

对比度增强技术

此示例显示了几种图像增强方法。三大功能尤其适用于对比度增强：`imadjust`、`histeq`、和`adapthisteq`。此示例比较了它们用于增强灰度和真彩色图像的用途。

Try it in MATLAB

增强灰度图像

使用默认设置，比较以下三种技术的有效性：

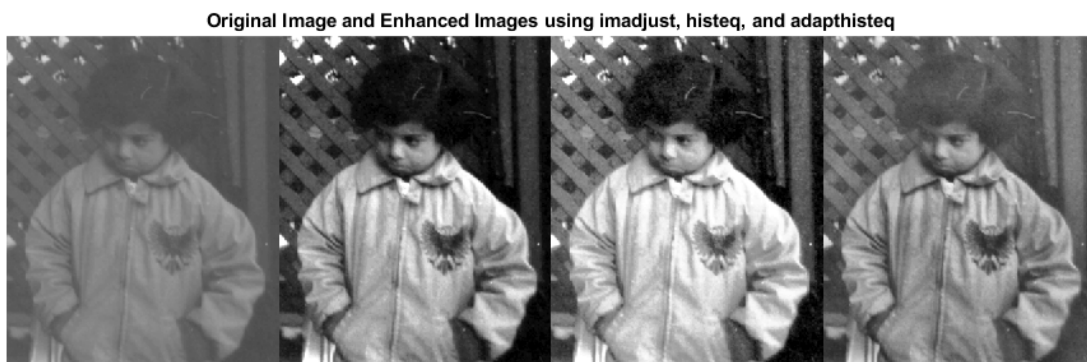
- **`imadjust`** 通过将输入强度图像的值映射到新值来增加图像的对比度，使得默认情况下，1%的数据在输入数据的低强度和高强度下饱和。
- **`histeq`** 执行直方图均衡。它通过变换强度图像中的值来增强图像的对比度，使得输出图像的直方图大致匹配指定的直方图（默认情况下均匀分布）。
- **`adapthisteq`** 执行对比度受限的自适应直方图均衡。`histeq`与之不同，它在小数据区域（图块）上运行，而不是在整个图像上运行。每个图块的对比度都得到增强，因此每个输出区域的直方图大致匹配指定的直方图（默认情况下均匀分布）。可以限制对比度增强，以避免放大可能存在于图像中的噪声。

将灰度图像读入工作区。使用三种对比度调整技术增强图像。

```
pout = imread('pout.tif');  
pout_imadjust = imadjust(pout);  
pout_histeq = histeq(pout);  
pout_adapthisteq = adapthisteq(pout);
```

将原始图像和三个对比度调整后的图像显示为剪辑画面。

蒙太奇（{`pout`, `pout_imadjust`, `pout_histeq`, `pout_adapthisteq`}, '尺寸', [1 4])
标题（“使用`imadjust`、`histeq`和`adapthisteq`的原始图像和增强图像”）



将第二个灰度图像读入工作区，并使用三种对比度调整技术增强图像。

```
tire = imread('tire.tif');  
tire_imadjust = imadjust(轮胎);  
tire_histeq = histeq(轮胎);  
tire_adapthisteq = adapthisteq(轮胎);
```

将原始图像和三个对比度调整后的图像显示为剪辑画面。

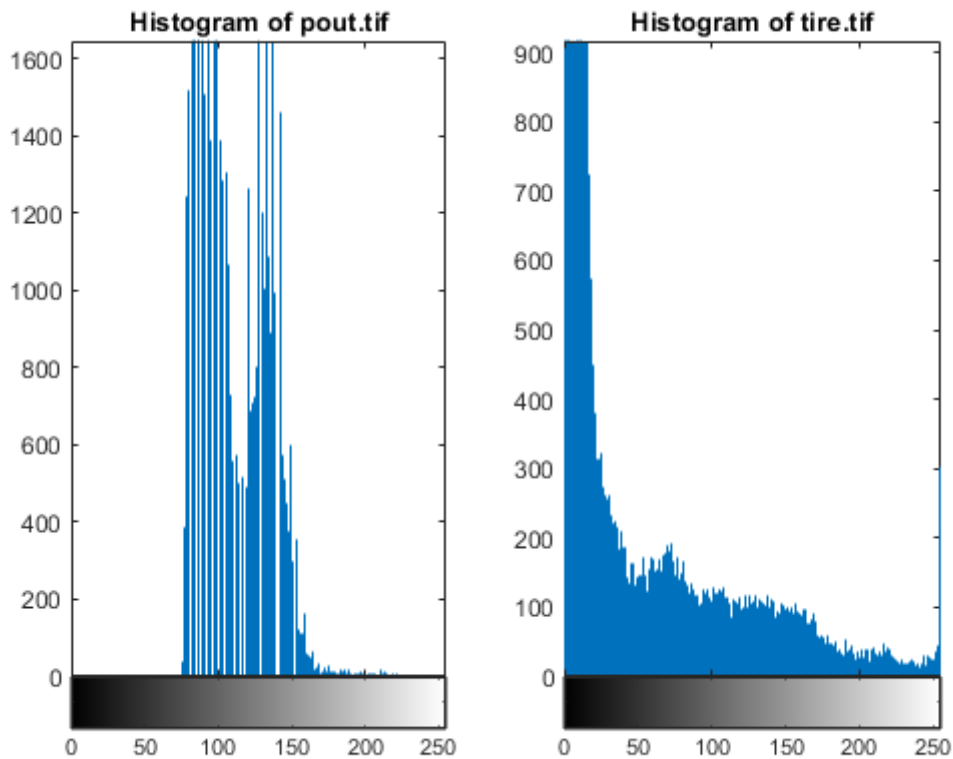
蒙太奇（{`tire`, `tire_imadjust`, `tire_histeq`, `tire_adapthisteq`}, 'Size', [1 4])
标题（“使用`imadjust`、`histeq`和`adapthisteq`的原始图像和增强图像”）

Original Image and Enhanced Images using imadjust, histeq, and adapthisteq



请注意，`imadjust`对轮胎的图像几乎没有影响，但是在噉嘴的情况下它引起了剧烈的变化。绘制直方图 `pout.tif`并`tire.tif`显示第一张图像中的大部分像素都集中在直方图的中心，而在这种情况下 `tire.tif`，这些值已经在最小值0和最大值255之间展开，从而防止 `imadjust`了有效调整图像的对比度。

```
数字
副区 (1,2,1)
imhist (噉嘴)
标题 ('pout.tif的直方图')
副区 (1,2,2)
imhist (轮胎)
标题 ('histogram of tire.tif') ;
```



另一方面，直方图均衡化实质上改变了两个图像。许多先前隐藏的特征被暴露，尤其是轮胎上的碎片颗粒。不幸的是，同时，增强过度使两个图像的几个区域过度饱和。注意轮胎的中心，孩子脸部的一部分和夹克是如何被洗掉的。

集中在轮胎的图像上，优选的是，车轮的中心保持大致相同的亮度，同时增强图像的其他区域的对比度。为了使发生，必须将不同的变换应用于图像的不同部分。实现的对比度限制自适应直方图均衡技术 `adapthisteq`可以实现这一点。该算法分析图像的各部分并计算适当的变换。还可以设置对比度增强水平的限制，从而防止由基本直方图均衡化方法引起的过饱和和`histeq`。这是本例中最复杂的技术。

增强彩色图像

彩色图像的对比度增强通常通过将图像转换为具有图像亮度作为其组件之一的颜色空间来完成，例如L * a * b *颜色空间。仅对亮度层'L'执行对比度调整，然后将图像转换回RGB色彩空间。操纵光度会影响像素的强度，同时保留原始颜色。

将图像读入工作区。的'shadow.tif'图像是索引图像，所以图像转换为真彩色（RGB）图像。然后，将图像从RGB颜色空间转换为L * a * b *颜色空间。

```
[X, map] = imread('shadow.tif');
shadow = ind2rgb(X, map);
shadow_lab = rgb2lab(shadow);
```

亮度值的范围为0到100.将值缩放到范围[0 1]，这是具有数据类型的图像的预期范围double。

```
max_luminosity = 100;
L = shadow_lab(:, :, 1) / max_luminosity;
```

在亮度通道上执行三种类型的对比度调整，并保持a *和b *通道不变。将图像转换回RGB色彩空间。

```
shadow_imadjust = shadow_lab;
shadow_imadjust(:, :, 1) = imadjust(L) * max_luminosity;
shadow_imadjust = lab2rgb(shadow_imadjust);

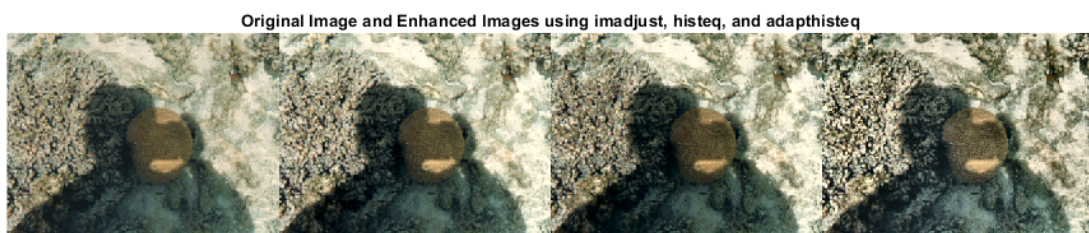
shadow_histeq = shadow_lab;
shadow_histeq(:, :, 1) = histeq(L) * max_luminosity;
shadow_histeq = lab2rgb(shadow_histeq);

shadow_adapthisteq = shadow_lab;
shadow_adapthisteq(:, :, 1) = adapthisteq(L) * max_luminosity;
shadow_adapthisteq = lab2rgb(shadow_adapthisteq);
```

将原始图像和三个对比度调整后的图像显示为剪辑画面。

数字

蒙太奇 ({shadow, shadow_imadjust, shadow_histeq, shadow_adapthisteq}, '尺寸', [1 4])
标题 ("使用imadjust, histeq和adapthisteq的原始图像和增强图像")



也可以看看

[adapthisteq](#) | [histeq](#) | [imadjust](#)

相关话题

- [自适应直方图均衡](#)
- [直方图均衡](#)

