

Các nội dung cơ bản về xử lý ảnh.

(Báo cáo Lab tuần 1-2)

Contents

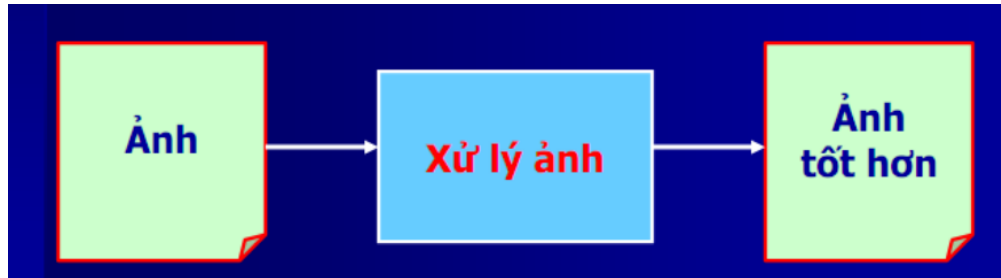
1.	Giới thiệu chung:.....	2
	Xử lý ảnh là?.....	2
2.	Ảnh số:	2
2.1	Điểm ảnh (<i>Pixel</i>) là gì?	2
2.2	Ảnh số:	2
2.3	Xử lý ảnh số (Digital Image Processing):	3
3.	Các phép biến đổi trên miền không gian:	4
3.1	Các phép biến đổi trên miền không gian:	4
3.2	Tổng quan:	4
3.2.	Một số phép biến đổi cấp xám cơ bản.....	6
3.2.1.	Phép biến đổi âm bản (Image Negatives)	6
3.2.2.	Biến đổi Logarit (Log Transformations).....	6
3.2.3.	Biến đổi mũ (Power-Law (Gamma) Transformations).	7
3.3.4.	Another Contrast Stretching Function:.....	8
3.3.5.	Các phép biến đổi tuyến tính từng mảnh (Piecewise-Linear Transformation Functions)	9
4.	Xử lý lược đồ xám (Histogram Processing)	12
4.1.	Lược đồ xám là gì?	12
4.2.	Cân bằng lược đồ xám (Histogram Equalization).....	14
4.3.	Cân bằng lược đồ xám với Python:.....	15
5.	Báo cáo:.....	15
6.	Tài liệu tham khảo.....	16

1. Giới thiệu chung:

Xử lý ảnh là?

- Là một phần của lĩnh vực xử lý tín hiệu số.
- Mục đích: Tăng cường chất lượng thông tin đối với quá trình tri giác của con người và biểu diễn trên máy tính

Có thể hiểu đơn giản qua sơ đồ sau



2. Ảnh số:

2.1 Điểm ảnh (Pixel) là gì?

Hiểu một cách đơn giản, một điểm ảnh sẽ gắn với một tọa độ (x, y) và có giá trị $f(x, y) \in \mathbb{R}^d, d \in \mathbb{Z}^+$

Trong đó:

Nếu $d = 1$: Ảnh đen trắng (Grayscale), $f(x, y) \in [0, 1]$ hoặc $[0, L - 1]$ phụ thuộc vào số lượng bit được sử dụng. Thông thường 8 bit.

Nếu $d = 3$: Ảnh màu (RGB Image), $f(x, y) = [r(x, y), g(x, y), b(x, y)]^T, \dots$

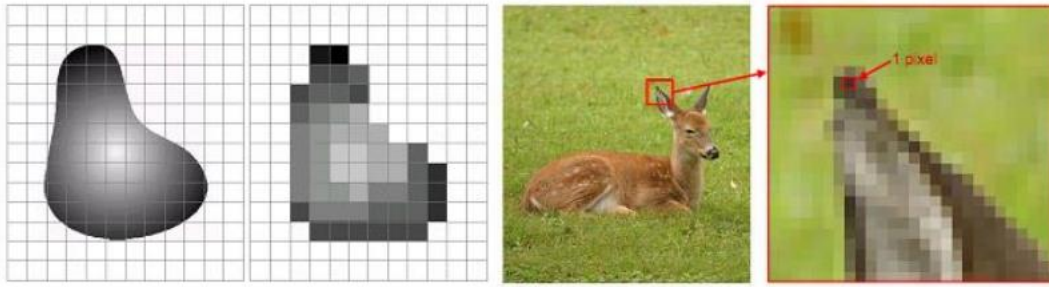
2.2 Ảnh số:

- Digital Image is a representation of a two – dimensional image as a finite set of digital values, called picture elements or pixels

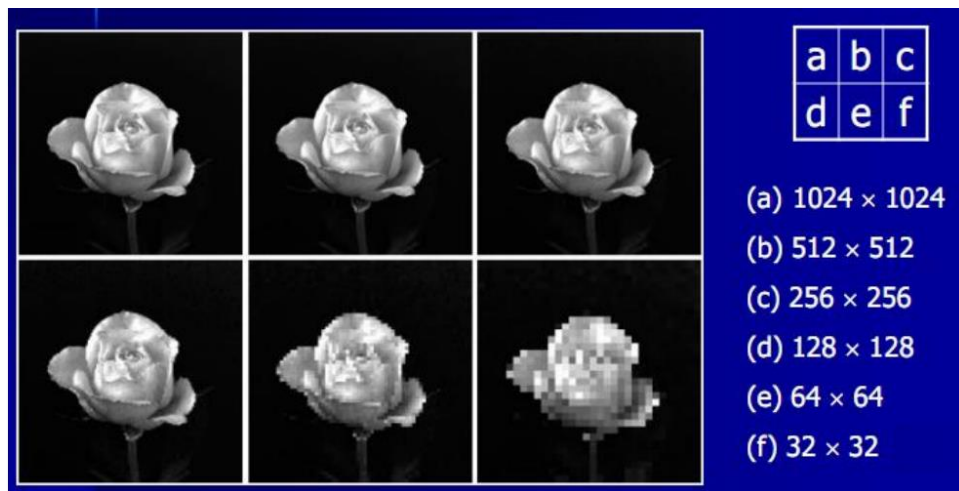
Có thể hiểu cơ bản, ảnh số là tập hợp hữu hạn các pixels!

Ta có thể kí hiệu ảnh số là ma trận $R \in U^{M \times N}, r_{ij} = f(i, j)$

Vai trò của ảnh số đơn giản là số hóa hình ảnh, cũng có thể hiểu ảnh số là một sự sắp xỉ với ảnh gốc!



Tùy vào số lượng pixels ta có khái niệm về độ phân giải (*resolution*), độ phân giải càng cao, ảnh càng gần với ảnh gốc!



2.3 Xử lý ảnh số (Digital Image Processing):

Quay lại sơ đồ tổng quát về xử lý ảnh:



Dễ thấy, ảnh số cũng theo quy tắc trên, tuy nhiên ta có thể tóm gọn nó như sau:

Về mặt toán học, sơ đồ trên có thể hiểu dưới công thức sau:

$$T: U^{M \times N} \rightarrow U^{M \times N}$$

$$R \mapsto T(R)$$

Trong đó:

R là ảnh đầu vào

$T(R)$ là ảnh đầu ra

U là miền giá trị điểm ảnh (có thể là số thực hoặc vector)

3. Các phép biến đổi trên miền không gian:

3.1 Các phép biến đổi trên miền không gian:

- Intensity transformation

- Spatial filtering

3.2 Tổng quan:

Miền không gian (*Spatial Domain*): Hiểu đơn giản là mặt phẳng chứa các điểm ảnh của một bức ảnh.

Các phép xử lý trong miền không gian tác động trực tiếp lên điểm ảnh có thể kí hiệu

$$g(x, y) = T(f(x, y))$$

Trong đó

$f(x, y)$ là ảnh đầu vào

$g(x, y)$ là ảnh đầu ra.

T là toán tử tác động lên f trong lân cận (x, y)

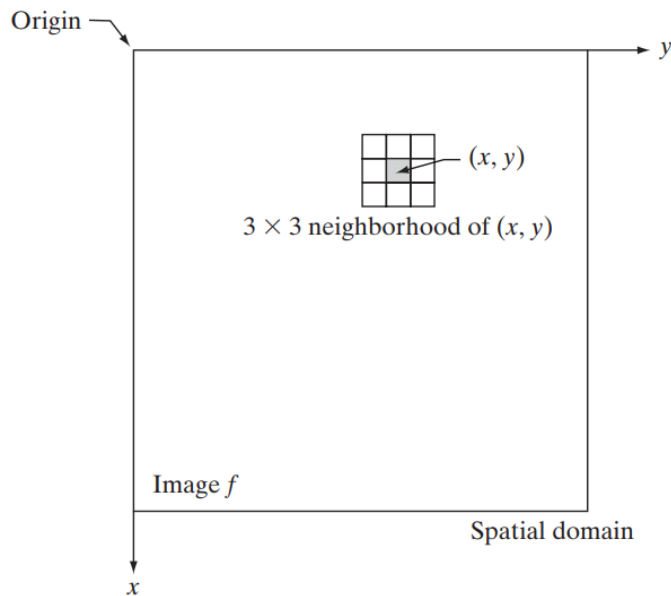


FIGURE 3.1
A 3×3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

Trường hợp cơ bản lân cận bằng 1×1 . Trong trường hợp này, g chỉ phụ thuộc vào f tại từng điểm (x, y) và T . Khi đó biểu thức trên trở thành biến đổi cường độ (biến đổi cấp xám) (*intensity (also called gray-level or mapping) transformation function*), dưới dạng

$$s = T(r)$$

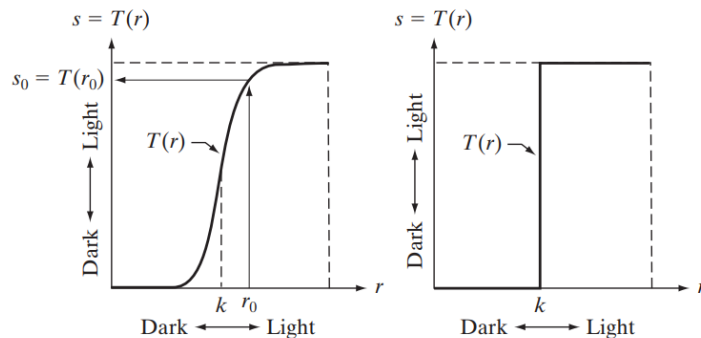
Trong đó

$$r = f(x, y)$$

$$s = g(x, y)$$

a b

FIGURE 3.2
Intensity transformation functions. (a) Contrast-stretching function. (b) Thresholding function.



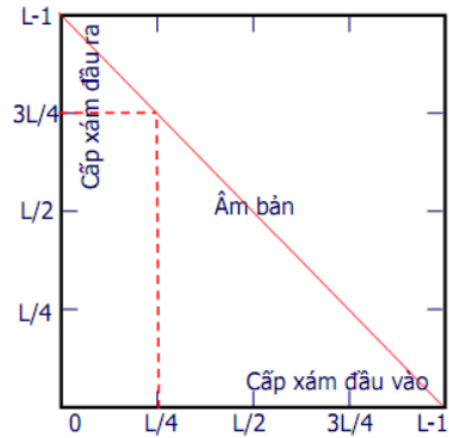
Giới hạn trong bài này, chúng ta sẽ dùng Intensity Transformations chủ yếu cho tăng cường ảnh.

3.2. Một số phép biến đổi cấp xám cơ bản

3.2.1. Phép biến đổi âm bản (Image Negatives)

Ảnh âm bản của ảnh với cấp xám trong khoảng $[0, L - 1]$ có thể thu được từ công thức sau:

$$s = L - 1 - r$$



original gray image



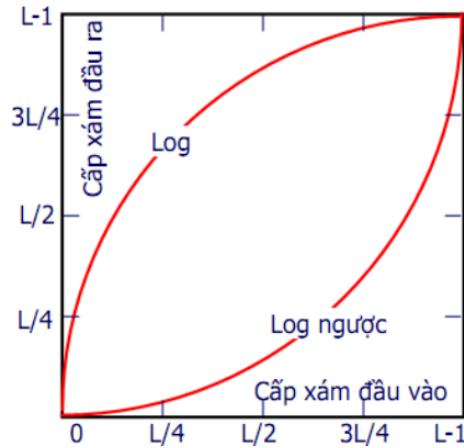
negative image



3.2.2. Biến đổi Logarit (Log Transformations).

Công thức tổng quát cho dạng này

$$s = c \log(1 + r), c = \text{constant}, \text{assume } r \geq 0$$



Nhận xét:

+ Với Biến đổi Log: Giá trị đầu vào với cấp sáng thấp sẽ thành giá trị đầu ra với cấp sáng cao

+ Ngược lại với Log ngược.

Chú ý: *Cấp xám trong ảnh đầu ra sẽ thay đổi, phụ thuộc vào c .*

3.2.3. Biến đổi mũ (Power-Law (Gamma) Transformations).

Công thức chung cho dạng này:

$$s = cr^\gamma, c, \gamma > 0$$

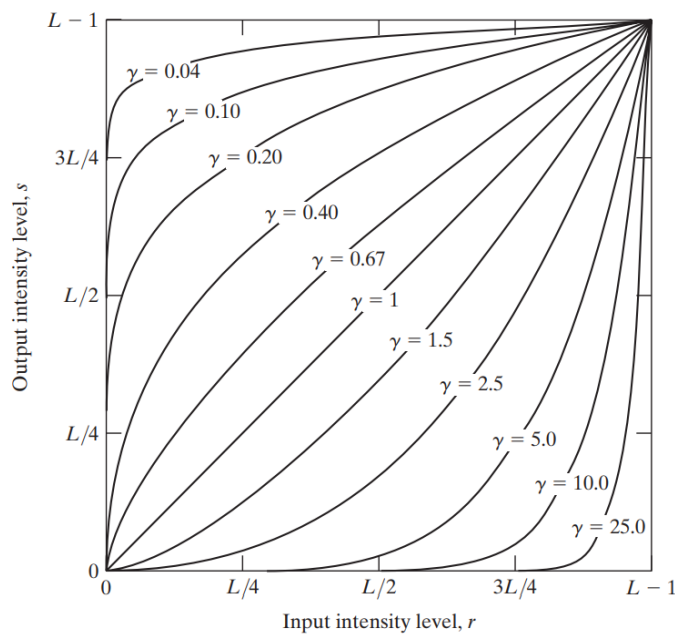


FIGURE 3.6 Plots of the equation $s = cr^\gamma$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.

Lưu ý: Trong phép biến đổi này, trước khi thực hiện ra cần chuẩn hóa ảnh. Tức là đưa tất cả các điểm ảnh về mức xám trong đoạn $[0, 1]$.

a b
c d

FIGURE 3.9

(a) Aerial image.
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0 , respectively. (Original image for this example courtesy of NASA.)



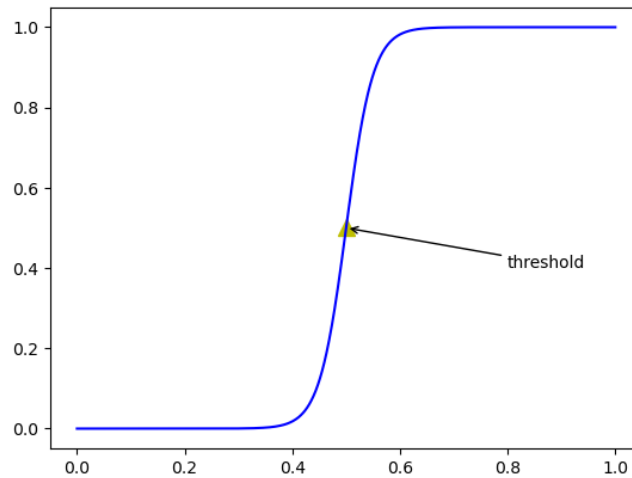
3.3.4. Another Contrast Stretching Function:

Giới thiệu hàm Sigmoid, một hàm khác được sử dụng để dẫn độ tương phản. Ta có:

$$\sigma = \frac{1}{1 + e^{-x}}, x \in \mathbb{R}$$

Nhưng dễ thấy, hàm gốc sẽ có điểm uốn nằm trên trục tung. Tuy nhiên qua các phép tịnh tiến, cùng với các phép đổi biến ta có hàm Sigmoid cho xử lý ảnh:

$$s = \frac{1}{1 + e^{-\gamma(r-c)}}, \quad \gamma, c \geq 0.$$



original gray image



After that



Đặc điểm: Nhờ vào tính mềm mại, những điểm có cấp xám nhỏ hơn điểm uốn sẽ tối hơn và những điểm có cấp xám lớn hơn điểm uốn sẽ sáng hơn.

Ngoài hàm Sigmoid, ta còn có hàm Tanh, cũng là một hàm có tính chất tương tự!

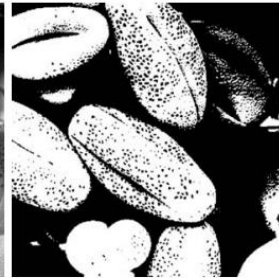
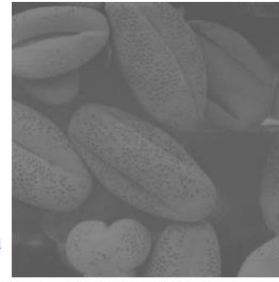
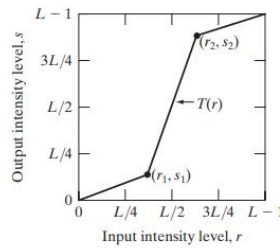
3.3.5. Các phép biến đổi tuyến tính từng mảnh (Piecewise-Linear Transformation Functions)

Giãn độ tương phản (Contrast stretching).

Ý tưởng cơ bản sử dụng hai điểm để điều khiển hình dạng của phép biến đổi

a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

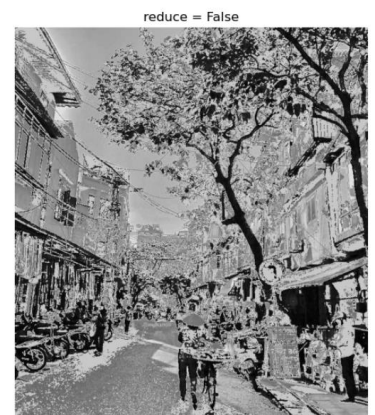
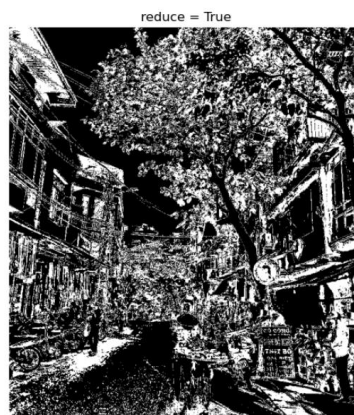
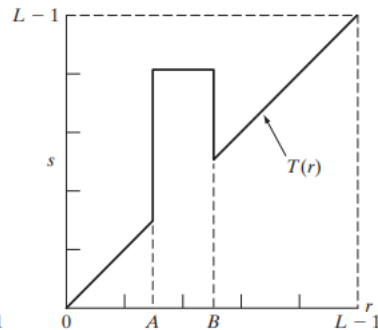
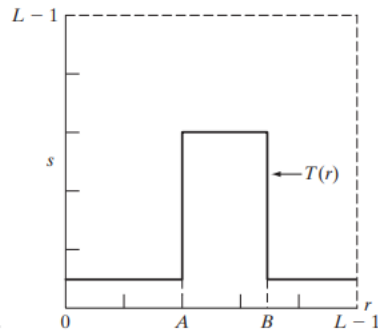


Làm mỏng mức xám (Intensity level slicing)

Ý tưởng cơ bản: Tăng độ sáng của những vùng đáng chú ý, giảm hoặc không giảm những vùng còn lại

a b

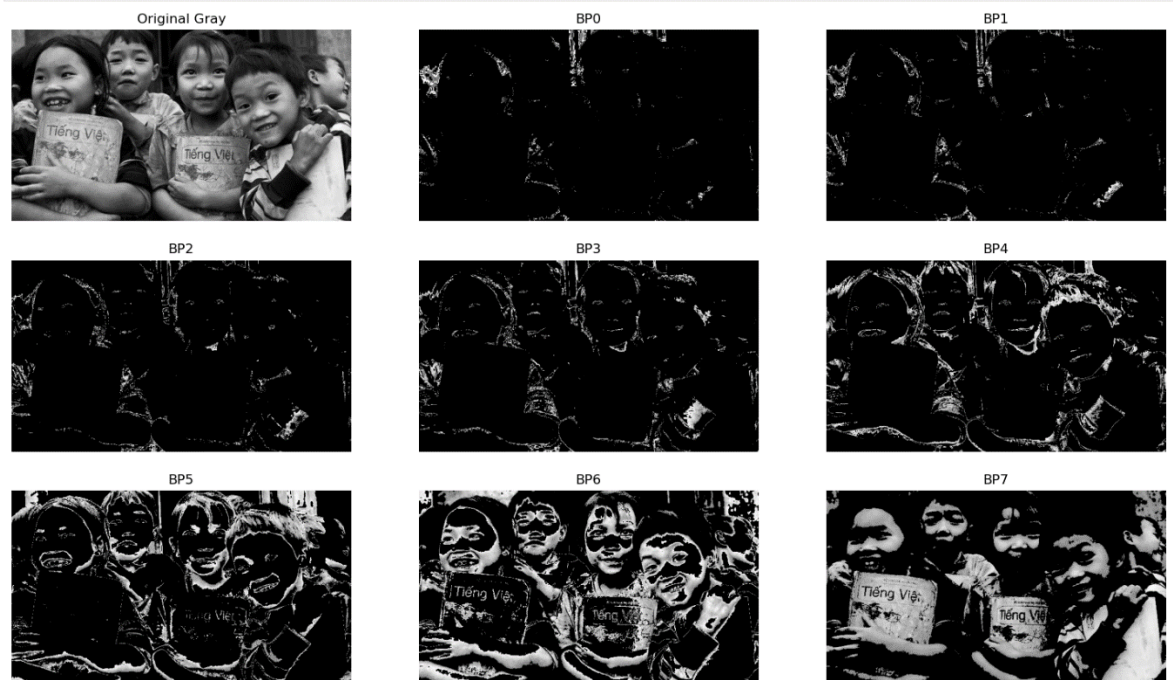
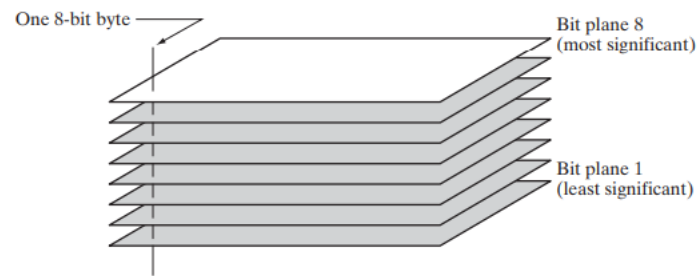
FIGURE 3.11 (a) This transformation highlights intensity range $[A, B]$ and reduces all other intensities to a lower level. (b) This transformation highlights range $[A, B]$ and preserves all other intensity levels.



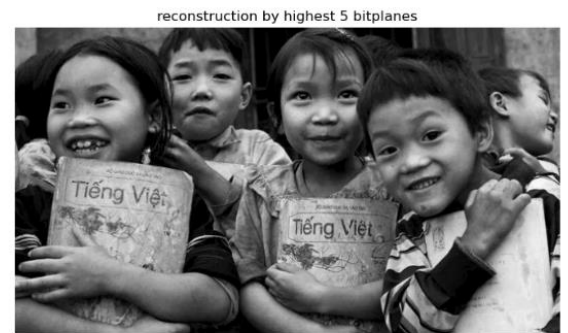
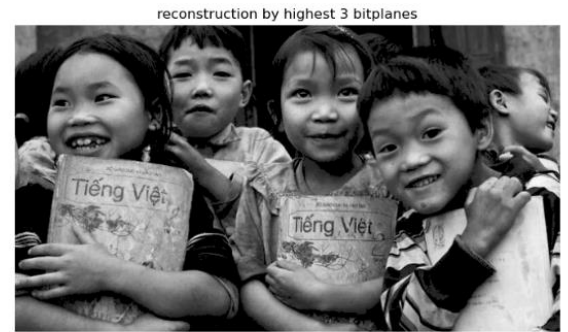
Làm mỏng mức mặt phẳng Bit (Bit-plane slicing)

Giả sử mỗi điểm ảnh được biểu diễn bởi 8 bit. Ta sẽ chia mỗi điểm ảnh về mỗi mặt phẳng bit tương ứng.

FIGURE 3.13
Bit-plane
representation of
an 8-bit image.



Nhận xét: BP7 sẽ chứa nhiều thông tin nhất, ta hoàn toàn có thể phục hồi lại ảnh gốc từ các BitPlane đã chia.



4. Xử lý lược đồ xám (Histogram Processing)

4.1. Lược đồ xám là gì?

Dưới con mắt toán học ta có thể đánh đồng khái niệm trên với “*Biểu đồ tần số*”. Với mẫu chính là ảnh đầu vào, kích thước $M \times N$, $X = [0, 1, 2, \dots, L - 1]$ là các lớp của mẫu.

Table 1.7: Relative Frequency Distribution of Battery Life

Class Interval	Class Midpoint	Frequency, f	Relative Frequency
1.5–1.9	1.7	2	0.050
2.0–2.4	2.2	1	0.025
2.5–2.9	2.7	4	0.100
3.0–3.4	3.2	15	0.375
3.5–3.9	3.7	10	0.250
4.0–4.4	4.2	5	0.125
4.5–4.9	4.7	3	0.075

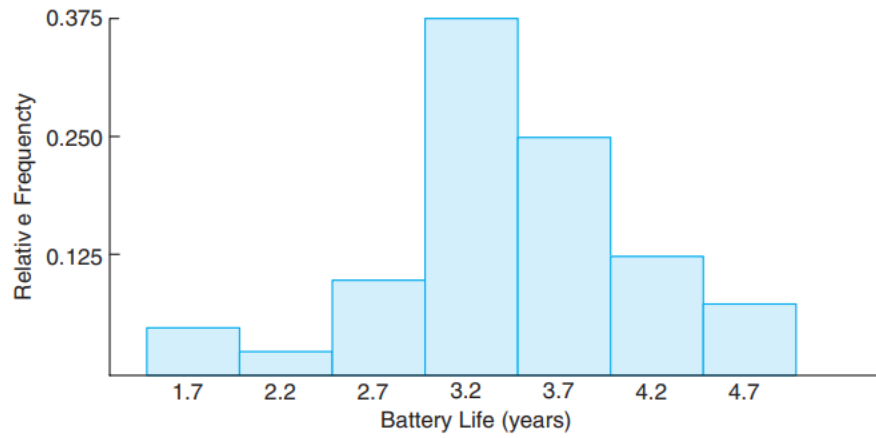


Figure 1.6: Relative frequency histogram.

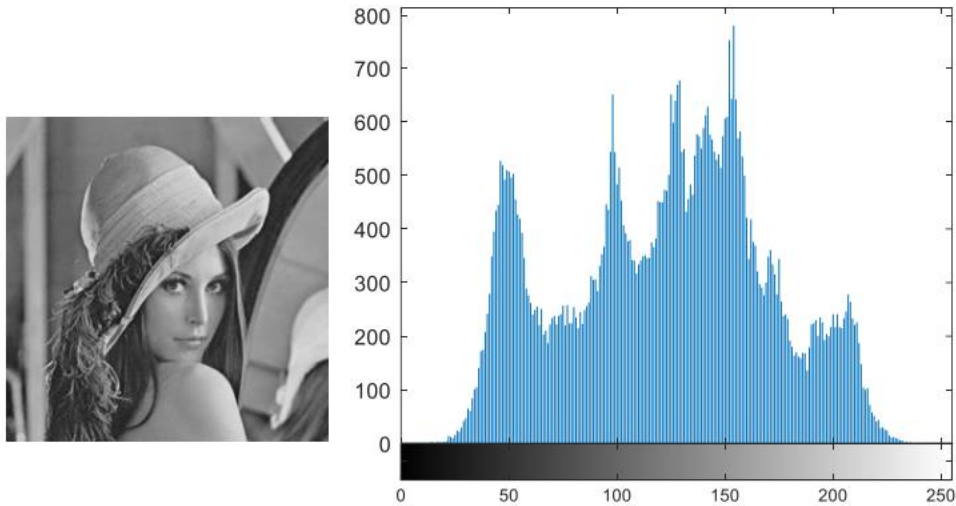
Cụ thể hơn, có thể hiểu như sau:

$$h(k) = n_k$$

k là mức xám, $k = 0, 1, 2, \dots, L - 1$.

n_k là số lượng điểm ảnh trong ảnh có mức xám bằng k .

$h(k)$ là lược đồ xám của ảnh với cấp xám bằng k .

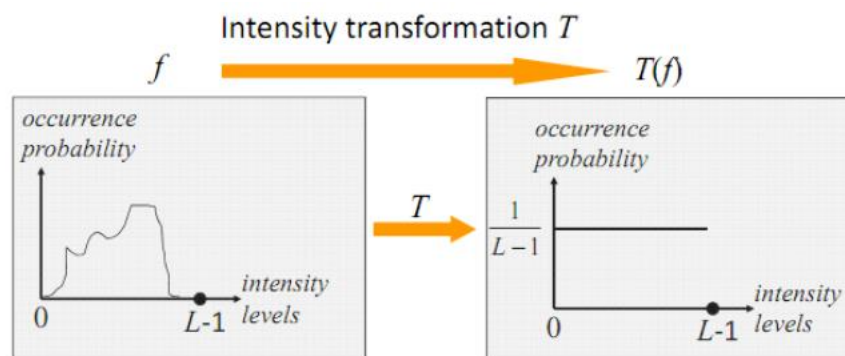


Nếu có yêu cầu chuẩn hóa, $p(k) = n_k/n$.

Lưu ý: Hai bức ảnh khác nhau có thể có cùng lược đồ xám (chỉ khác nhau về phân bố các điểm ảnh). Nên ta không thể khôi phục lại ảnh từ lược đồ xám!

4.2. Cân bằng lược đồ xám (Histogram Equalization).

Mục tiêu: Làm ảnh đầu ra có độ tương phản tốt hơn ảnh đầu vào. Tức là biến lược đồ xám gần nhất với lược đồ phân bố đều



Trong phạm vi bài này, ta chỉ làm việc với các giá trị rời rạc. Để cân bằng lược đồ xám, ta có thể theo phương thức sau:

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j)$$

$$\text{Tương đương, } s_k = T(r_k) = \frac{L - 1}{M - N} \sum_{j=0}^k n_j; 0 \leq k \leq L - 1$$

4.3. Cân bằng lược đồ xám với Python:

MÃ NGUỒN CHO TÀI LIỆU NÀY CÓ THỂ TÌM Ở ĐÂY:

<https://github.com/thuante210823/Computer-Vision-IPSAL-LAB-/tree/main/week%201-2/assignment>

tham khảo các phần sau ở đây:

<https://github.com/thuante210823/Computer-Vision-IPSAL-LAB->

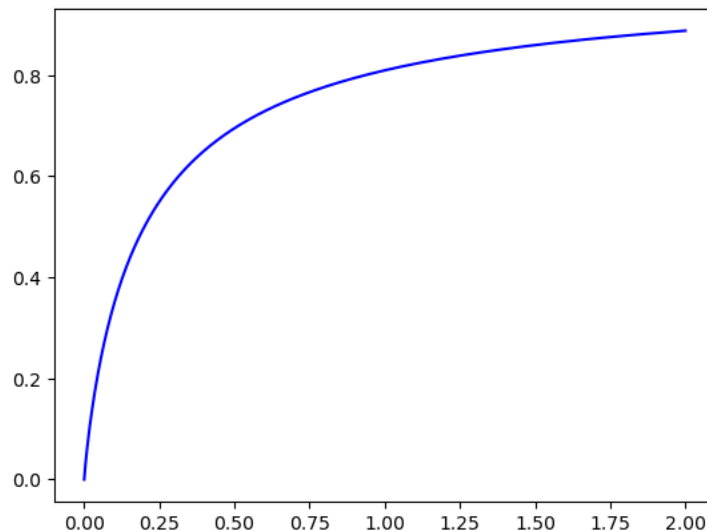
5. Báo cáo:

Trên đây là phần tóm tắt của em cho trong lần báo cáo này!

+ Giới hạn bài báo cáo này chỉ dừng ở Lược đồ xám, theo tiến độ thì sẽ không có phần Cân bằng lược đồ xám, nhưng để đầy đủ em đã thêm vào như ở trên.

+ Em đã thêm một số nhận xét và lưu ý ở mỗi phép biến đổi, trong đó rõ ràng nhất là phép biến đổi mũ, nếu không chuẩn hóa ảnh thì sẽ không có kết quả như lý thuyết.

+ Ngoài ra, trong quá trình học tập em có đề ý có điểm em thấy không hợp lý lắm trong Slide cụ thể là mục Another Contrast Stretching Function. Hàm trong Slide không cho đồ thị như hình vẽ và không cho ra được kết quả như lý thuyết.



Đây là đồ thị của hàm đó, và đây là kết quả với ảnh

original gray image



After that



Tất cả các mức sáng qua hàm trên đều sẽ sáng hơn!

Em đã giới thiệu hàm Sigmoid, và có kết quả khá tốt! Thầy, cô có thể xem qua ở trên ạ!

+ Những kiến thức trên em tự tóm tắt chủ yếu theo ý hiểu, và trong quá trình dịch một số từ chuyên ngành có thể không đúng, em mong thầy cô bỏ qua ạ! Nếu được em rất mong nhận được phản hồi góp ý của thầy và cô ạ!

+ Khó khăn với em có lẽ chính là tự mày mò code, và học thư viện. Vì em chưa được học qua Matlab trước đó, cho nên thay vì tham khảo code có sẵn trên slide em phải tự học thư viện và tự viết chương trình! Âu, đây cũng quả là một cơ hội, qua đây em rèn luyện đọc rất nhiều về kỹ năng sử dụng Matplotlib, cũng như Numpy của mình, đồng thời cũng biết thêm các thư viện mới như OpenCV, PIL, PyTorch. Hiện tại em chưa cài được OpenCV trên conda. Và em cũng khá lười dùng Colab, thỉnh thoảng em cứ bị gặp lỗi, em hi vọng sẽ khắc phục được trong tương lai ạ!

+ File báo cáo này thực ra e làm xong lâu rồi! Nhưng mà hôm nay em gõ lại do em không thể đi được buổi liên hoan ngày mai ạ! Em và các bạn lớp em có lịch thi giữa kỳ môn VLĐC 2 vào đúng trưa mai! Em hi vọng thầy cô và các anh chị trong Lab bỏ qua ạ!

+ Còn về báo, em thấy đọc khó hơn em tưởng ạ! Em đã định sẽ hoàn thành nhưng chắc lần này em không thể hoàn thành được rồi ạ, do đợt assignment lần này cũng sát lịch thi giữa kỳ... và em cũng khá bị bánh cuốn vào những đường bóng WC 😞. Em xin lỗi ạ! Em sẽ cố gắng để trong lần nộp bài tới ngoài những yêu cầu bắt buộc thì em sẽ hoàn thành tóm tắt bài báo kia ạ!

+ Đề xuất của em chắc là sách thôi ạ, em thấy hơi bí về việc tham khảo sách vở, em thấy mình hiểu những cái này rất nhanh, hiện tại về xử lý ảnh thôi em cũng đã xong tuần 3-4, em có đọc qua tuần 5-6 và thấy có vẻ sẽ khó hiểu hơn, và slide thôi chắc là không đủ ạ! Còn về mấy cái tutorial, em chưa biết học như thế nào, với hình như cô cũng up thiếu ạ, em phải vào teams CV2021 của các achi K63 thì mới có đủ, nhưng riêng phần này, em mong sẽ được thầy cô hướng dẫn kỹ ạ, và em cũng mong sẽ có nhiều sách tham khảo về những phần này ạ!

+ Em cũng muốn hỏi về việc nộp assignment một lúc 2 bài, em không biết có được không ạ?

+ Với em thì hai tuần qua cũng khá căng thẳng ạ, chắc do em tự làm khó mình. Trước đó song song với việc học em có tự học ML nhưng giờ em đang phân vân ạ, không có định hướng rõ ràng, em rất hi vọng sau thi giữa kỳ này, em có thể trao đổi để hỏi ý kiến thầy cô ạ, do đây không phải là việc trong Lab em mong thầy cô nếu được có thể giúp em ạ!

+ Trong link github ở trên sẽ có chắc là sẽ có những phần của cả đợt assignment sau ạ, thầy cô có thể xem qua ạ!

6. Tài liệu tham khảo.

1. TRUONG. PV, THAO. TT, COMPUTER VISION LECTURE 1 (PART2): INTRODUCTION TO DIGITAL IMAGE PROCESSING.

2. TRUONG. PV, THAO. TT, COMPUTER VISION LECTURE 3: INTENSITY TRANSFORMATIONS AND SPATIAL FILTERING.
3. R.C. GONZALEZ AND R.E. WOODS, DIGITAL IMAGE PROCESSING (3rd EDITION), PRENTICE HALL, 2008.
4. R. WALPOLE, R. MYERS, S. MYERS, K. YE, PROBABILITY & STATISTICS FOR ENGINEERS & SCIENTISTS (NINTH EDITION), PRENTICE HALL.
5. QUY. TD, GIÁO TRÌNH XÁC SUẤT THỐNG KÊ (5th EDITION), NHÀ XUẤT BẢN BÁCH KHOA -HÀ NỘI.