图3所示，蜕变测试过程可以分为以下几个阶段[2]：

1. 使用测试用例生成技术构造原始测试用例；
2. 构造多条蜕变关系；
3. 基于蜕变关系，使用成功通过测试的原始测试用例生成衍生测试用例；
4. 执行待测软件，检测多次输出结果是否满足蜕变关系，判断软件是否存在错误。
   1. CycleGAN基本原理

生成式对抗网络（GAN，Generative Adversarial Nets）[20]包含生成器（G，Generator）和判别器（D，Discriminator）两种模型，通过两种模型之间的对抗博弈，使GAN能够生成一些以假乱真的合成数据。生成器G负责生成新的图像，训练过程中试图让生成的图像以假乱真；判别器D负责判断图像是否真实，训练过程中试图准确分辨真实的图像和生成的图像。生成器G和判别器D的训练过程如式1所示：

（式1）

其中，表示来源于真实数据，表示x来自真实数据而非生成数据的概率；表示为生成器G的输入噪声变量，表示生成器G生成的数据。判别器D期望真实数据的判别概率更高，生成数据的判别概率更低，因此训练判别器D以最大化目标函数。生成器G期望生成的数据能够骗过判别器D，即生成数据的判别概率更高，因此训练生成器G以最小化目标函数，即最小化。通过对抗训练的方式让生成器G能够生成高质量的数据。

循环一致性生成对抗网络（CycleGAN，Cycle-Consistent Generative Adversarial Networks）[21]是对GAN的优化，在训练过程中不需要提供成对的图像。假设图像源域、目标域，目标是训练生成器实现映射。CycleGAN中提出逆映射，并引入循环一致性与，即源域图像通过生成器生成目标域图像，目标域图像再经过逆映射生成的源域图像应与相似（反之亦然）。此外，还引入两个对立的判别器和，负责区分源域图像和转换图像，负责区分目标域图像和转换图像。因此训练目标包含两层：对抗性损失与循环一致性损失。

对抗性损失的目标函数如式2所示：

（式2）

其中，生成器G试图生成与域相似的图像，而试图正确区分生成图像和真实图像，因此有二元极小极大问题。类似的，为映射函数和判别器引入二元极小极大问题。

循环一致性损失的目标函数如式3所示，其中目标函数为最小值问题：

（式3）

因此CycleGAN的目标函数如式4所示，极小极大值问题如式5所示：

（式4）

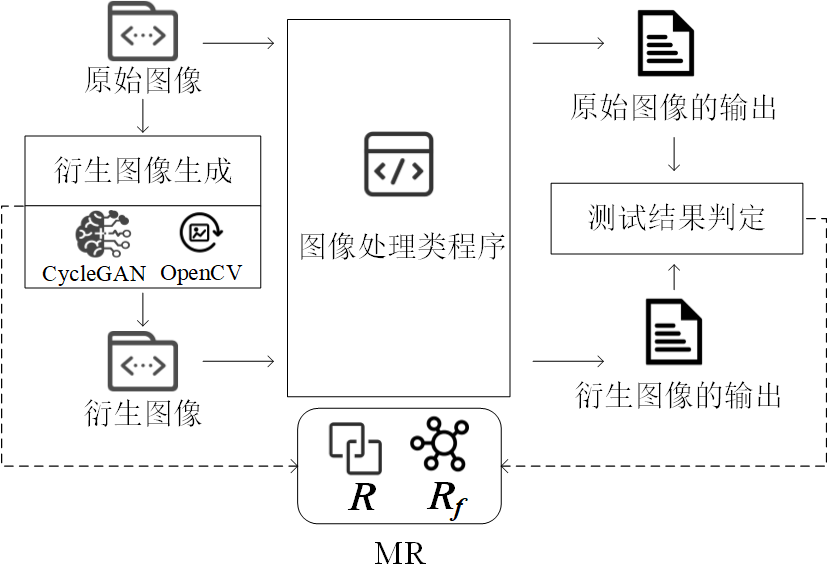
（式5）

本文采用CycleGAN作为图像风格转换方式，支持蜕变测试过程中衍生测试用例的生成，实现图像晴天到冬天的风格转换。

* 1. 小结

主要介绍了相关技术，包括软件测试的概念、传统软件测试技术的过程与缺点、蜕变测试。最后介绍了GAN以及CycleGAN的基本原理。

1. 研究总体思路



**图4 整体技术架构**

图4展示了引入CycleGAN的面向图像处理类程序的蜕变测试技术整体架构，主要过程如下：

1. 衍生图像生成：对于一个MR，依据其关系*R*对每一个原始测试图像进行特定的变换，得到衍生测试图像集。具体来说，通过OpenCV实现放缩、转置、侵蚀等简单的图像变换；通过训练好的CycleGAN模型实现风格迁移（如实现图像晴天到冬天的风格迁移）等复杂的图像变换。
2. 测试执行和结果验证：对于一个MR，分别使用原始测试图像集和衍生测试图像集执行图像处理类程序，并获得相应的输出结果；依据MR中的关系*Rf*验证输出结果，判断图像处理类程序是否存在缺陷。迭代步骤（1）和（2）直至遍历完所有MR。

本项目主要关注面向图像处理应用程序的蜕变测试，分别对图像分类器、膨胀算法和图像目标检测算法三类程序总结并选取蜕变关系。

对于图像分类器选择出20种蜕变关系，如表2所示。蜕变关系中包括了3个复合蜕变关系，即MR-15.1是MR-11.1和MR-14.1的复合蜕变关系、MR-16.1是MR-7.1和MR-14.1的复合蜕变关系、MR-17.1是MR-11.1和MR-13.1的复合蜕变关系。

**表2 图像分类器蜕变关系**

|  |  |
| --- | --- |
| **编号** | **蜕变关系** |
| MR-1.1 | 图像的BGR通道顺序改变，分类结果相似。 |
| MR-2.1 | 图像顺时针旋转90°/180°/270°，分类结果相似。 |
| MR-3.1 | 图像转置，分类结果相似。 |
| MR-4.1 | 图像顺时针旋转90°/180°/270°+转置，分类结果相似。 |
| MR-5.1 | 改变图像对比度，分类结果相似。 |
| MR-6.1 | 改变图像亮度，分类结果相似。 |
| MR-7.1 | 对图像作模糊处理，分类结果相似。 |
| MR-8.1 | 对图像作锐化处理，分类结果相似。 |
| MR-9.1 | 对图像作噪声扰动处理，分类结果相似。 |
| MR-10.1 | 对图像作仿射变换，用黑色填充，分类结果相似。 |
| MR-11.1 | 图像在x轴正方向作微小平移，分类结果相似。 |
| MR-12.1 | 图像微小放大，分类结果相似。 |
| MR-13.1 | 图像微小缩小，分类结果相似。 |
| MR-14.1 | 图像微小旋转，分类结果相似。 |
| MR-15.1 | 图像微小旋转+平移，分类结果相似。 |
| MR-16.1 | 图像微小旋转+模糊，分类结果相似。 |
| MR-17.1 | 图像微小平移+缩小，分类结果相似。 |
| MR-18.1 | 图像微小侵蚀，分类结果相似。 |
| MR-19.1 | 图像微小膨胀，分类结果相似。 |
| MR-21.1 | 图像风格由晴天转为雪天，分类结果相似。 |

对于二值图像膨胀算法选择出4种蜕变关系，如表3所示。其中MR-20是二值图像膨胀算法的特殊蜕变关系。

对于图像目标检测算法选择出20种蜕变关系，如表4所示。图像目标检测算法的20种蜕变关系均为通用蜕变关系。例如MR-1.1与MR-1.2的图像变换方式相同，但应用的图像处理软件不同，所以结果对比部分存在差异，即输入满足关系*r*，但输出并不满足同一关系*rf*。

蜕变关系还设定了具体的参数，因此更改蜕变关系后，可调节参数同时发生改变，具体调节参数如表5所示。

**表3 二值图像膨胀算法蜕变关系**

|  |  |
| --- | --- |
| **编号** | **蜕变关系** |
| MR-2.3 | 二值图像旋转90°/180°/270°后膨胀结果与膨胀后旋转90°/180°/270°结果相似。 |
| MR-3.3 | 二值图像转置后膨胀结果与膨胀后转置结果相似。 |
| MR-4.3 | 二值图像旋转90°/180°/270°+转置后膨胀结果与转置后旋转90°/180°/270°结果相似。 |
| MR-20 | 二值图像补码的侵蚀结果与膨胀后的补码结果相似，二值图像补码的膨胀结果与侵蚀后的补码结果相似。 |

**表4 图像目标检测算法蜕变关系**

|  |  |
| --- | --- |
| **编号** | **蜕变关系** |
| MR-1.2 | 图像的BGR通道顺序改变，目标检测结果相似。 |
| MR-2.2 | 图像顺时针旋转90°/180°/270°，目标检测结果相似。 |
| MR-3.2 | 图像转置，目标检测结果相似。 |
| MR-4.2 | 图像顺时针旋转90°/180°/270°+转置，目标检测结果相似。 |
| MR-5.2 | 改变图像对比度，目标检测结果相似。 |
| MR-6.2 | 改变图像亮度，目标检测结果相似。 |
| MR-7.2 | 对图像作模糊处理，目标检测结果相似。 |
| MR-8.2 | 对图像作锐化处理，目标检测结果相似。 |
| MR-9.2 | 对图像作噪声扰动处理，目标检测结果相似。 |
| MR-10.2 | 对图像作仿射变换，用黑色填充，目标检测结果相似。 |
| MR-11.2 | 图像在x轴正方向作微小平移，目标检测结果相似。 |
| MR-12.2 | 图像微小放大，目标检测结果相似。 |
| MR-13.2 | 图像微小缩小，目标检测结果相似。 |
| MR-14.2 | 图像微小旋转，目标检测结果相似。 |
| MR-15.2 | 图像微小旋转+平移，目标检测结果相似。 |
| MR-16.2 | 图像微小旋转+模糊，目标检测结果相似。 |
| MR-17.2 | 图像微小平移+缩小，目标检测结果相似。 |
| MR-18.2 | 图像微小侵蚀，目标检测结果相似。 |
| MR-19.2 | 图像微小膨胀，目标检测结果相似。 |
| MR-21.2 | 图像风格由晴天转为雪天，目标检测结果相似。 |

**表5 蜕变关系与可调节参数**

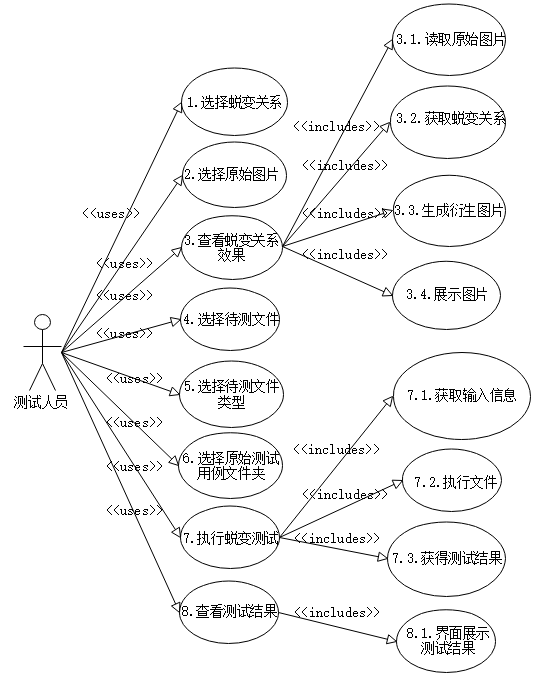
|  |  |
| --- | --- |
| **蜕变关系图像变换方式** | **可调节参数** |
| MR-1：BGR通道顺序改变（初始BGR） | BRG、RGB、RBG、GBR、GRB |
| MR-2：顺时针旋转90°/180°/270° | 90°、180°、270° |
| MR-3：转置 | 无 |
| MR-4：顺时针旋转90°/180°/270°+转置 | 90°、180°、270° |
| MR-5：改变对比度 | 0.75、0.8、0.85、0.9、0.95 |
| MR-6：改变亮度 | 5、10、15、20、25 |
| MR-7：模糊处理 | 无 |
| MR-8：锐化处理 | 无 |
| MR-9：噪声扰动处理 | 无 |
| MR-10：仿射变换 | 无 |
| MR-11：x轴正方向微小平移 | 0.01、0.02、0.03、0.04、0.05 |
| MR-12：微小放大 | 0.01、0.02、0.03、0.04、0.05 |
| MR-13：微小缩小 | 0.01、0.02、0.03、0.04、0.05 |
| MR-14：微小旋转 | 1.6°、1.8°、2.0°、2.2°、2.4° |
| MR-15：微小旋转+平移 | 无 |
| MR-16：微小旋转+模糊 | 1.6°、1.8°、2.0°、2.2°、2.4° |
| MR-17：微小平移+缩小 | 无 |
| MR-18：微小侵蚀 | 无 |
| MR-19：微小膨胀 | 无 |
| MR-20：二值图像补码 | 无 |
| MR-21：图像风格晴天转雪天 | 无 |

1. 图像处理软件的蜕变测试支持工具设计与实现

讨论图像处理软件的蜕变测试支持工具设计与实现，包括需求分析、总体设计、详细设计以及工具实现与演示。

* 1. 需求分析

图像处理软件的蜕变测试支持工具——MT4I主要由测试人员使用，MT4I的用例图如图5所示。

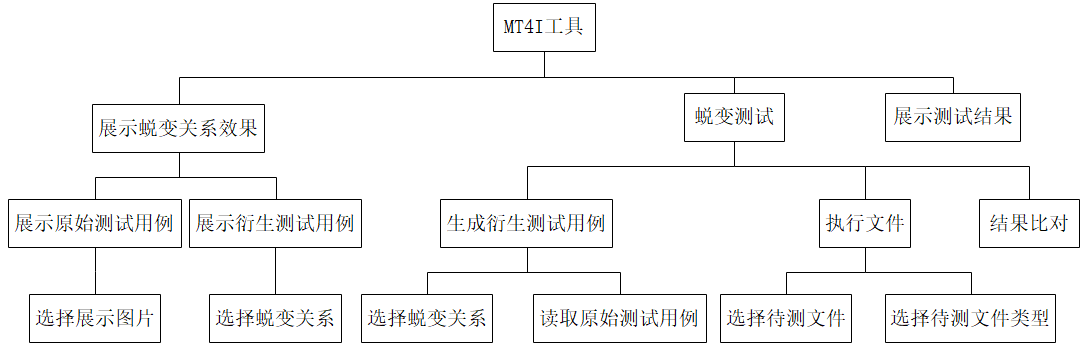


**图5 MT4I工具用例图**

下面，我们详细讨论图5中每个用例。

1. **选择蜕变关系：**测试人员可以指定蜕变关系，用于展示效果以及执行蜕变测试方法；
2. **选择原始图片：**指定原始图片，用于展示蜕变关系效果；
3. **查看蜕变关系效果：**根据以上指定的蜕变关系与原始图片生成衍生图片，在工具界面上展示，与原始图片相对比直观地呈现蜕变关系效果；
4. **选择待测文件：**指定待测文件，为执行蜕变测试做准备；
5. **选择待测文件类型：**指定待测文件类型，用于选择适合的蜕变测试框架；
6. **选择原始测试用例文件夹：**指定原始测试用例路径，为执行蜕变测试做准备；
7. **执行蜕变测试：**获取以上输入信息，使用蜕变测试方法执行文件，获得测试结果；
8. **查看测试结果：**在工具的界面上查看文件的测试结果。
   1. 总体设计

基于上述需求分析的结果，进一步讨论图像处理软件的蜕变测试支持工具MT4I的总体设计。MT4I包括展示蜕变关系效果、蜕变测试以及展示测试结果三个模块，具体如图6所示。



**图6 MT4I工具层次图**

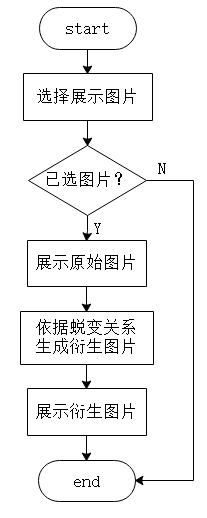
下面对图6中的3大模块进行具体描述：

1. **展示蜕变关系效果：**在界面上展示测试人员选择的图片；以原始图片为基础，依据测试人员选择的蜕变关系生成衍生图片；在界面上展示衍生图片，与原始图片相对比可以在视觉上直观地感受到蜕变关系效果。
2. **蜕变测试：**蜕变测试部分是MT4I工具的核心模块，首先测试人员指明原始测试用例集以及选择蜕变关系，工具以此为基础生成衍生测试用例集；接下来根据测试人员选择的待测文件和待测文件类型，工具以不同的待测文件类型选择不同的蜕变测试框架，使用上一步骤中的原始测试用例和衍生测试用例分别作为输入执行待测文件；获得两次执行结果，并判断结果是否满足事先选择的蜕变关系，得出测试结论。
3. **展示测试结果：**将测试结果展示在界面上，包括参与测试的图片总数量、未通过测试的图片数量以及未通过测试比率。
   1. 详细设计

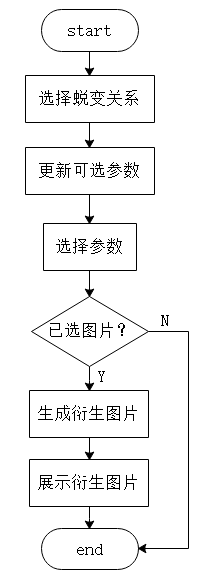
讨论MT4I工具的详细设计，依据工具的功能要求，分别从前端设计和后端设计两部分进行具体描述。

* + 1. 前端设计

测试人员通过界面对工具进行操作，包括查看蜕变关系效果以及查看蜕变测试结果。按照界面的功能要求，前端设计主要分为实现展示蜕变关系效果、展示执行蜕变测试的测试结果两部分。

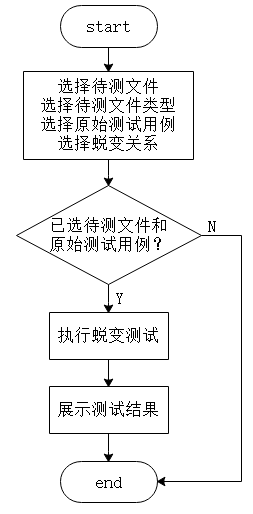


**图7 展示原始图片与衍生图片流程图**



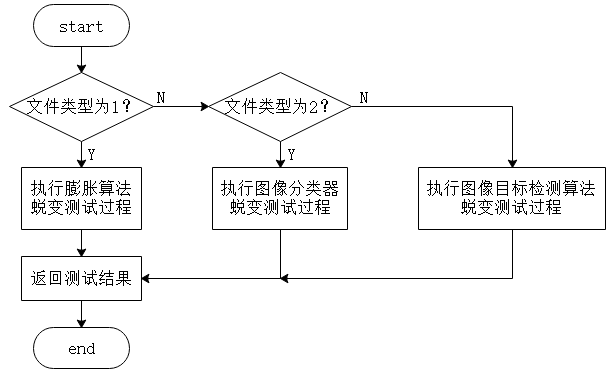
**图8 选取蜕变关系流程图**

1. **展示蜕变关系效果：**图7所示为界面展示原始测试用例图片以及衍生测试用例图片的程序流程图。工具初始化时会默认一种蜕变关系，所以只要测试人员选择了原始图片，工具就可以直接生成衍生图片，并且在界面上展示原始图片和衍生图片。图8所示为显示蜕变关系效果部分，更改蜕变关系后的程序流程图。由于每种蜕变关系设置了不同的可调节参数，所以更改蜕变关系后，需要同步更新参数选择栏。在选择蜕变关系以及具体参数之后，程序判断是否已选择展示图片，如果选择了则以展示图片为基础，依据更改的蜕变关系与参数生成新的衍生图片，更新界面上展示的衍生图片。
2. **展示蜕变测试结果：**图9所示为界面上为待测文件执行蜕变测试的程序流程图。在开始蜕变测试之前，需要测试人员在工具中选择待测文件、待测文件类型、原始测试用例以及蜕变关系。这里的蜕变关系仍带有可调节参数，与上述内容类似，这里不再赘述。待测文件类型用于后端选择适合的蜕变测试框架，本文涉及到膨胀算法、图像分类器以及图像目标检测算法三种类型。在工具初始化时文件类型会默认为膨胀算法，蜕变关系也会是默认值，因此只要测试人员选择了待测文件以及原始测试用例集，蜕变测试即可开始，在蜕变测试结束后，将测试结果展示在界面上。



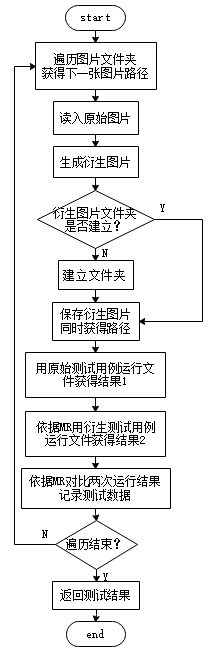
**图9 操作界面执行蜕变测试流程图**

* + 1. 后端设计



**图10 按文件类型执行蜕变测试流程图**

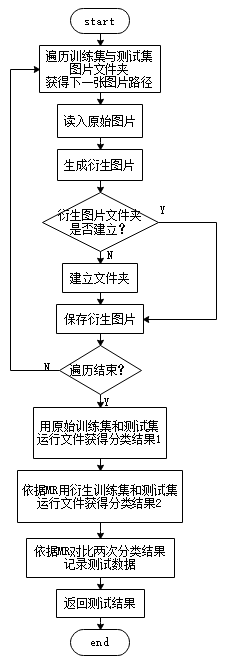
本项目涉及到膨胀算法、图像分类器以及图像目标检测算法三种图像处理软件类型：膨胀算法以单张图片作为输入，结果返回膨胀后的图像；图像分类器以训练集和测试集图像作为输入，返回测试集的分类结果；图像目标检测算法以测试集和目标检测结果存储路径作为输入，返回测试集的目标检测结果。因为三种类型程序的输入与输出存在差异，所以在工具中用图10所示判断方法加以区分，不同类型的程序执行不同的蜕变测试框架。膨胀算法、图像分类器和图像目标检测算法的蜕变测试程序流程图如图11、图12、图13所示。



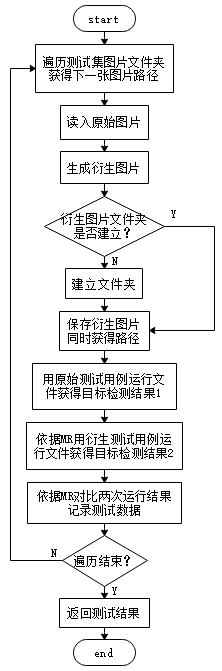
**图11 膨胀算法蜕变测试程序流程图**

图11所示为膨胀算法进行蜕变测试的程序流程图。膨胀算法的特点是处理单张图片，因此在蜕变测试过程中每读入一张图片，依据蜕变关系生成对应的衍生测试用例。主要步骤如下：

1. 遍历界面中选择的原始测试用例集，获取下一张原始测试用例；
2. 依据蜕变关系生成衍生测试用例；
3. 以原始测试用例作为输入，执行待测文件获得运行结果；
4. 以衍生测试用例作为输入，执行待测文件获得运行结果；
5. 依据蜕变关系对比两次运行结果，记录测试数据；
6. 返回第一步继续读入原始测试用例进行蜕变测试，直至原始测试用例均执行完毕；
7. 最后统计测试结果，返回给界面。



**图12 图像分类器蜕变测试程序流程图**



**图13 图像目标检测算法蜕变测试程序流程图**

图12所示为图像分类器的蜕变测试程序流程图。因为图像分类器以整个训练集和测试集作为输入，在执行文件之前需要生成训练集和测试集的所有衍生测试用例。主要步骤如下：

1. 遍历训练集与测试集，获取下一个原始测试用例；
2. 依据蜕变关系生成衍生测试用例；
3. 将衍生测试用例保存到对应路径内，以便执行时调用；
4. 返回第一步，继续获取原始测试用例，直至训练集与测试集所有原始测试用例均已生成衍生测试用例；
5. 以原始训练集和测试集作为输入，执行待测文件，获得分类结果；
6. 依据蜕变关系选择对应的衍生训练集和衍生测试集作为输入，执行待测文件，获得分类结果；
7. 对比两次分类结果是否满足蜕变关系，记录测试数据；
8. 将测试结果返回给界面。

图13所示为图像目标检测算法的蜕变测试程序流程图。因为图像目标检测算法常以整个测试集和目标检测结果图存储路径作为输入，所以在蜕变测试过程中需要生成测试集的所有衍生测试用例。主要步骤如下：

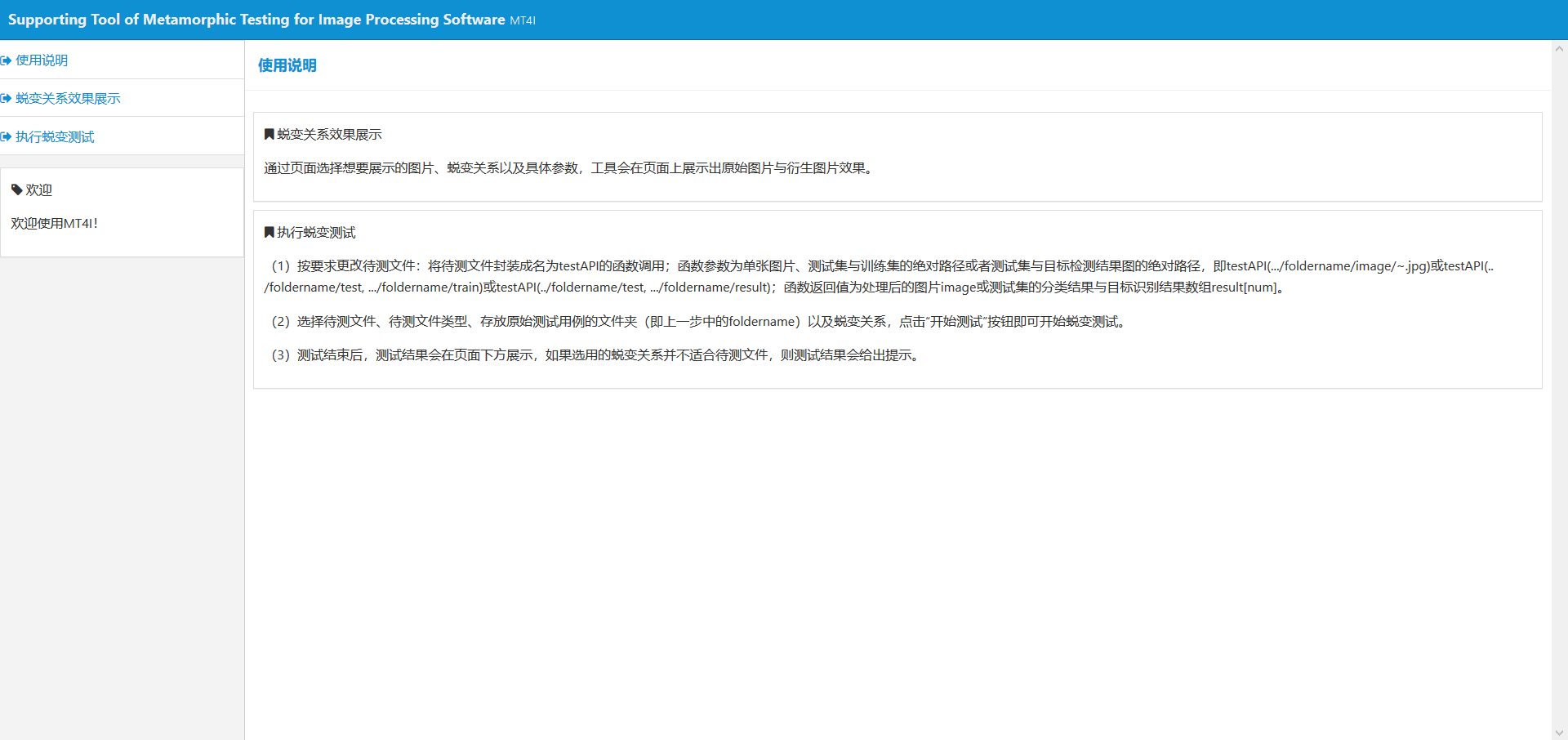
1. 遍历测试集，获取下一个原始测试用例；
2. 依据蜕变关系生成衍生测试用例；
3. 以原始测试用例作为输入，执行待测文件获得目标检测结果，具体为图像中各物体的类别检测结果；
4. 以衍生测试用例作为输入，执行待测文件获得目标检测结果；
5. 依据蜕变关系对比两次运行结果，记录测试数据；
6. 重复上述步骤，继续读入原始测试用例进行蜕变测试，直至测试集中所有原始测试用例均执行完毕；
7. 最后统计测试结果，返回给界面。
   1. 工具实现与演示

本项目采用Python和Java编程语言实现系统，使用JSP完成系统的可视化页面。下面分别从工具使用说明页面、工具蜕变关系效果展示以及执行蜕变测试这三方面进行MT4I工具演示。

**（1）工具使用说明页面**

在工具初始化时，显示使用说明页面，提醒测试人员工具的具体功能、操作流程以及操作注意事项，如图14所示。使用说明具体内容如下：

1. 通过页面选择想要展示的图片、蜕变关系以及具体参数，工具会在页面上展示出原始图片与衍生图片效果。
2. 将待测文件封装成名为testAPI的函数调用；函数参数为单张图片或者测试集与训练集的绝对路径；函数返回值为处理后的图片image或测试集的分类结果与目标识别结果数组result[num]。
3. 选择待测文件、待测文件类型、存放原始测试用例的文件夹以及蜕变关系，点击“开始测试”按钮即可开始蜕变测试。
4. 测试结束后，测试结果会在页面下方展示，如果选用的蜕变关系并不适合待测文件，则测试结果会给出提示。



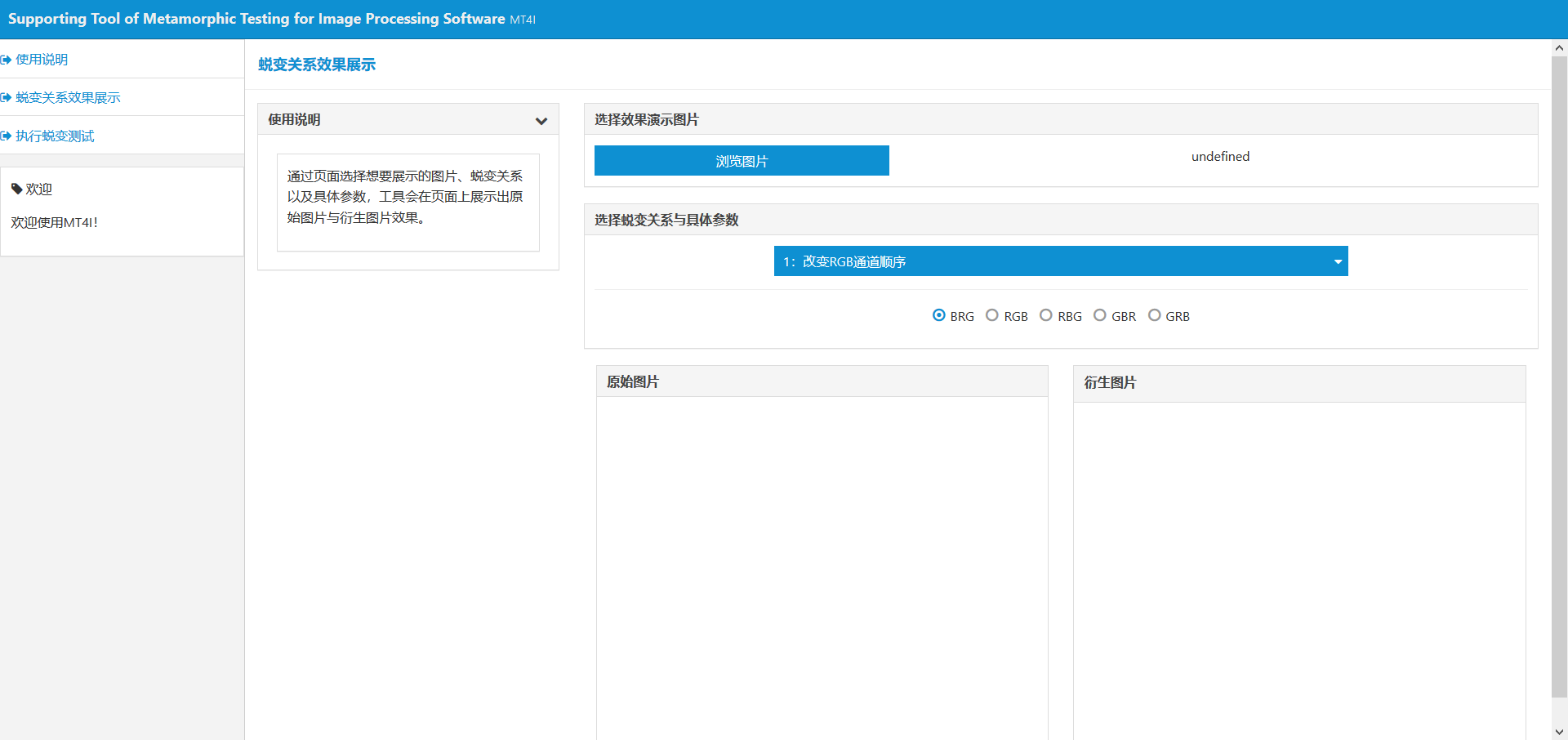
**图14 工具使用说明页面**

**（2）工具蜕变关系效果展示页面**

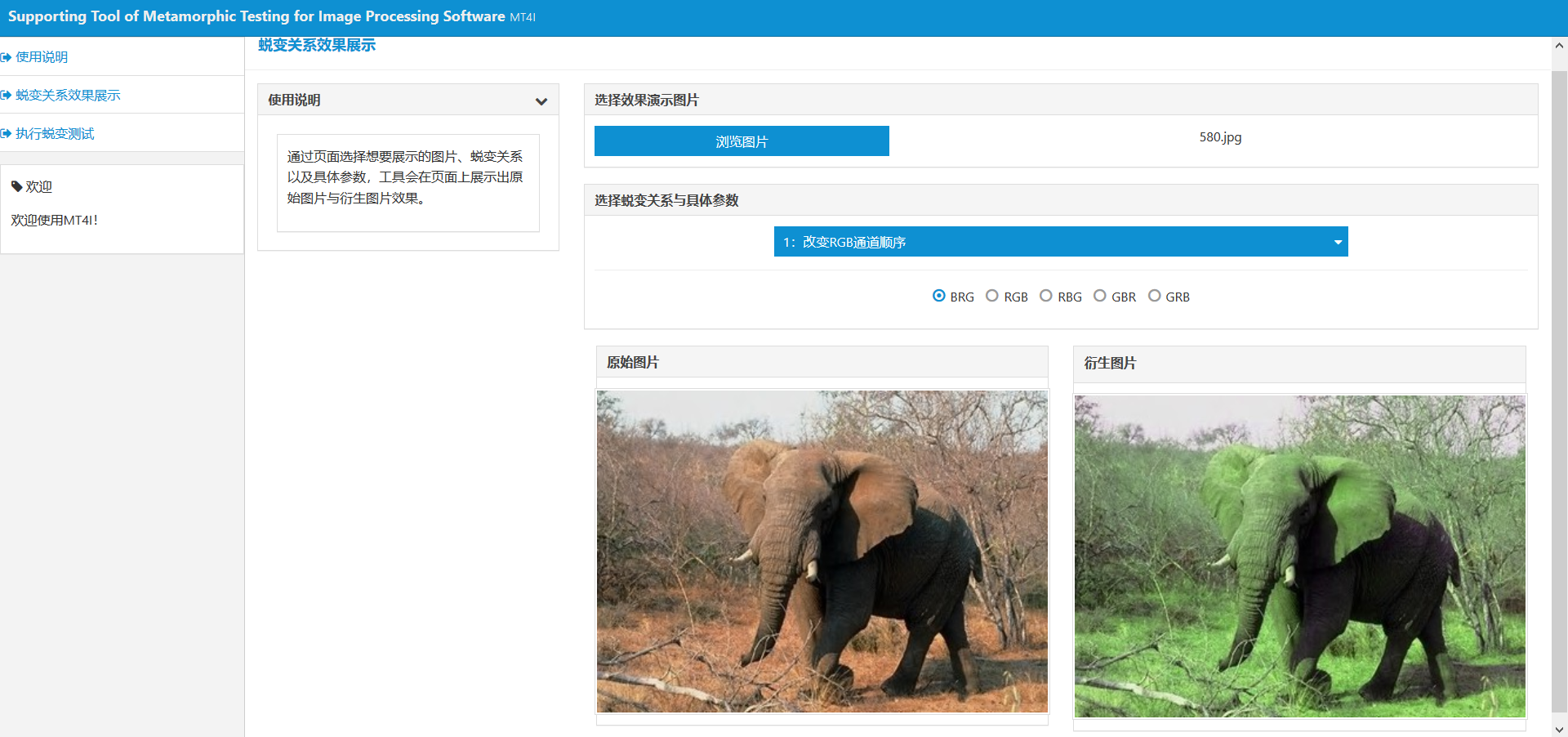
图15所示为工具蜕变关系效果展示页面的初始化状态，此时蜕变关系默认为MR-1、参数默认为第1种，因为是初始化状态并没有选择展示的图片，所以原始图片和衍生图片位置都是空白。测试人员可以选择想要展示的图片以及蜕变关系。

图16所示为选择展示图片后的效果，工具左侧展示出原始图片，右侧展示出依据蜕变关系生成的衍生图片。

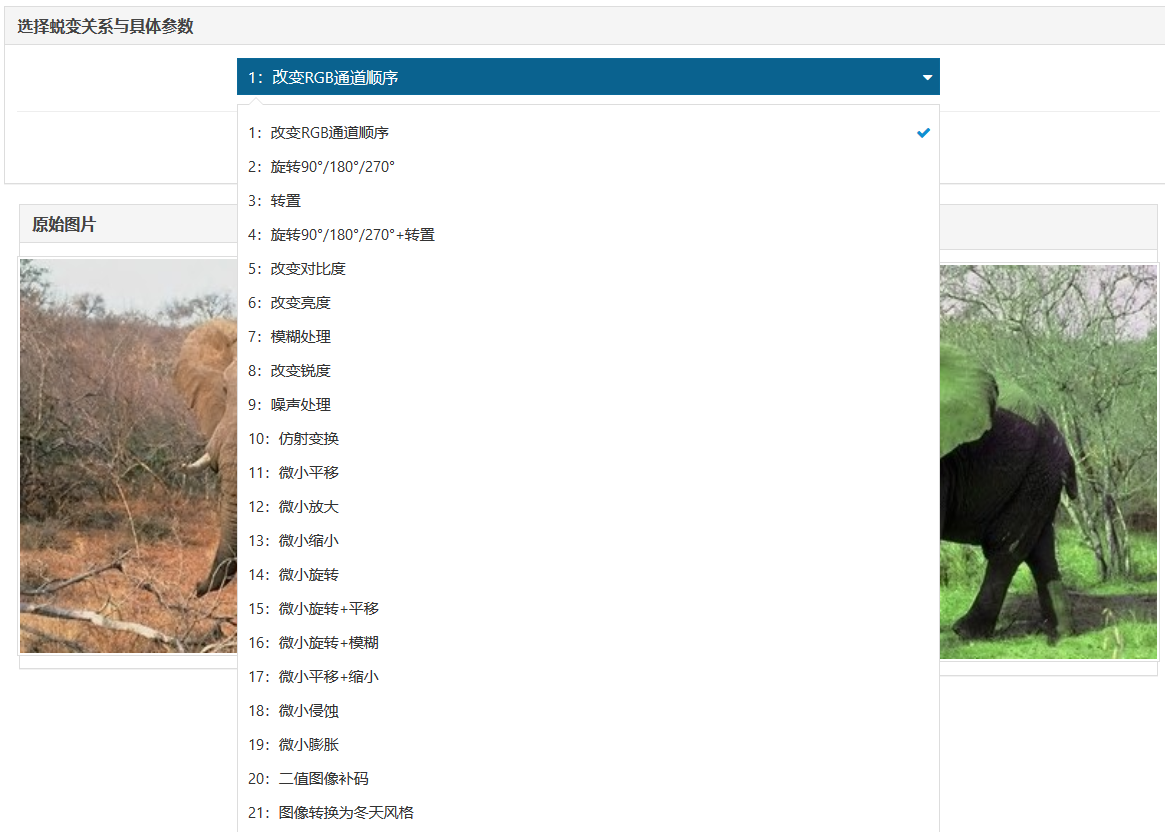
工具目前共有21种蜕变关系可供选择，如图17所示。图18所示为蜕变关系更改成MR-2（图像顺时针旋转90°/180°/270°）的效果，此时衍生图片更新，原始图片不变。



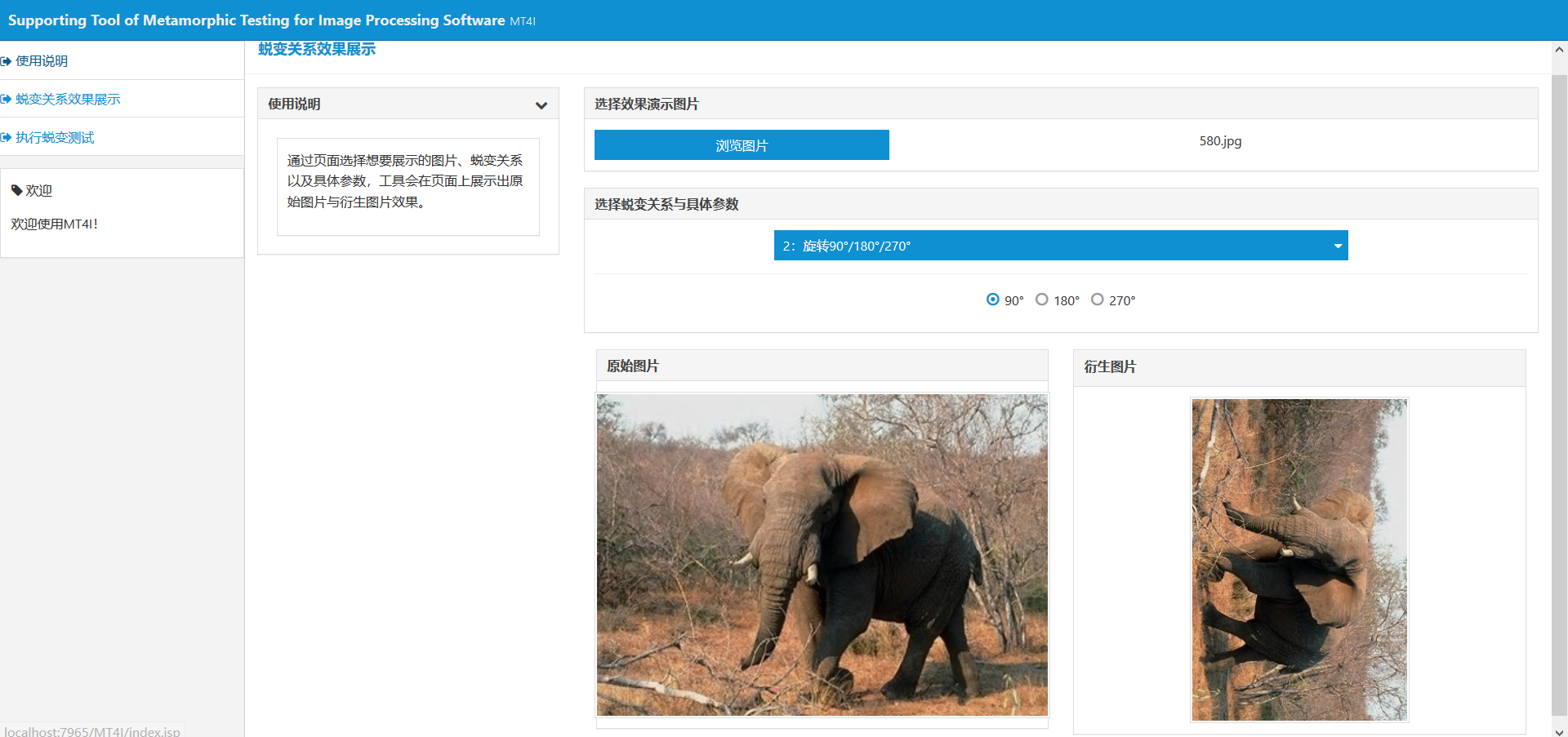
**图15 蜕变关系效果展示初始化页面**



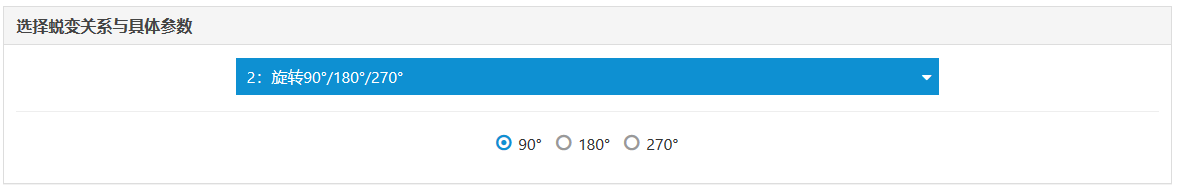
**图16 选择展示图片后的效果图**



**图17 蜕变关系选取图**



**图18 更改蜕变关系后的效果图**



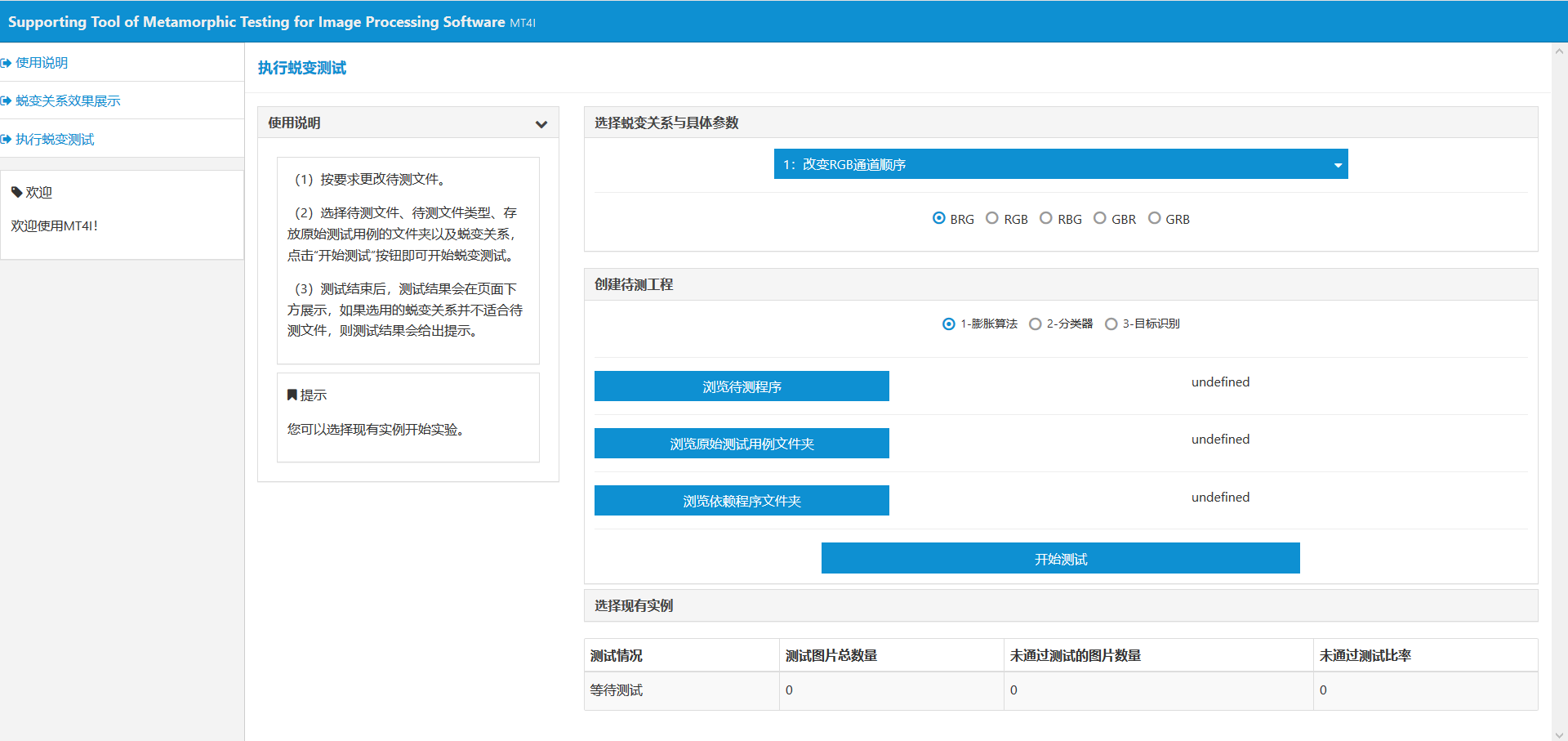
**图19 可调节参数选取图**

蜕变关系的可调节参数选择如图19所示。

**（3）执行蜕变测试**

图20所示为工具执行蜕变测试的初始化页面。其中待测文件类型初始化时默认为膨胀算法，蜕变关系默认为MR-1。测试人员需要从页面中选择待测文件类型、待测文件、原始测试用例文件夹、蜕变关系，然后点击“开始测试”按钮开始蜕变测试，测试结果会在页面下方展示出来。

最后以KNN图像分类器作为例子，Corel-1000作为原始测试用例，选择蜕变关系MR-3.1执行蜕变测试，得到如图21所示结果。此次测试情况为“测试成功”，测试集总共400张图片，有16张未通过测试，即这16张图片的分类结果不等，最后计算出的未通过测试比率为0.04。



**图20 执行蜕变测试初始化界面**



**图21 蜕变测试结果图**

* 1. 小结

详细阐述了图像处理软件的蜕变测试支持工具MT4I的设计与实现，对工具的关键问题给出解决方案，最后以KNN图像分类器为例子，演示了工具界面及主要功能。

参考文献

1. Chen T Y, Kuo F C, Liu H, et al. Metamorphic Testing: A Review of Challenges and Opportunities [J]. ACM Computing Surveys, 2018, 51(1): 4:1-4:27.
2. 董国伟，徐宝文，陈林，等．蜕变测试技术综述[J]．计算机科学与探索，2009， 3（2）：130—143．
3. Chen T Y, Tse T H, Zhou Z Q. Fault-based testing without the need of oracles [J]. Information and Software Technology, 2003, 45(1): 1-9.
4. Brilliant S S, Knight J C, Ammann P E. On the Performance of Software Testing Using Multiple Versions [C]. //Proceedings of the 20th International Symposium on Fault-Tolerant Computing (FTCS). IEEE, 1990: 408-415.
5. Sim K Y, Low C S, Kuo F C. Eliminating Human Visual Judgment from Testing of Financial Charting Software [J]. Journal of Software, 2014, 9(2): 298-312.
6. Jameel T, Lin M, Liu C. Test Oracles Based on Metamorphic Relations for Image Processing Applications [C]. //Proceedings of the 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). IEEE, 2015: 1-6.
7. Mayer J, Guderlei R. On Random Testing of Image Processing Applications [C]. //Proceedings of the 6th International Conference on Quality Software. IEEE, 2006: 85-92.
8. Jameel T, Lin M, Liu C. Metamorphic Relations Based Test Oracles for Image Processing Applications[J]. International Journal of Software Innovation (IJSI), 2016, 4(1): 16-30.
9. Jameel T, Lin M, Liu C. A Framework of Automatic Testing of Image Processing Applications [C]. //Proceedings of the 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST). IEEE, 2016: 312-317.
10. 黄凯奇，任伟强，谭铁牛．图像物体分类与检测算法综述[J]．计算机学报，2014，37（6）：1225—1240．
11. Xie X, Ho J, Murphy C, et al. Application of Metamorphic Testing to Supervised Classifiers [C]. //Proceedings of the 9th International Conference on Quality Software. IEEE, 2009: 135-144.
12. Dwarakanath A, Ahuja M, Sikand S, et al. Identifying Implementation Bugs in Machine Learning Based Image Classifiers using Metamorphic Testing [C]. //Proceedings of the 27th ACM SIGSOFT International Symposium on Software Testing and Analysis. ACM, 2018: 118-128.
13. Braiek H B, Khomh F. DeepEvolution: A Search-Based Testing Approach for Deep Neural Networks [C]. //Proceedings of the International Conference on Software Maintenance and Evolution (ICSME). IEEE, 2019: 454-458.
14. 王赞，闫明，刘爽，等．深度神经网络测试研究综述[J]．软件学报，2020，31（5）：1255—1275．
15. Segura S, Troya J, Durán A, et al. Performance Metamorphic Testing: Motivation and Challenges [C]. //Proceedings of the 39th International Conference on Software Engineering: New Ideas and Emerging Technologies Results Track (ICSE-NIER). IEEE, 2017: 7-10.
16. Jiang M, Chen T Y, Kuo F C, et al. Testing Central Processing Unit Scheduling Algorithms Using Metamorphic Testing [C]. //Proceedings of the 4th International Conference on Software Engineering and Service Science (ICSESS). IEEE, 2013: 530-536.
17. Chen T Y, Kuo F C, Ma W, et al. Metamorphic Testing for Cybersecurity [J]. Computer, 2016, 49(6): 48-55.
18. Segura S, Fraser G, Sanchez A B, et al. A Survey on Metamorphic Testing [J]. IEEE Transactions on Software Engineering, 2016, 42(9): 805-824.
19. 董国伟，聂长海，徐宝文．基于程序路径分析的有效蜕变测试[J]．计算机学报，2009，32（5）：1002—1013．
20. Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative Adversarial Nets [C]. //Proceedings of the 27th International Conference on Neural Information Processing Systems. 2014: 2672-2680.
21. Zhu J, Park T, Isola P, et al. Unpaired Image-to Image Translation using Cycle-Consistent Adversarial Networks [C]. //Proceedings of the International Conference on Computer Vision (ICCV). IEEE, 2017: 2242-2251.