Homework 5

Introduction

Implement a Noise Filtering function using Adaptive Medium Filtering that can clean up noise on an 8bit gray level image.

Approach

While encourage to use MATLAB my implementation was done with python and OpenCV A package that is useful for image and data processing. I used an Object Oriented approach and created a class for Intensity transformations. I hope to add more algorithms from homework and projects related to Intensity transformations in the future in the file. I hope this will pay off in the future when studying for tests. Currently when the class is called the user must input an image, a mode representing the type of filtering transform, and optional parameters for window size and intensity threshold value for the gamma intensity. After the transformed image is created it is stored in the class object and retrieved using an accessor method. The functions can be called directly to perform transforms on the image. The user would need to set the image using the class. They would then call the intensity transform function to traverse the image and perform the selected function. The function is selected by providing an intensity transform function. The user can call the transform functions in the class, and they can also create their own function and pass it through as a parameter.

Automated usage:

```
path = “my/image/path”
img = cv2.imread(path)
nf = Noise_Filters(img, mode=”amf”, window=1, threshold = 0)
new_img = nf.get_img()
```

Manuel usage:

```
path = “my/image/path”
img = cv2.imread(path)

nf= Noise_Filters(img)
nf.transform_image(nf.adaptive_median_filter)
new_img = ixf.get_img()

def custom_func(local_pixels):
    mid_x = (1,local_pixel.shape[1]-1)
    mid_y = (1,local_pixel.shape[0]-1)
    return custom_function(local_pixels[mid_y,mid_x])
nf.intensity_transform(custom_func)
new_img = nf.get_img()
```

image width - w

image height - h
number of images - i

Time complexity
T(w,h,i) = O(w*h*i)
Space Complexity
S(w,h,i) = O(w*h)

The Transform functions:

adaptive_median_filter= median of pixel window if the center pixel value is considered an outler compared to threshold

median_filter= median of pixel window

Experimental Results

The results of implementing the class were successful but had unexpected consequences. What surprised me at first the drop in quality but I later realized this was due to the image being represented as 8bit.

Adaptive Median Filtering Results

Window_size = 3x3 or border of 1 pixel away from center
Threshold = 0

Window_size = 201x201 or border of 100 pixel away from center
Threshold = 0

Window_size = 3x3 or border of 1 pixel away from center
Threshold = 3

Window_size = 3x3 or border of 1 pixel away from center
Threshold = 2

Discussion

By changing the window size and threshold we get different noise filtering effects. When the threshold is too higher more pixel values with high contrast compared to their local window values won't be filtered out keeping the image somewhat noisy. When the threshold is too low it acts the same as the median filter and edited most pixels to the medium of their local window values. This causes the image to lose its Sharpness. When the window size is too small the noise may be reduced but the image is left with a more gritty effect. This is due to the average median intensities being very localized to the areas of the image. If the window size is too large then this causes the image to lose its contrast, increases blur, and reduces the size of the image as the median values used on each pixel are more likely to result in a global median intensity for more pixels reducing the range of intensities.

Median Filtering Result

Discussion

The medium filter has the same effect as the adaptive medium filter with a threshold of 0. This is because the adaptive medium filter is a more generalized formula of the medium filter by adding the parameters of window size and threshold.

Conclusion

The algorithm was tricky to implement but also very powerful for image noise reductions and highlights the benefits of parameterizing a function to provide more control over an operation.