安装gaps

$ git clone https://github.com/nemanja-m/gaps.git

$ cd gaps

$ pip **install** -r requirements.txt

$ sudo apt-**get** **install** python-tk

$ pip install -e .

# Creating puzzles from images

To create puzzle from image use create\_puzzle script.

$ create\_puzzle images/pillars.jpg --size=48 --destination=puzzle.jpg

[SUCCESS] Puzzle created **with** 420 pieces

will create puzzle with 420 pieces from images/pillars.jpg where each piece is 48x48 pixels.

Run create\_puzzle --help for detailed help.

**NOTE** Created puzzle dimensions may be smaller then original image depending on given puzzle piece size. Maximum possible rectangle is cropped from original image.

# Solving puzzles

In order to solve puzzles, use gaps script.

$ gaps --image=puzzle.jpg --generations=20 --population=600

This will start genetic algorithm with initial population of 600 and 20 generations.

Following options are provided:

| **Option** | **Description** |
| --- | --- |
| --image | Path to puzzle |
| --size | Puzzle piece size in pixels |
| --generations | Number of generations for genetic algorithm |
| --population | Number of individuals in population |
| --verbose | Show best solution after each generation |
| --save | Save puzzle solution as image |

Run gaps --help for detailed help.

## Size detection

If you don't explicitly provide --size argument to gaps, piece size will be detected automatically.

However, you can always provide gaps with --size argument explicitly:

$ gaps --image=puzzle.jpg --generations=20 --population=600 --size=48

**NOTE** Size detection feature works for the most images but there are some edge cases where size detection fails and detects incorrect piece size. In that case you can explicitly set piece size.

## Termination condition

The termination condition of a Genetic Algorithm is important in determining when a GA run will end. It has been observed that initially, the GA progresses very fast with better solutions coming in every few iterations, but this tends to saturate in the later stages where the improvements are very small.

gaps will terminate:

* when there has been no improvement in the population for X iterations, or
* when it reachs an absolute number of generations

# License

This project as available as open source under the terms of the [MIT License](http://opensource.org/licenses/MIT)

# 安装

克隆回购：

$ git clone https： / /github.com/nemanja -m / gaps.git

$ cd差距

安装要求：

$ pip **install** -r requirements.txt

$ sudo apt- **获取** **安装** python-tk

以可编辑模式安装项目：

$ pip install -e 。

# 从图像创建拼图

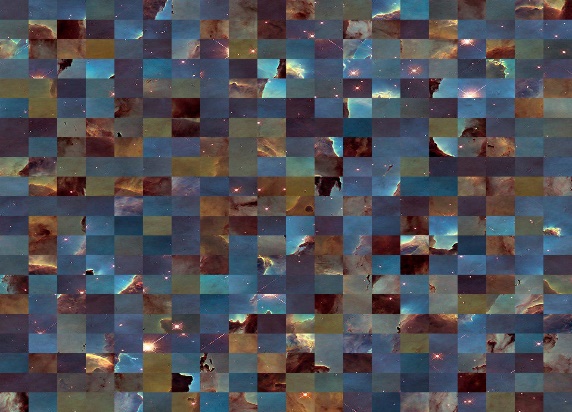
要使用图像创建拼图，请使用create\_puzzle脚本。

即

/ pillars.jpg --size = $ create\_puzzle图像48 --destination = puzzle.jpg

[SUCCESS]拼图创建**与** 420件

将从每个片段为48x48像素的420片段中创建拼图images/pillars.jpg。

[](https://github.com/nemanja-m/gaps/raw/master/images/pillars.jpg)             [](https://github.com/nemanja-m/gaps/raw/master/images/demo_puzzle.jpg)

运行create\_puzzle --help以获取详细帮助。

**注意** 根据给定的拼图尺寸，创建的拼图尺寸可能会小于原始图像。从原始图像中裁剪出最大可能的矩形。

# 解决难题

为了解决难题，请使用gaps脚本。

即

$ gaps --image = puzzle.jpg --generations = 20 --population = 600

这将开始遗传算法，初始种群为600和20代。

提供以下选项：

| **选项** | **描述** |
| --- | --- |
| --image | 困惑之路 |
| --size | 拼图块大小（以像素为单位） |
| --generations | 遗传算法的代数 |
| --population | 人口总数 |
| --verbose | 在每一代之后显示最佳解决方案 |
| --save | 将拼图解决方案另存为图像 |

运行gaps --help以获取详细帮助。

## 尺寸检测

如果您未明确提供--size参数gaps，则会自动检测件数。

但是，可以随时提供gaps与--size参数明确：

$间隙--image = puzzle.jpg --generations = 20 --population = 600 --size = 48

**注意** 尺寸检测功能适用于大多数图像，但是在某些极端情况下，尺寸检测会失败并检测到不正确的尺寸。在这种情况下，您可以明确设置件数。

## 终止条件

遗传算法的终止条件对于确定GA运行何时结束很重要。已经观察到，最初，GA进步很快，每隔几次迭代就会有更好的解决方案，但是在以后的改进很小的阶段，趋于饱和。

gaps 将终止：

* 如果X迭代的总体没有改善，或者
* 当它达到绝对世代数时