

## 使用OCR

eggPlant Functional设计用于处理图像，因此最好尽可能使用图像作为首选搜索工具。比使用OCR更快更可靠。但是，不能使用图像时，OCR是一个非常有用的工具。搜索动态文本时，容易出现图像不能使用的问题。

为了提高特定情境下OCR的可靠性，可在搜索中添加以下参数：

### SearchRectangle

在任何时候，添加搜索矩形框都是有用的。OCR搜索整个屏幕时，不仅速度更慢，精确度也更低，因为可能出现额外的匹配，或者根本无匹配。

使用精确的坐标可指定搜索框，但是eggPlant Functional通常使用图像指定搜索框，以防文本位置不一致。例如，在测试过程中包含文本的窗口每次显示在屏幕上的位置不同时，可能出现上述问题。

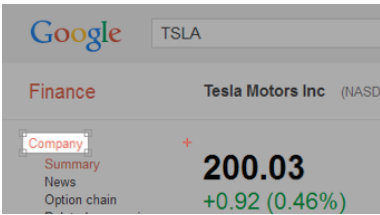
使用图像确定搜索框限时，记住用于设置搜索框的点基于热点，可移动到图像外。

示例：

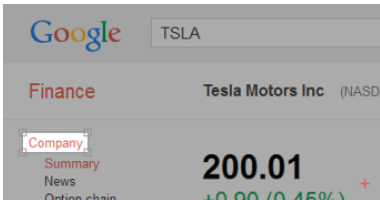
测试可能使用谷歌财经网页，搜索指定的公司，然后读取股票价格。为确保OCR正确读取股票价格，可使用图像确定搜索框。所使用的代码可以很简单，如下所示：

```
Log ReadText ("TLImage", "BRImage")
—代码也可以很复杂，取决于所需的输出结果。
```

需要为以上代码捕获两张图像，确定搜索框的左上角和右下角。本例中的两张图像是"TLImage"和"BRImage"。为捕获图像，需要选择屏幕上的一个或多个元素，它们与OCR将要读取文本的关系相对固定位置。本例使用一个元素，即"Company"（公司）标签。

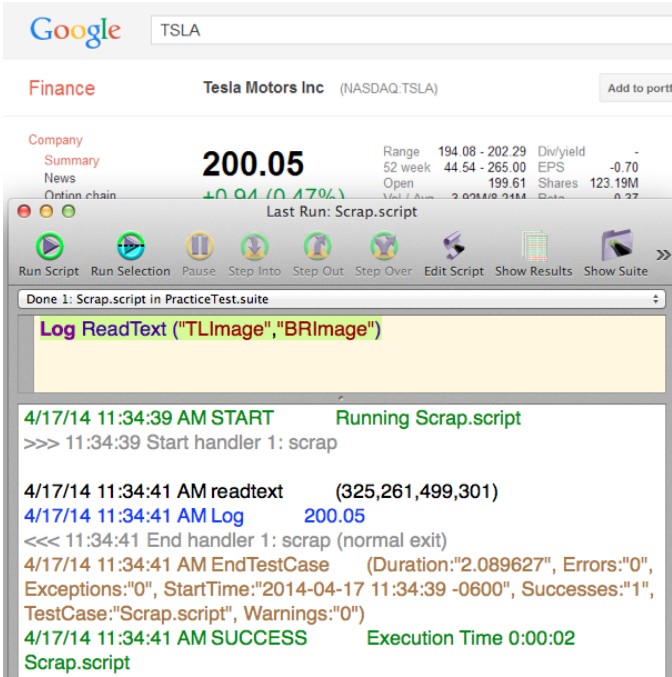


"TLImage"是"Company"标签的图像，热点移动到该区域左上角显示股票价格的地方。



不移动捕获区域，将热点移动到该区域右下角显示股票价格的地方，捕获 "BRImage"图像。

一旦捕获完图像和编写完SenseTalk代码，上述范例就可在谷歌财经页面，读取任意公司的股票价格。



“TLImage”和“BRImage”可用于设置搜索框，通过利用OCR成功利用ReadText()函数搜索。

如果不能使用图像设置搜索框，可使用坐标。在此情况下，光标位置工具栏图标很有用，因为它可显示鼠标在SUT上的当前位置。如何将光标位置添加到工具栏，请参考下文对比色一节。

ValidCharacters

另一个在通常情况下有用的参数是ValidCharacters。ValidCharacters指示OCR应考虑的字符，并忽视其它所有字符。通常可用于防止OCR混淆“O”和“0”。

可手动设置实现对脚本的最佳控制，或者使用validCharacters:"\*"自动将validCharacters设置为被搜索的文本。

继续使用上述例子搜索“Brown”，以下代码可用于手动设置validCharacters：

```
Click (Text: "Brown"? , ValidCharacters: "Brown"? , SearchRectangle: ("UpperLeftImage"? , "LowerRightImage"?))
```

也可像本例一样使用ValidCharacters和变量，首先将所需的字符添加到变量MyText：

```
Put "Brown"? into MyText
Click (Text:MyText, ValidCharacters:MyText)
```

可使用以下代码，将validCharacters设置为搜索的文本：

```
Click (Text:"Brown"? , ValidCharacters:"*"? , SearchRectangle: ("UpperLeftImage"? , "LowerRightImage"?))
```

对比色

如果已经知道搜索文本的背景颜色，可使用Contrast参数设置对比色。

该参数使eggPlant Functional只能分辨黑色和白色。对比色（即背景色）将被转换成白色，其它内容转换成黑色。白底黑色文本代码如下：

```
Click (Text: "hello"? , Contrast:On, ContrastColor:White, ValidCharacters: "hello" , Searchrectangle: ("UpperLeftImage"? , "LowerRightImage"?))
```

该情景下，接近于白色的每个像素（容差范围45以内）将转换成白色，其它内容按黑色处理。这可以帮助OCR更清晰地读取文本。

OCR实际看见的如下图所示，没有采用文本抗锯齿：



Contrast参数设置为ON时OCR所看见的图像

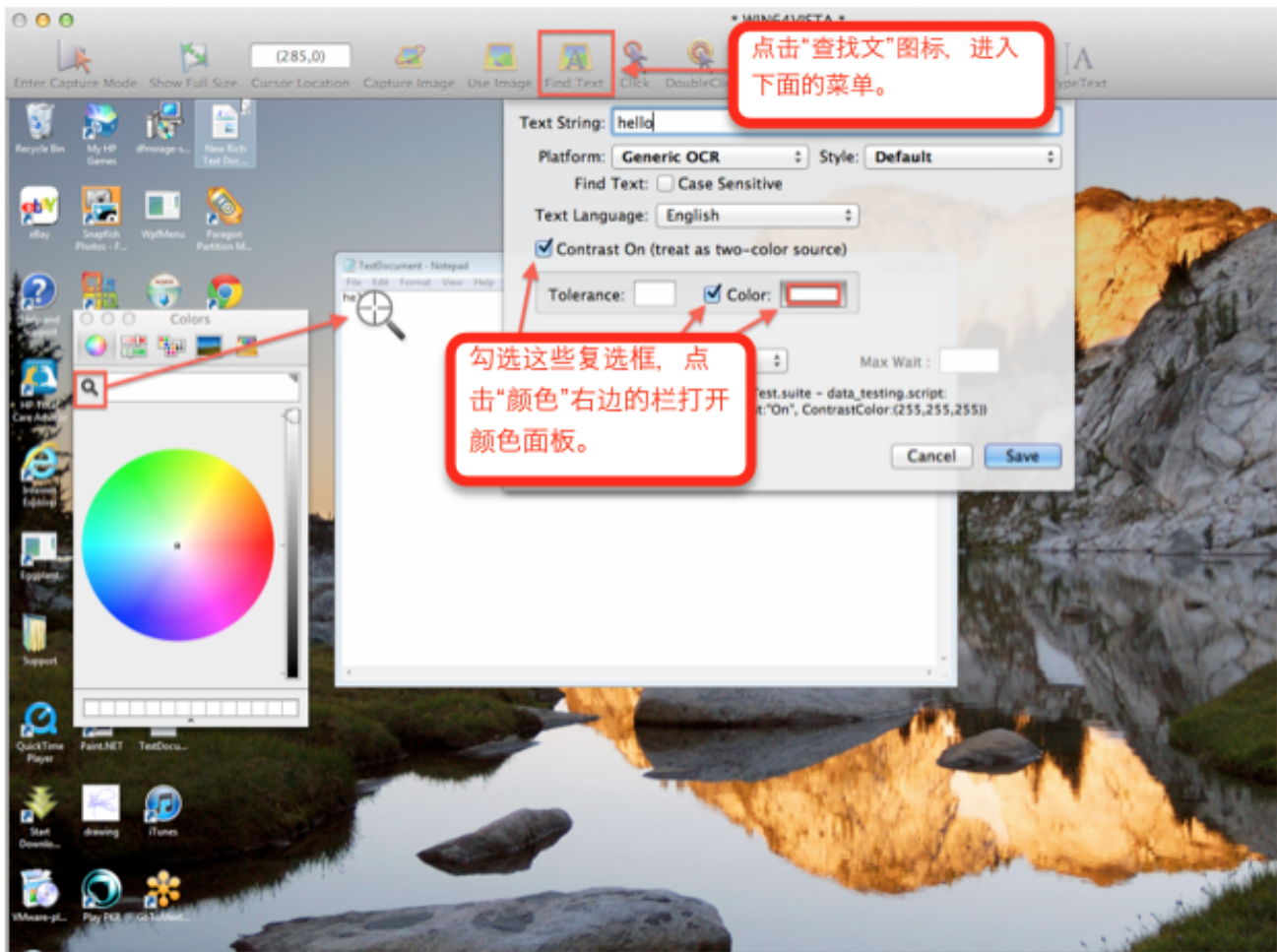
搜索灰色（或其它中间值颜色）背景上的文本稍微有点复杂。最好将ContrastTolerance设置低一点（降低到20左右），减少OCR需要转换为白色的像素数量（注意上图中“hello”中“h”和“e”之间的像素为黑色，将这两个字母连在一起。）本例中，OCR仍然可以读取这些字母，但是在某些情景下，OCR很难处理此类字母。这就是为什么需要在运行完整脚本前测试代码，核查它们如何运行。

```
Click (Text:"hello" , Contrast:On, ContrastColor:White, ContrastTolerance:20)
```

确定背景颜色

如果不知道背景颜色（对比色），有两种方法可确定任何平台上特定位置背景颜色的RGB值。

第一种方法是使用颜色选择器。点击远程屏幕窗口工具栏中的查找文本图标，选择“对比度”复选框。然后选择“颜色”复选框，并点击右边的对话框，选择指定的颜色。弹出此窗口时，点击放大镜，对文本附近的背景颜色取样（确保不要捕获文本抗锯齿的区域）。



使用放大镜, 确定Mac设备上的对比色

第二种方法是, 使用`ColorAtLocation()`函数。现场模式下调出远程屏幕窗口。如果工具栏上没有光标位置图标, 可点击工具栏右边的箭头添加, 选择“自定”, 然后选择光标位置图标。将光标移动到期望RGB颜色值的背景上。使用该光标位置坐标 (显示在光标位置工具栏图标框), 在运行窗口底部的临时执行框中, 或者在脚本中, 运行该行代码:

```
put colorAtLocation(x,y)
-- (x,y) 指在远程屏幕窗口中查找到的坐标。
```

运行该行命令, 将返回指定位置颜色的RGB值。

## IgnoreSpaces

作为文本搜索的一个参数, `IgnoreSpaces`可用于忽视搜索中的任何空格。该功能很有用, 因为字符间的空格不一致, 并且有时候OCR会在没有空格的地方, 也视为有空格, 或者忽略原本存在的空格。`IgnoreSpaces`设置为ON, OCR在搜索“flowerpot”时会匹配“flower pot”, 在搜索“flower pot”时会匹配“flowerpot”。使用该参数时, OCR将对搜索的字符串和查找到有空格的字符串作出匹配。

```
Click (Text:"flower pot", validCharacters:"*?", ignoreSpaces:ON, searchRectangle:("UpperLeftImage?", "LowerRightImage?"))
```

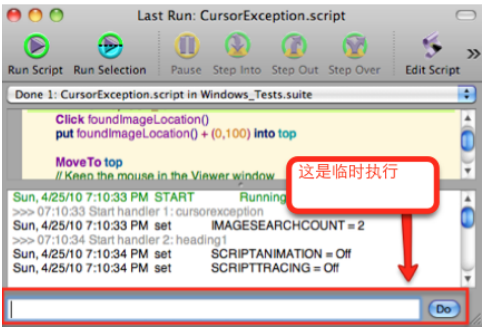
**\*\* 注意利用以下技巧!**

### 技巧1

OCR未查找到你认为应该查找到的文本时, 尝试使用`readText()`函数, 核查OCR查找到的内容。知道OCR查找到的内容后, 就能更好地解决问题。例如, 当所要查找的文本实际上是密码“hello”, 但是却发现`readText()`函数返回“hello”, 设置`validCharacters:"hello"`可使查找结果更精确。

### 技巧2

验证脚本和确认脚本是否按设计目的运行时, 使用临时执行框也很有用。使用临时执行框也可获取其它帮助信息。比如, 让临时执行框返回eggPlant已知的颜色列表, 以及颜色的RGB值。方法是进入临时执行框, 并输入“put the namedColors”。



Contrast参数设置为ON时OCR所看见的图像