Digital Image Processing, Spring 2023

Assignment 1 - Report

R11922133, 廖政華, 資工所碩一

Problem 0: WARM-UP

(a) To transform a RGB image to grayscale image, one can simply average the R, G, and B channels as the grayscale value.

$$Y = R/3 + G/3 + B/3$$

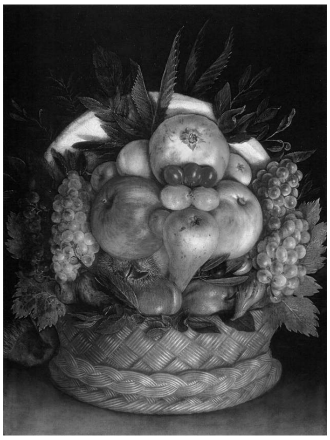
However, these three colors are not perceptually equal to human visual system, *BT.601* recommends a more appropriate weighting:

$$Y = 0.299R + 0.587G + 0.114B$$

(b) To vertically flip an $H \times W$ image, perform the following transformation to each pixel:

$$F[j, k] \leftarrow F[W - j, k]$$

(a) sample1.png



(b) result1.png

(c) result2.png

Problem 1: IMAGE ENHANCEMENT

(a)(b)(c) The histogram of `sample2.png` indicates a large number of low-amplitude pixels in the image, mainly contributed by the cat's body. From the histogram of `result3.png` we see that dividing the pixel values by three transforms the them to a narrower range in the low-amplitude region. The corresponding image appears darker and has poor details. `Result4.png` has a histogram that closely resembles that of `sample2.png`. However, the former appears to be more "discrete". In other words, pixels of some amplitude are absent, while some others appear more frequently in `result4.png`. This occurs because pixel values, which are all integers, are multiplied by three to obtain `result4.png`, resulting in only pixel values that are multiples of three.

(a) sample2.png

(b) result3.png

(c) result4.png

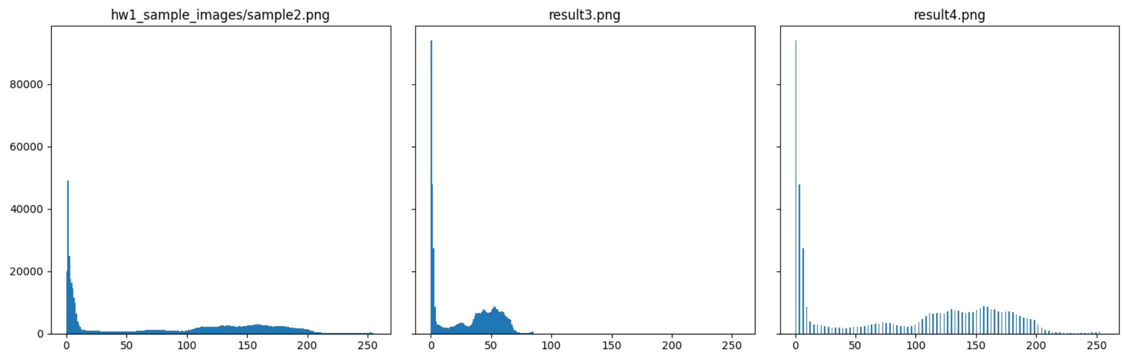


Figure 3: Histogram

(d) Global histogram equalization works as follows: Given a grayscale image, define n_i as the number of occurrences of gray level i. Let $F_I(x)$ be the corresponding cumulative distribution function (cdf)

$$F_I(x) = \sum_{i=0}^{x} n_i.$$

Let $F_T(x)$ be the cdf of target distribution. The transfer function is defined as

$$T(x) = [F_T^{-1}(F_I(x))],$$

in the case of uniform distribution

$$T(x) = [\frac{F_I(x)}{(H \times W)/255}].$$

Performing global histogram equalization has revealed many details of the cat. As we can see in `result5.png`, the right face, the right leg, and the pattern of the cat are clearly visible now. Additionally, the histogram is more evenly distributed across the dynamic range.

In comparison, `result6.png` and `result7.png`, which are visually indistinguishable and the histograms are similar, seem to be of less details. For example, the tabby pattern on the cat's body appears less intricate and the pupil of the cat's right eye is not visible. This loss of detail could be attributed to the sparsity of pixel values as explained in the previous section.

(a) result5.png (b) result6.png (c) result7.png

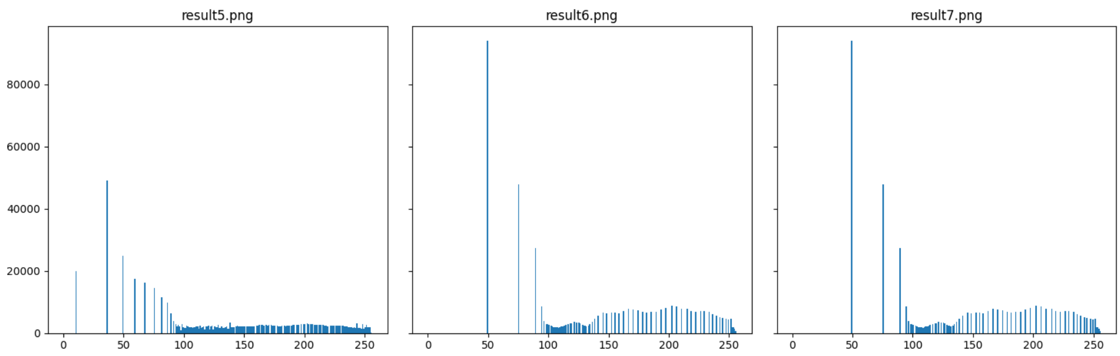


Figure 5: Histogram

(e) In local histogram equalization, the transfer function is calculated individually for each pixel by considering only the cdf of pixels in an $r \times r$ window around the target pixel. Table 1 displays the results for different window size r.

Comparing `result8.png` to `result9.png` and `result10.png`, it can be observed that the former preserves more details. This phenomenon can also be attributed to the same issue described in problem 2-(d). It is also worth noting the presence of a large white area on the cat's body in both `result9.png` and `result10.png`. This happens when the majority of the surrounding pixels fall within the low-amplitude range (in which the cdf approaches the asymptotes very early), resulting in histogram equalization that maps most of the pixel values to the high-amplitude region.

The three histograms look similar, with all of them being more uniformly distributed than the histogram obtained from global histogram equalization.

(a) result8.png

(b) result9.png

(c) result10.png

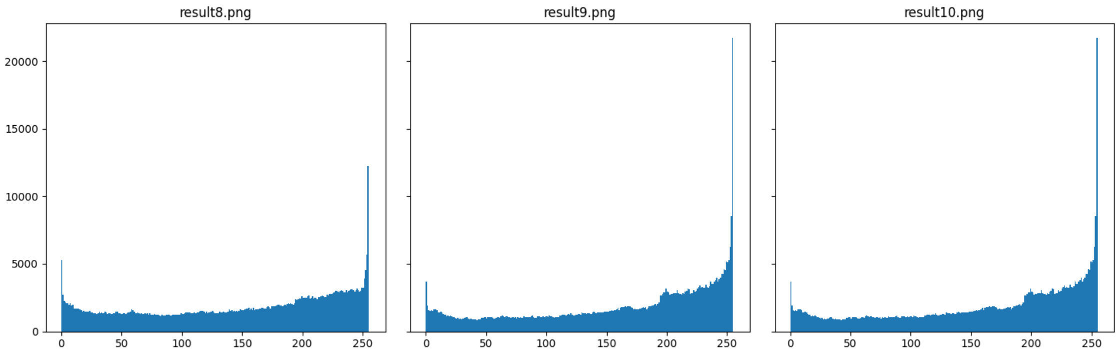


Figure 7: Histogram

Additionally, I implemented an alternative algorithm as proposed in [1], which has lower time complexity but the resulting images suffer from blocky appearance. The results are shown in Table 2.

	result8-*	result9-*	result10-*
$r = 200$			
$r = 100$			
$r = 50$			
$r = 20$			

Table 1: Naive local histogram equalization with different window size r.

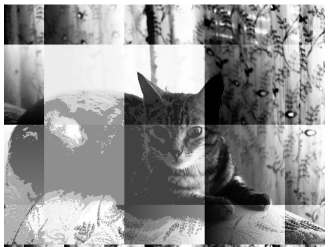
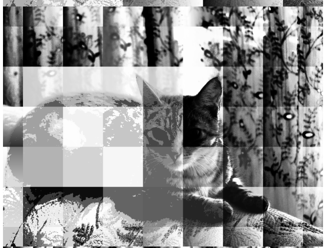
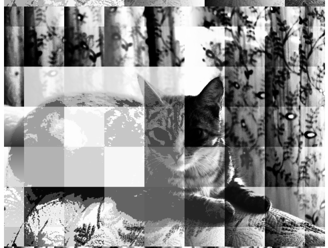
	result8-*	result9-*	result10-*
$r = 200$			
$r = 100$			
$r = 50$			
$r = 20$			

Table 2: Pizer's local histogram equalization with different window size r.

(f) The transfer function is defined as follows:

$$T(x) = \begin{cases} 15\log_2 x + 30, & \text{if } x \leq 120 \\ x, & \text{otherwise} \end{cases}$$

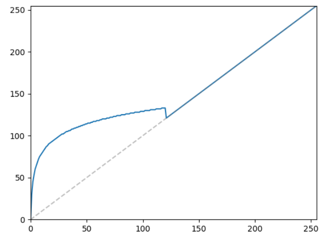


Figure 8: Curve of the transfer function.

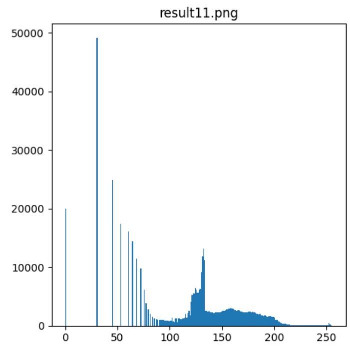
The main focus here is to enhance contrast of the cat's body, which comprises pixel values ranging from 0 to approximately 30. However, as illustrated in Figure 10, setting the control point at 30 results in an unnatural boundary between pixels that undergo transformation and those that do not. Therefore, a higher control point value is selected. For simplicity, a logarithmic function is used as the transfer function for pixel values below the control point, which has two parameters to consider:

$$a\log_2 x + b$$

Table 3 indicates that larger values of a enhance the contrast between neighboring pixel values, and larger values of b simply amplify the amplitude, making pixels appear brighter.

The resulting image also reveals hidden details, such as the right eye and the tabby pattern of the cat, although some details of the curtain in the background are lost.

(a) result11.png



(b) Histogram of result11.png

(a) control point = 30

(b) control point = 60

(c) control point = 120

Figure 10: Transfer function with different control point. ($a = 15$, $b = 30$)

	$b = 15$	$b = 30$	$b = 60$
$a = 5$			
$a = 15$			
$a = 30$			

Table 3: Results of the proposed tranfer function with different a, b.

Problem 2: NOISE REMOVAL

`Sample4.png` is a typical example of uniform noise. Here, I use a Guassian filter $H \in \mathbb{R}^{r \times r}$

$$H(i,j) = \frac{1}{2\pi\sigma^2} e^{\frac{-(i^2+j^2)}{2\sigma^2}}$$

to perform spatial filtering on the given image ($\sigma = 1$, $r = 7$). The image is padded symmetrically for boundary pixels. There are two parameters to consider: variance σ^2 and window size r. As shown in Table 4, increasing both σ and window size results in more noise reduction, but the resulting image becomes blurrier.

sample3.png	sample4.png	result12.png
		
-	PSNR=20.174	PSNR=17.192

`Sample5.png` is a typical example of impulse noise. Here, I use pseudo-median filtering with cross-shape filter of size r ($r = 7$). Specifically, the new pixel value $G[j,k]$ is obtained by

$$G[j,k] = \frac{1}{2} max(MAXMIN(x_C), MAXMIN(y_R) + \frac{1}{2} min(MINMAX(x_C), MINMAX(y_R))$$

where

$$x_C = \{F[j - \lfloor \frac{r}{2} \rfloor, k], ..., F[j + \lfloor \frac{r}{2} \rfloor, k]\}$$
$$y_R = \{F[j, k - \lfloor \frac{r}{2} \rfloor], ..., F[j, k + \lfloor \frac{r}{2} \rfloor]\}$$

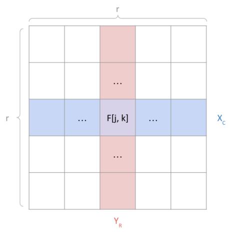


Figure 11 illustrates that there is a significant improvement at $r = 7$. Using a window size larger than 7 will result in a more blurry image, making it less desirable.

sample3.png	sample5.png	result13.png
		
-	PSNR=21.327	PSNR=20.444

While `result12.png` appears less noisy than `sample4.png`, it has a relatively lower PSNR value. The same is true for `result13.png` and `sample5.png`. This indicates that PSNR does not always represent perceived visual quality.

	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$
$r = 3$			
$r = 5$			
$r = 7$			

Table 4: Results of Gaussian filtering with different σ and window size r.

References

[1] Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing 39*, 3 (1987), 355–368.

(a) $r = 3$

(b) $r = 5$

(c) $r = 7$

(d) $r = 9$

(e) $r = 11$

(f) $r = 13$

Figure 11: Results of pseudo-median filtering with different window size r.