Homework02

1600012960周小妹

一、作业要求

以下两题任选一项完成：

1. 模拟Matlab的imresize()写一个你自己的imresize()函数，至少应实现'nearest'和'bilinear'两种方法。附件的"football"和"kids"可作测试之用。
2. 自行设计算法和编码完成一个图像修补（Image inpainting）的程序，附件的"Lena_damaged"和"inpaintingExample"可作测试之用。

提交代码、实验结果和作业报告。鼓励自行测试更多图像数据。

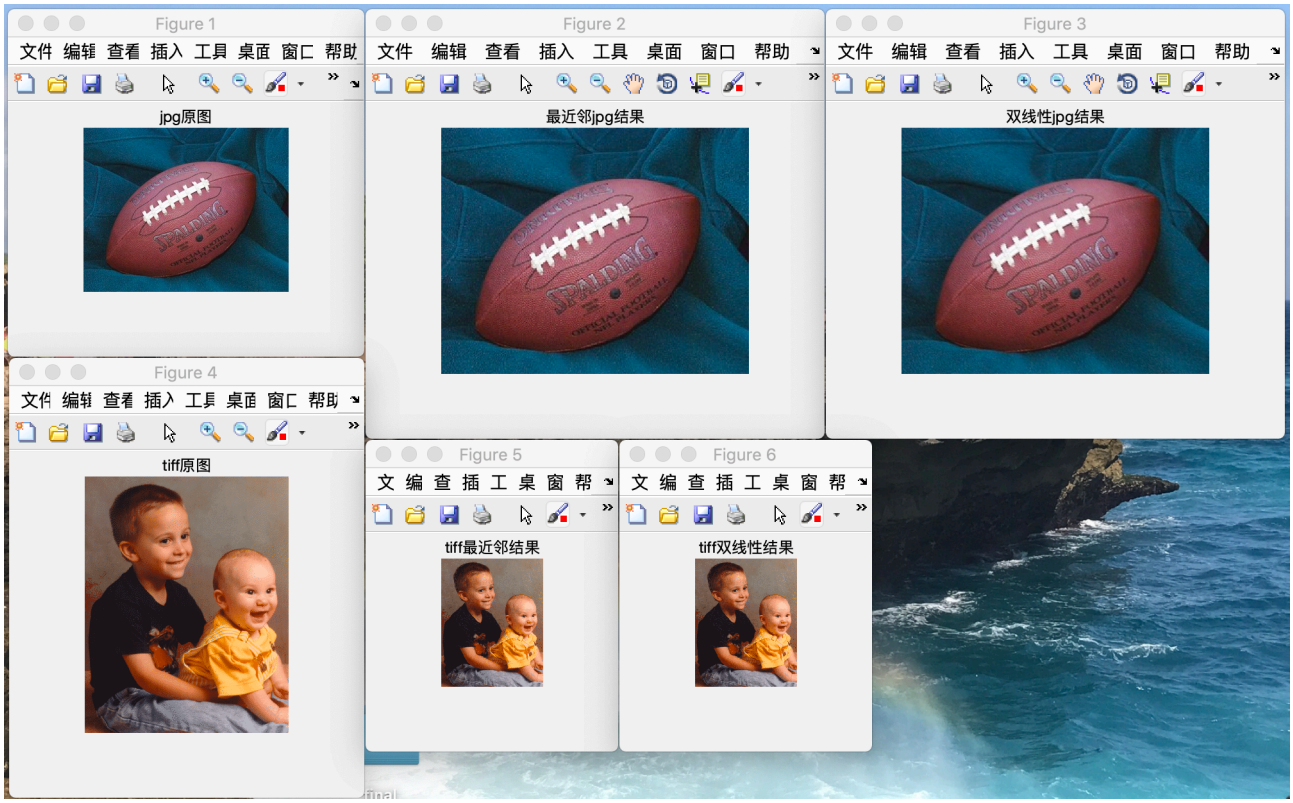
我选择的题目是第一题

二、最终成果

实现了将老师提供的jpg和tiff格式的图像通过两种方式（nearest和bilinear）进行放大和缩小的功能。

测试得到的图像有：

1. footballnearest.jpg：football.jpg通过最近邻插值放大1.5倍得到的图像
2. footballbilinear.jpg：football.jpg通过双线性插值放大1.5倍得到的图像
3. kidsnearest.jpg：kids.tiff通过最近邻插值缩小至0.5得到的图像
4. kidsbilinear.jpg：kids.tiff通过双线性插值缩小至0.5得到的图像



三、具体实现

插值分别通过两个函数实现。

1. mynearestimresize.m：输入一个图像矩阵（inimage）以及放大的倍数（times），输出通过最近邻插值得到的图像大小改变的图像矩阵（outimage）。

先通过size()的到原图像的大小，接下来利用这个大小求出结果图像的高度（outheight）、宽度（outwidth）、通道数的大小（inchanel），并由此创建结果图像的初始化矩阵（outimage）。

对结果矩阵中的每一个点，求出其在原图矩阵中的位置（xres/times，yres/times），并对原图位置进行四舍五入，得到结果矩阵应该取值的原图矩阵中的点坐标（xori，yori），并对边界情况进行特殊的处理，最后再给结果矩阵赋值。

```
function [outimage] = mynearestimresize(inimage, times)

% read the original image and the size of the it
% inimage = imread(file_name);
[inheight, inwidth, inchanel] = size(inimage);

% get the size of the result image and initialize it
outheight = round(inheight * times);
outwidth = round(inwidth * times);
outimage = zeros(outheight, outwidth, inchanel, class(inimage));

% get the pixel value of the result image through
for xres = 1:outheight
    for yres = 1:outwidth
        xori = round(xres/times);
        yori = round(yres/times);
        if xori < 1
            xori = 1;
        elseif xori > inheight
            xori = inheight;
        end
        if yori < 1
            yori = 1;
        elseif yori > inwidth
            yori = inwidth;
        end
        outimage(xres,yres,:) = inimage(xori,yori,:) ;
    end
end
```

2. mybilinearimresize.m：输入一个图像矩阵（inimage）以及放大的倍数（times），输出通过双线性插值得到的图像大小改变的图像矩阵（outimage）。

先通过size()的到原图像的大小，接下来利用这个大小求出结果图像的高度（outheight）、宽度（outwidth）、通道数的大小（inchanel），并由此创建结果图像的初始化矩阵（outimage）。

对结果矩阵中的每一个点，求出其在原图矩阵中的位置（xres/times，yres/times），再求出改点周围的四个整数点，对这四个点处的值进行加权平均，对边界情况进行特殊的处理，给结果矩阵赋值。

```
function [outimage] = mybilinearimresize(inimage, times)

% read the original image and the size of the it
% inimage = imread(file_name);
[inheight, inwidth, inchanel] = size(inimage);

% get the size of the result image and initialize it
outheight = round(inheight * times);
outwidth = round(inwidth * times);
outimage = zeros(outheight, outwidth, inchanel, class(inimage));

% get the pixel value of the result image through
for xres = 1:outheight
    for yres = 1:outwidth
        xori = xres/times;
        xori1 = floor(xori);
        delx = xori - xori1;
        xori2 = xori1 + 1;
        yori = yres/times;
        yori1 = floor(yori);
        dely = yori - yori1;
        yori2 = yori1 + 1;
        if xori1 < 1
            xori1 = 1;
        elseif xori2 > inheight
            xori2 = inheight;
        end
        if yori1 < 1
            yori1 = 1;
        elseif yori2 > inwidth
            yori2 = inwidth;
        end
        outimage(xres,yres,:) = inimage(xori1,yori1,:) * (1 - delx) * (1 - dely)
            + inimage(xori1,yori2,:) * (1 - delx) * dely
            + inimage(xori2,yori2,:) * delx * (1 - dely)
            + inimage(xori2,yori2,:) *  delx * dely;
    end
end
```

四、其他问题

1. 初始化结果矩阵使用zeros(height, width, chanels, class(input))

当class(input)不在其中时，tiff格式的图像能够正常运行，但jpg格式的图像会呈现出全白。加入后就变正常了。

2. tiff格式图像的特殊处理

最初通过main()函数进行测试的时候，我将tiff图片按照jpg一样处理，结果得到的是一张黑白图片，看了课程群中的两天记录，知道了可以通过[X, map] = imread()和ind2rgb将tiff转成rgb矩阵后处理，通过这个方法最终才得到了彩色图片。

```
% 测试tiff图片
[X, map] = imread('kids.tiff', 'tiff');
inimage = ind2rgb(X, map);
figure(4);
imshow(inimage);
title('tiff原图');
% 最近邻
outimage = mynearestimresize(inimage, 0.5);
figure(5);
imshow(outimage);
title('tiff最近邻结果');
imwrite(outimage, 'kidsnearest.tiff', 'tiff');
% 双线性
outimage = mynearestimresize(inimage, 0.5);
figure(6);
imshow(outimage);
title('tiff双线性结果');
imwrite(outimage, 'kidsbilinear.tiff', 'tiff');
```