AI VIET NAM
@aivietnam.edu.vn

Statistics and Its Applications

(Mean, Median, Standard Deviation, PMF, PDF & CDF)

Vinh Dinh Nguyen
PhD in Computer Science

Outline

- Mean and Its Application
- Median and Its Application
- Variance and Its Application
- What is a PMF, PDF, and CDF?
- Histogram
- Histogram Equalization
- Summary

Mean

❖ Definition

Data

$$X = \{X_1, \dots, X_N\}$$

Given the data

$$X = \{2, 8, 5, 4, 1, 4\}$$

$$N = 6$$

Formula

$$\mu_X = \frac{1}{N}\sum_{i=1}^{N} X_i$$

$$\mu_X = \frac{1}{N}\sum_{i-1}^{N} X_i = \frac{1}{6}(2 + 8 + 5 + 4 + 1 + 4)$$

$$= \frac{24}{6} = 4$$

Mean

Data

$$X = \{X_1, \dots, X_N\}$$

Đây là gì vậy nhỉ?

Formula

$$E(X) = \sum_{i=1}^{N} X_i P_X(X_i)$$

$$P_X(X = 2) = \frac{1}{6} \qquad P_X(X = 4) = \frac{2}{6}$$

$$P_X(X = 8) = \frac{1}{6} \qquad P_X(X = 1) = \frac{1}{6}$$

$$P_X(X = 5) = \frac{1}{6}$$

Given the data

$$X = \{2, 8, 5, 4, 1, 4\}$$

$$N = 6$$

$$E(X) = 2 \times \frac{1}{6} + 8 \times \frac{1}{6} + 5 \times \frac{1}{6} + 4 \times \frac{2}{6} + 1 \times \frac{1}{6}$$

$$= \frac{2}{6} + \frac{8}{6} + \frac{5}{6} + \frac{8}{6} + \frac{1}{6} = 4$$

Expected Values

Given 100 students's information in AIO2025:

Love Math	Not Love Math	Total
30	70	100

Game: Bạn sẽ mất 1$ nếu sinh viên kế tiếp tham gia vào lớp AIO2025 thích toán. Bạn có tham gia không?

What is the probablity that a randomly selected student in AIO 2025 love Math: 30/100 = 0.3

What is the probablity that a randomly selected student in AIO 2025 love Math: 70/100 = 0.7

Thay vì chơi gam này 1 lần. Tôi có thể chơi game này 100 lần không?

	Love Math	Not Love Math	Total
# of students	30	70	100
Probability	0.3	0.7	1.0
Outcome	-1	+1	

Xác xuất bị mất 1$ Xác xuất thắng 1$

Expected Values

	Love Math	Not Love Math	Total
# of students	30	70	100
Probability	0.3	0.7	1.0
Outcome	-1	+1	

Times you will lose: 100 x 0.3 = 30

Times you will win: 100 x 0.7 = 70

Money you will lose: 100 x 0.3 x (-1)=-30

Money you will win: 100 x 0.7 x (+1) = 70

The expected number of losses

The expected number of wins

Expect to gain: -30 + 70 = 40$, after 100 games

On average, We expect to gain 0.4$ for each game

Average amount of money will gain per game: 40/100 = 0.4$

Expected value

Expected Values

	Love Math	Not Love Math	Total
# of students	30	70	100
Probability	0.3	0.7	1.0
Outcome	-1	+1	

Expect to gain: -30 + 70 = 40$, after 100 games

Average amount of money will gain per game: 40/100 = 0.4$

$$E(\mathrm{X}) = \frac{[(0.3 \times 100) * -1[+[(0.7 \times 100) * 1]}{100} = 0.4$$

Expected value

$$E(\mathrm{X}) = (0.3 * -1) + (0.7 * 1) = 0.4 = \sum \mathrm{x} * \mathrm{P}(\mathrm{X} = \mathrm{x})$$

Specific outcome

The probablity for the specific outcome

Expected Values

	Love Math	Not Love Math	Total
# of students	30	70	100
Probability	0.3	0.7	1.0
Outcome	-1	+1	

Expected value

$$E(\mathrm{X}) = (0.3 * -1) + (0.7 * 1) = 0.4\$ = \sum \mathrm{x} * \mathrm{P}(\mathrm{X} = \mathrm{x})$$

Specific outcome

The probablity for the specific outcome

Tôi sẽ trả cho bạn 10$ nếu sinh viên kế tiếp tham gia AIO2025 thích Toán, Nhưng nếu ko phải bạn phải thua tôi 1$?

What do you think about this game, are you willing to play?

$$E(\mathrm{X}) = (0.3 * 10) + (0.7 * -1) = 2.3\$ = \sum \mathrm{x} * \mathrm{P}(\mathrm{X} = \mathrm{x})$$

Mean Vs. Expected Value

Expected value represents the average value we expect to occur before collecting any data.

$$E(X) = \sum_{i=1}^{N} X_i P_X(X_i)$$

Mean represents the average value of raw data that we've already collected.

$$\mu_X = \frac{1}{N} \sum_{i=1}^{N} X_i$$

AI VIET NAM
@aivietnam.edu.vn

Mean

❖ Application

```
1.  def calculate_mean(numbers): #1
2.      s = sum(numbers)         #2
3.      N = len(numbers)         #3
4.      mean = s/N               #4
5.      return mean              #5
6.
7.  # Tạo mảng donations đại diện cho số tiền quyên góp trong 12 ngày
8.  donations = [100, 60, 70, 900, 100, 200, 500, 500, 503, 600, 1000,1200]
9.
10. mean_value = calculate_mean(donations)
11. print('Trung bình số tiền quyên góp là: ', mean_value)
```

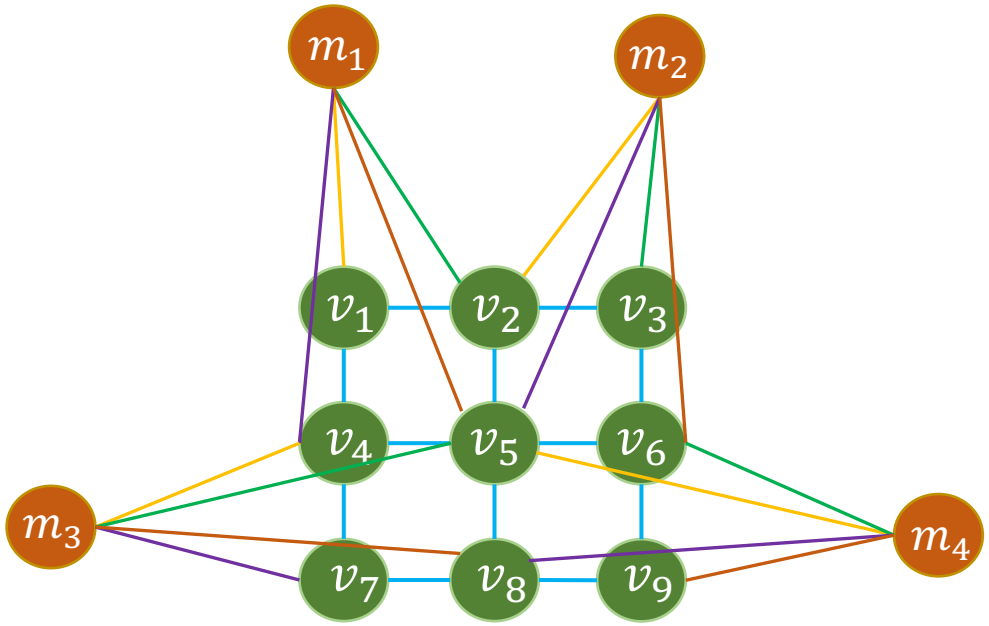
Làm mờ ảnh dựa vào mean →

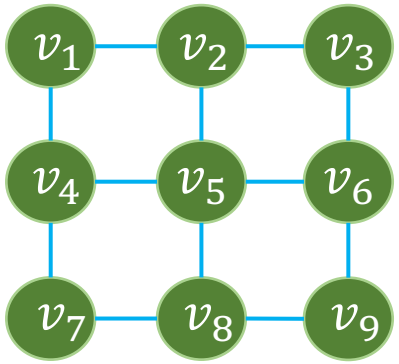
Mean Applications

❖ Correlation (~convolution)

Kernel

$$m_1 = v_1 w_1 + v_2 w_2 + v_4 w_3 + v_5 w_4$$



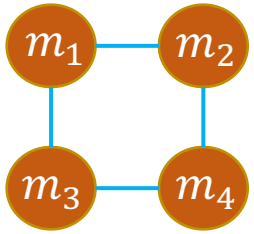


Image

*

Kernel

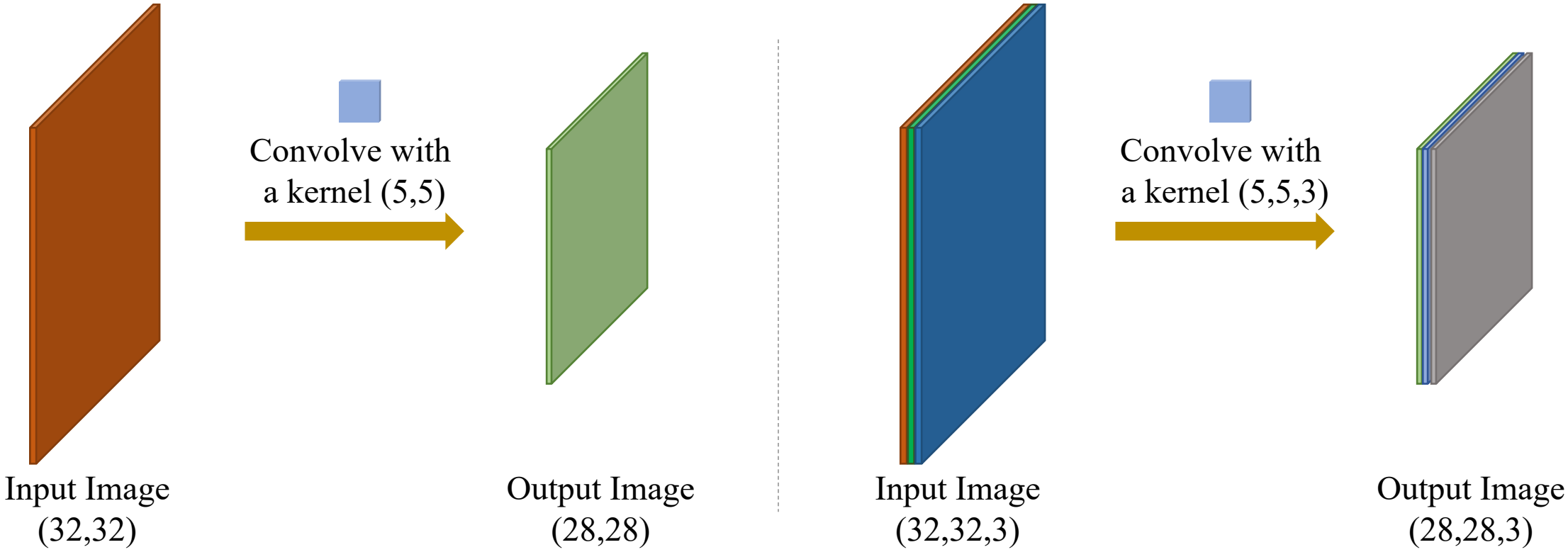
➡



Output

Mean Applications

❖ Correlation (~convolution)



https://medium.com/apache-mxnet/multi-channel-convolutions-explained-with-ms-excel-9bbf8eb77108

Mean Applications

❖ Correlation (~convolution)

Numpy	np.einsum()
Scipy	scipy.signal.convolve2d()
OpenCV	cv2.filter2D()

Kernels for computing mean

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

(3x3) kernel

$$\frac{1}{25}\begin{bmatrix}1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\end{bmatrix}$$

(5x5) kernel

```
output_image = cv2.filter2D(input_image, cv2.CV_8U, kernel)
```

AI VIET NAM
@aivietnam.edu.vn

Mean

❖ Correlation (~convolution)

```python
# load image and blurring

import numpy as np
import cv2

# load image in grayscale mode
image = cv2.imread('mrbean.jpg', 0)

# create kernel
kernel = np.ones((5,5), np.float32) / 25.0

# compute mean for each pixel
dst = cv2.filter2D(image, cv2.CV_8U, kernel)

# show images
cv2.imshow('image', image)
cv2.imshow('dst', dst)

# waiting for any keys pressed and close windows
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Given the face region

AI VIET NAM
@aivietnam.edu.vn

Numpy Review

❖ Slicing

arr[for_axis_0, for_axis_1, …]

‘**:**’: get all the elements
‘**a:b**’: get the elements from a^{th} to $(b^{th}-1)$

data

	0	1
0	1	2
1	3	4
2	5	6

data[0, 1]

	0	1
0	1	2
1	3	4
2	5	6

data[1: 3]

	0	1
0	1	2
1	3	4
2	5	6

data[0: 1 , 0]

	0
0	1
1	3
2	5

data[: , :]

	0	1
0	1	2
1	3	4
2	5	6

```python
# aivietnam.ai
import numpy as np

# Khởi tạo numpy array a_arr
a_arr = np.array([[1,2,3],
                  [5,6,7]])

# Sử dụng slicing để tạo mảng b_arr
# bằng cách lấy tất cả các dòng và cột 1,2
b_arr = a_arr[:, 1:3]

print(a_arr)
print(b_arr)
```

```
[[1 2 3]
 [5 6 7]]
[[2 3]
 [6 7]]
```

a_arr =

	0	1	2
0	1	2	3
1	5	6	7

b_arr = a_arr[:, 1:3]

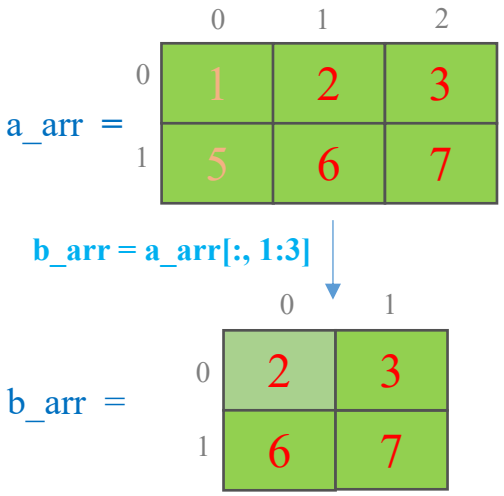
b_arr =

	0	1
0	2	3
1	6	7

Numpy Review

❖ Slicing

❖ Mutable



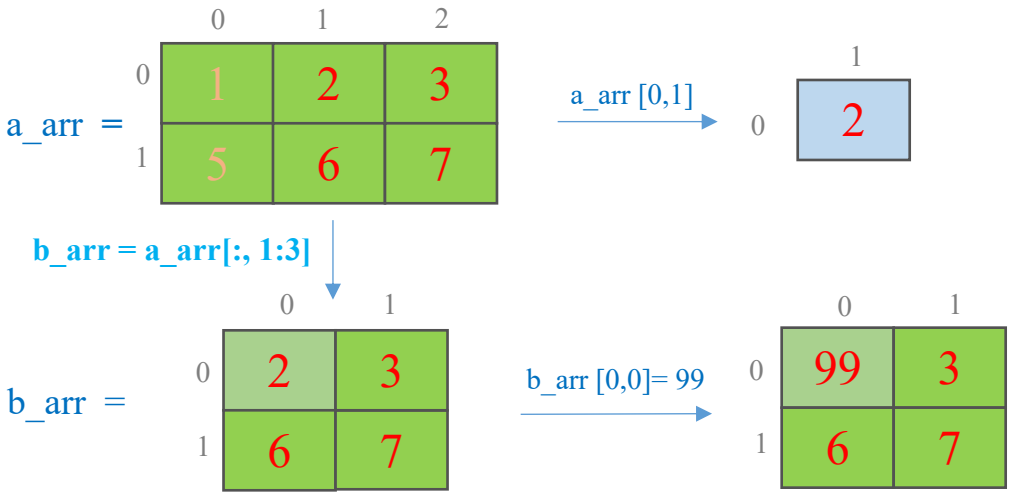

```python
# aivietnam.ai
import numpy as np

# Khởi tạo numpy array a_arr
a_arr = np.array([[1,2,3],
                  [5,6,7]])
print('a_arr \n', a_arr)

# Sử dụng slicing để tạo mảng b_arr
b_arr = a_arr[:, 1:3]
print('b_arr \n', b_arr)
```

AI VIET NAM
@aivietnam.edu.vn

Numpy Review

❖ Slicing

❖ Mutable



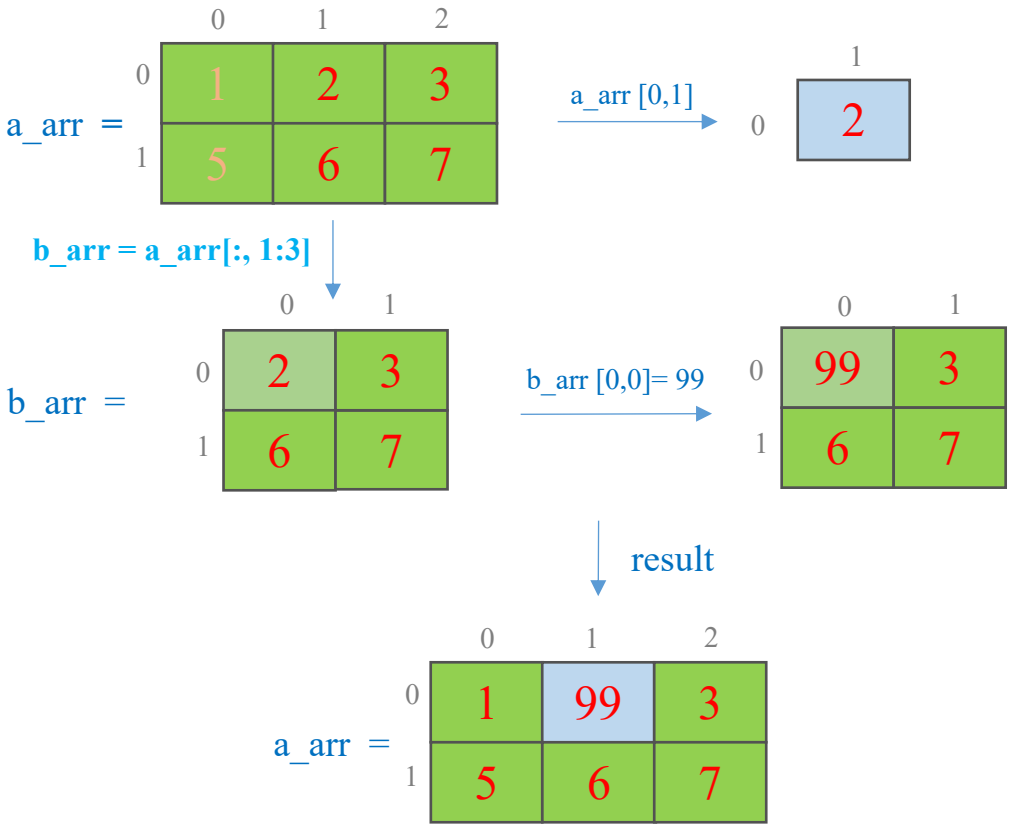

```python
# aivietnam.ai
import numpy as np

# Khởi tạo numpy array a_arr
a_arr = np.array([[1,2,3],
                  [5,6,7]])
print('a_arr \n', a_arr)

# Sử dụng slicing để tạo mảng b_arr
b_arr = a_arr[:, 1:3]
print('b_arr \n', b_arr)

print('before changing \n', a_arr[0, 1])
b_arr[0, 0] = 99
print('after changing \n', a_arr[0, 1])
```

AI VIET NAM
@aivietnam.edu.vn

Numpy Review

❖ Slicing

❖ Mutable




```python
# aivietnam.ai
import numpy as np

# Khởi tạo numpy array a_arr
a_arr = np.array([[1,2,3],
                  [5,6,7]])
print('a_arr \n', a_arr)

# Sử dụng slicing để tạo mảng b_arr
b_arr = a_arr[:, 1:3]
print('b_arr \n', b_arr)

print('before changing \n', a_arr[0, 1])
b_arr[0, 0] = 99
print('after changing \n', a_arr[0, 1])
```

```
a_arr
 [[1 2 3]
 [5 6 7]]
b_arr
 [[2 3]
 [6 7]]
before changing
 2
after changing
 99
```

AI VIET NAM
@aivietnam.edu.vn

Accessing Values in a Numpy Array

Indexing

data

	0	1
0	1	2
1	3	4
2	5	6

data[0, 1]

	0	1
0	1	2
1	3	4
2	5	6

data[1, 1]

	0	1
0	1	2
1	3	4
2	5	6

data[2, 0]

	0	1
0	1	2
1	3	4
2	5	6

Slicing

a_arr =

	0	1	2
0	1	2	3
1	5	6	7

b_arr = a_arr[:, 1:3]

b_arr =

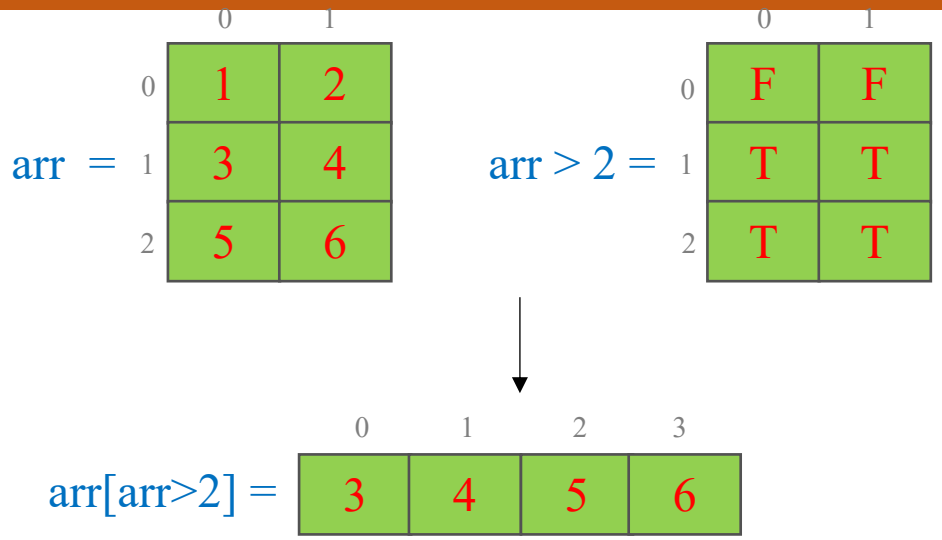
	0	1
0	2	3
1	6	7

```python
# aivietnam
import numpy as np

# Khởi tạo numpy array a_arr
a_arr = np.array([[1,2,3],
                  [5,6,7]])
print('a_arr \n', a_arr)

# Sử dụng slicing để tạo mảng b_arr
b_arr = a_arr[:, 1:3]
print('b_arr \n', b_arr)
```

```
a_arr
 [[1 2 3]
 [5 6 7]]
b_arr
 [[2 3]
 [6 7]]
```

AI VIET
@aivietnam

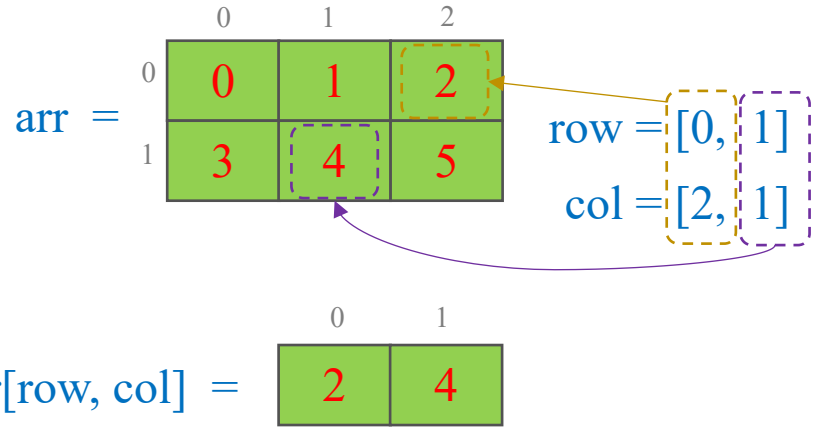
Masking




```python
# aivietnam
import numpy as np

arr = np.array([[1, 2],
                [3, 4],
                [5, 6]])

# tìm các phần tử lớn hơn 2
bool_idx = (arr > 2)

# get satisfying elements
result = arr[bool_idx]
print(result)
```

```
[3 4 5 6]
```

Fancy Indexing




```python
import numpy as np

arr = np.arange(6).reshape((2, 3))
print(arr)

row = np.array([0, 1])
col = np.array([2, 1])
print(arr[row, col])
```

```
[[0 1 2]
 [3 4 5]]
[2 4]
```

AI VIET NAM
@aivietnam.edu.vn

Mean

❖ Numpy review

```
1  # numpy review
2  import numpy as np
3
4  arr = np.ones((5,5))
5  print(arr)
6
7  roi = arr[1:4, 1:4]
8  roi = roi + 1
9  print(roi)
10
11 arr[1:4, 1:4] = roi
12 print(arr)
```

```
[[1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]]
```

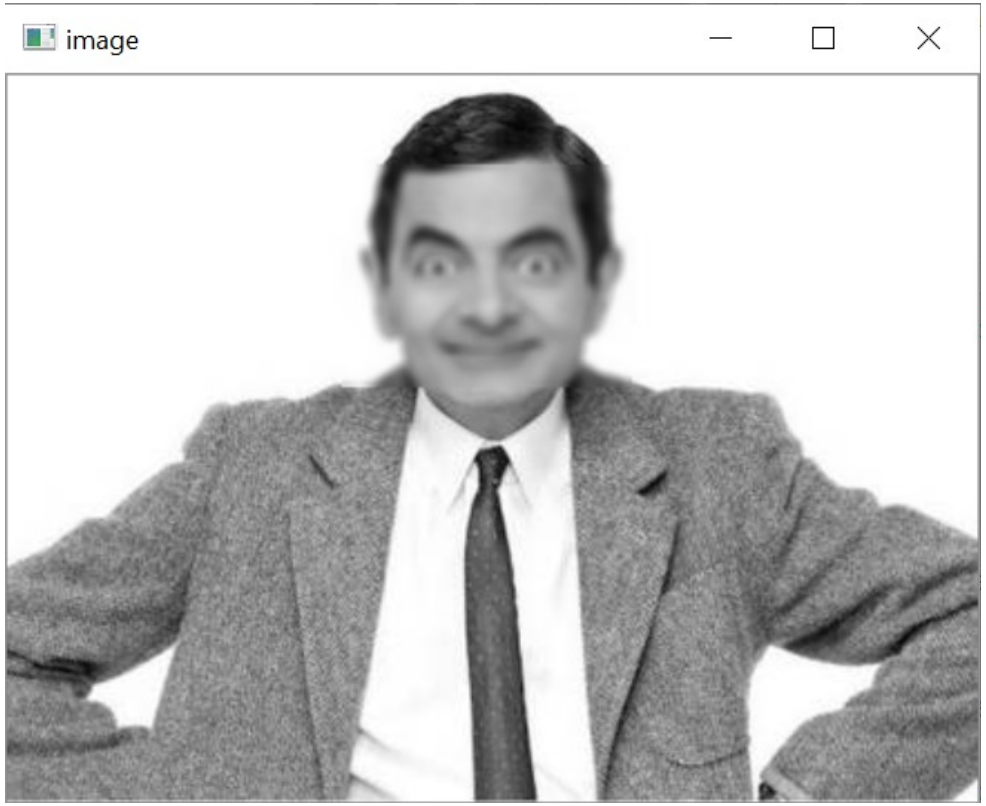
```
[[2. 2. 2.]
 [2. 2. 2.]
 [2. 2. 2.]]
```

```
[[1. 1. 1. 1. 1.]
 [1. 2. 2. 2. 1.]
 [1. 2. 2. 2. 1.]
 [1. 2. 2. 2. 1.]
 [1. 1. 1. 1. 1.]]
```

Mean

❖ Image Blurring

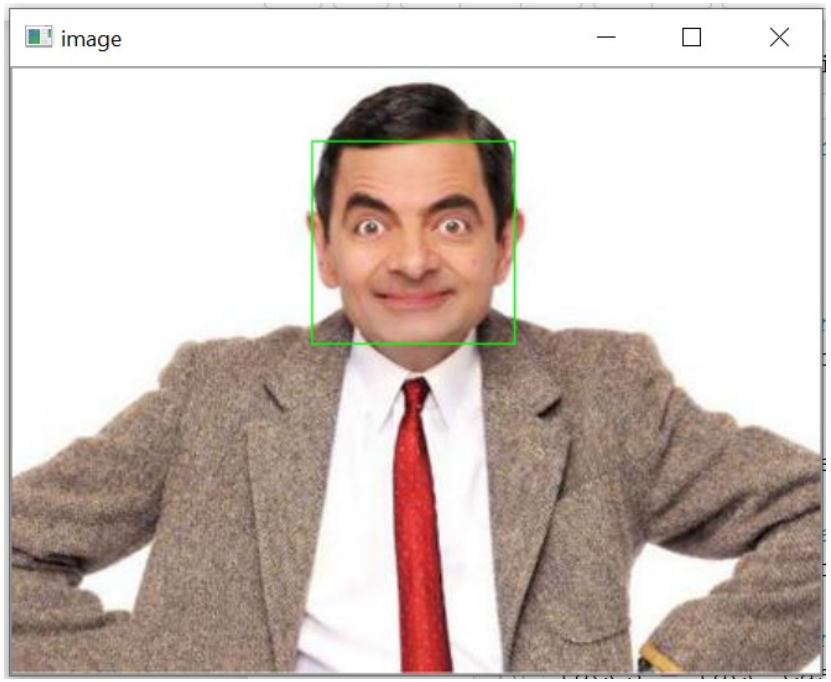
```python
# load image and blurring using mask-simple

import numpy as np
import cv2

# load image in grayscale mode
image = cv2.imread('mrbean.jpg', 0)

# create kernel
kernel = np.ones((5,5), np.float32) / 25.0

# Select ROI (top_y,top_x,height, width)
roi = image[40:140,150:280]

# compute mean for each pixel
roi = cv2.filter2D(roi, cv2.CV_8U, kernel)

image[40:140,150:280] = roi

# show images
cv2.imshow('roi', roi)
cv2.imshow('image', image)

# waiting for any keys pressed and close windows
cv2.waitKey(0)
cv2.destroyAllWindows()
```



AI VIET NAM
@aivietnam.edu.vn

Mean

❖ Image Blurring

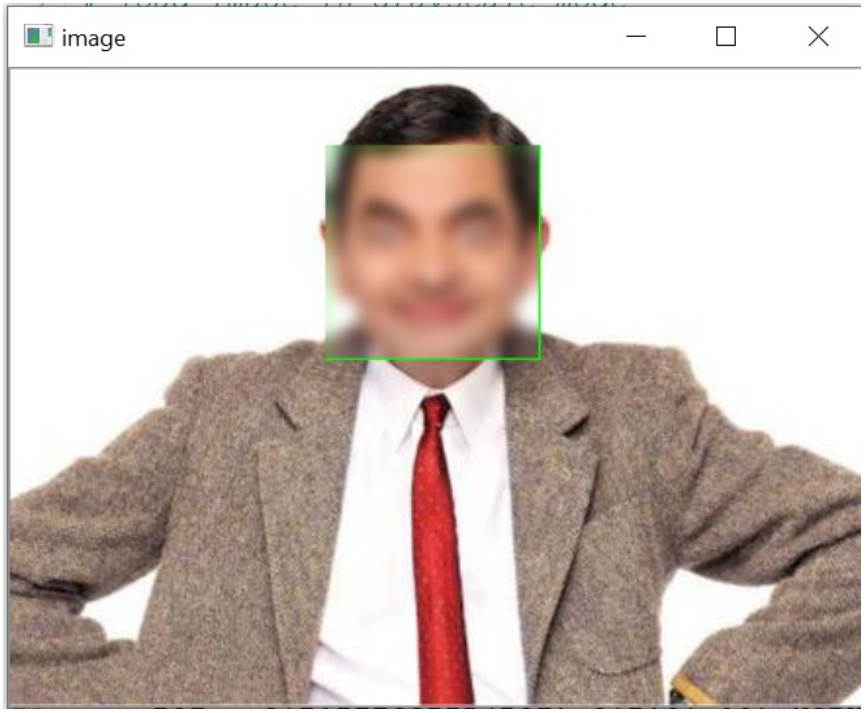



```python
# load image and blurring using face detection

import numpy as np
import cv2

# face detection setup
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# load image in grayscale mode
image = cv2.imread('mrbean.jpg', 1)

# Convert to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# face detection
faces = face_cascade.detectMultiScale(gray, 1.1, 4)

# Draw the rectangle around each face
for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 1)

# show images
cv2.imshow('image', image)

# waiting for any keys pressed and close windows
cv2.waitKey(0)
cv2.destroyAllWindows()
```

https://towardsdatascience.com/face-detection-in-2-minutes-using-opencv-python-90f89d7c0f81

Mean

❖ Image Blurring




```python
# load image and blurring using face detection

import numpy as np
import cv2

# face detection setup
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# load image in grayscale mode
image = cv2.imread('mrbean.jpg', 1)

# Convert to grayscale
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# face detection
faces = face_cascade.detectMultiScale(gray, 1.1, 4)

# create kernel
kernel = np.ones((7,7), np.float32) / 49.0

# Draw the rectangle around each face
for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x+w, y+h), (0, 255, 0), 1)
    roi = image[y:y+h,x:x+w]

    # compute mean for each pixel
    roi = cv2.filter2D(roi, cv2.CV_8U, kernel)
    roi = cv2.filter2D(roi, cv2.CV_8U, kernel)
    roi = cv2.filter2D(roi, cv2.CV_8U, kernel)

    # update
    image[y:y+h,x:x+w] = roi

# show images
cv2.imshow('image', image)

# waiting for any keys pressed and close windows
cv2.waitKey(0)
cv2.destroyAllWindows()
```


f10
delete
D
E
M
O

Outline

- Mean and Its Application
- Median and Its Application
- Variance and Its Application
- What is a PMF, PDF, and CDF?
- Histogram
- Histogram Equalization
- Summary

Median

❖ Definition

Data

$$X = \{X_1, \dots, X_N\}$$

Formula

Step 1: Sort X → S

Step 2

If N is odd, then $m = S_{\left(\frac{N+1}{2}\right)}$

If N is even, then $m = \left(S_{\left(\frac{N}{2}\right)} + S_{\left(\frac{N}{2}+1\right)}\right)/2$

Given the data

$$X = \{2, 8, 5, 4, 1\}$$

$$N = 5$$

Step 1

$$S = \{1, 2, 4, 5, 8\}$$
$$1 \quad 2 \quad 3 \quad 4 \quad 5$$

Step 2; $N = 5$

$$k = \frac{N+1}{2} = 3$$

$$m = S_k = 4$$

Median

❖ Definition

Data

$$X = \{X_1, \dots, X_N\}$$

Formula

Step 1: Sort X → S

Step 2

If N is odd, then $m = S_{\left(\frac{N+1}{2}\right)}$

If N is even, then $m = \left(S_{\left(\frac{N}{2}\right)} + S_{\left(\frac{N}{2}+1\right)}\right)/2$

Given the data

$$X = \{2, 8, 5, 4, 1, 8\}$$

$$N = 6$$

Step 1

$$S = \{1, 2, 4, 5, 8, 8\}$$
$$1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6$$

Step 2; $N = 6$

$$m = \frac{S_3 + S_4}{2}$$

$$= \frac{4 + 5}{2} = 4.5$$

Median

❖ Code

Data $X = \{X_1, \dots, X_N\}$

Formula

Step 1: Sort X → S

Step 2

If N is odd, then $m = S_{\left(\frac{N+1}{2}\right)}$

If N is even, then $m = \left(S_{\left(\frac{N}{2}\right)} + S_{\left(\frac{N}{2}+1\right)}\right)/2$

```
1.  def calculate_median(numbers):
2.      N = len(numbers)
3.      numbers.sort()
4.      if N%2 == 0:
5.          m1 = N/2
6.          m2 = (N/2) + 1
7.          m1 = int(m1)-1
8.          m2 = int(m2)-1
9.          median = (numbers[m1] + numbers[m2])/2
10.     else:
11.         m = (N+1)/2
12.         m = int(m)-1
13.         median = numbers[m]
14.     return median
```

Median

❖ Image Denoising

Input Image

(3x3) kernel

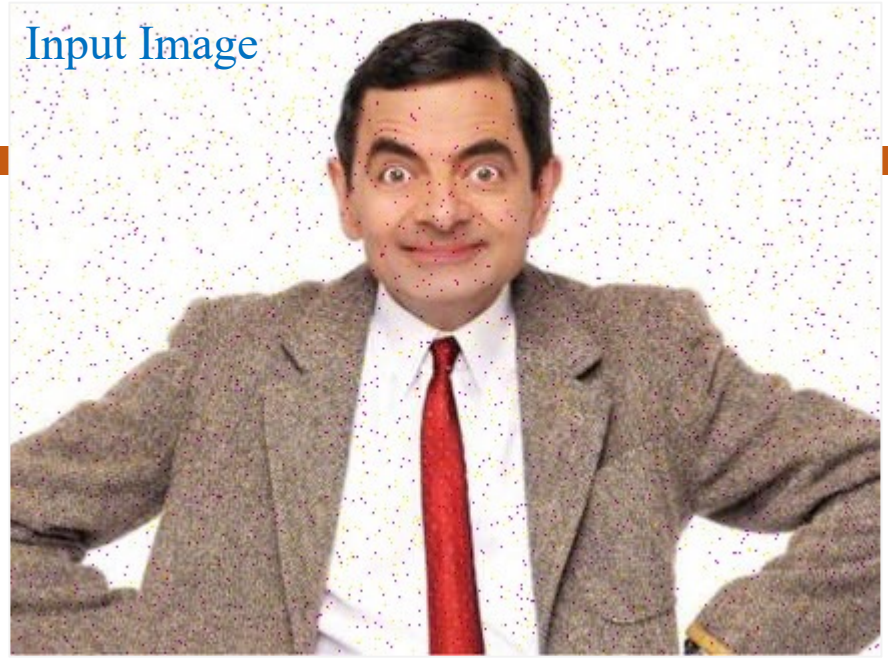
(5x5) kernel

Median

❖ Image Denoising

```python
import numpy as np
import cv2

img1 = cv2.imread('mrbean_noise.jpg')
img2 = cv2.medianBlur(img1, 3)

# show images
cv2.imshow('img1', img1)
cv2.imshow('img2', img2)

# waiting for any keys pressed and clo
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Input Image



(3x3) kernel

Mean and Median

❖ Comparison

Data

$$X = \{X_1, \dots, X_N\}$$

Formula

$$\mu = \frac{1}{N}\sum_{i-1}^{N} X_i$$

Formula

Step 1: Sort X → S

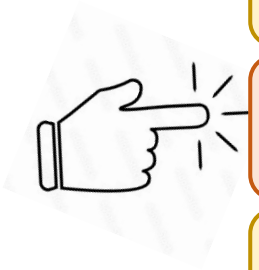
Step 2

If N is odd, then $m = S_{\left(\frac{N+1}{2}\right)}$

If N is even, then $m = \left(S_{\left(\frac{N}{2}\right)} + S_{\left(\frac{N}{2}+1\right)}\right)/2$


Q
&
A

Outline



- Mean and Its Application
- Median and Its Application
- Variance and Its Application
- What is a PMF, PDF, and CDF?
- Histogram
- Histogram Equalization
- Summary

Variance

❖ Definition

Formula:

mean	$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$
variance	$var(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$
Standard deviation	$\sigma = \sqrt{var(X)}$

Example: $X = \{5, 3\ 6, 7, 4\}$

	$\mu = \frac{1}{5}\sum_{i=1}^{n}(5+3+6+7+4) = \frac{25}{5} = 5$
	$var(X) = \frac{1}{5}[(5-5)^2 + (3-5)^2 + (6-5)^2 + (7-5)^2 + (4-5)^2]$ $= \frac{1}{5}(0+4+1+4+1)=2$
	$\sigma = \sqrt{var(X)} = 1.41$

Ignore the differences between population and sample, we will discuss them later

Variance

Formula

mean

$$E(X) = \sum_{i=1}^{N} X_i P_X(X_i)$$

variance

$$var(X) = E\left(\left(X - E(X)\right)^2\right)$$

$$= \sum_{i=1}^{N} \left(X_i - E(X)\right)^2 P_X(X_i)$$

Standard deviation

$$\sigma = \sqrt{var(X)}$$

Example: $X = \{5, 3\ 6, 7, 4\}$

$$E(X) = 5\times\frac{1}{5} + 3\times\frac{1}{5} + 6\times\frac{1}{5} + 7\times\frac{1}{5} + 4\times\frac{1}{5}$$

$$= 5$$

$$var(X) = \frac{1}{5}[(5-5)^2 + (3-5)^2 + (6-5)^2 + (7-5)^2 + (4-5)^2]$$

$$= \frac{1}{5}(0+4+1+4+1)=2$$

$$\sigma = \sqrt{var(X)} = 1.41$$

Variance

Formula

mean

$$E(X) = \sum_{i=1}^{N} X_i P_X(X_i)$$

variance

$$var(X) = E\left(\left(X - E(X)\right)^2\right)$$

$$= E(X^2) - \left(E(X)\right)^2$$

Standard deviation

$$\sigma = \sqrt{var(X)}$$

$$var(X) = E[(X - E(X))^2]$$

$$= \mathrm{E}[\mathrm{X}^2 - 2\mathrm{XE(X)} + \mathrm{E(X)}^2]$$

$$= \sum_{i=1}^{N} \left(X_i^2 - 2X_i E(X) + E(X)^2\right) P_X(X_i)$$

$$= \sum_{i=1}^{N} X_i^2 P_X(X_i) - \sum_{i=1}^{N} 2X_i E(X) P_X(X_i) + \sum_{i=1}^{N} E(X)^2 P_X(X_i)$$

$$= E(X^2) - 2E(X)\left[\sum_{i=1}^{N} X_i P_X(X_i)\right] + E(X)^2$$

$$= E(X^2) - \left(E(X)\right)^2$$

Variance

```python
import numpy as np

x = np.array([5, 3, 6, 7, 4])

var = (x - x.mean())**2
var = np.mean(var)
print(var)

sdv = np.sqrt(var)
print(sdv)
```

```
2.0
1.4142135623730951
```

mean $\mu = \frac{1}{n}\sum_{k=1}^{n} x_i$

variance $var(X) = \frac{1}{n}\sum_{k=1}^{n} (x_i - \mu)^2$

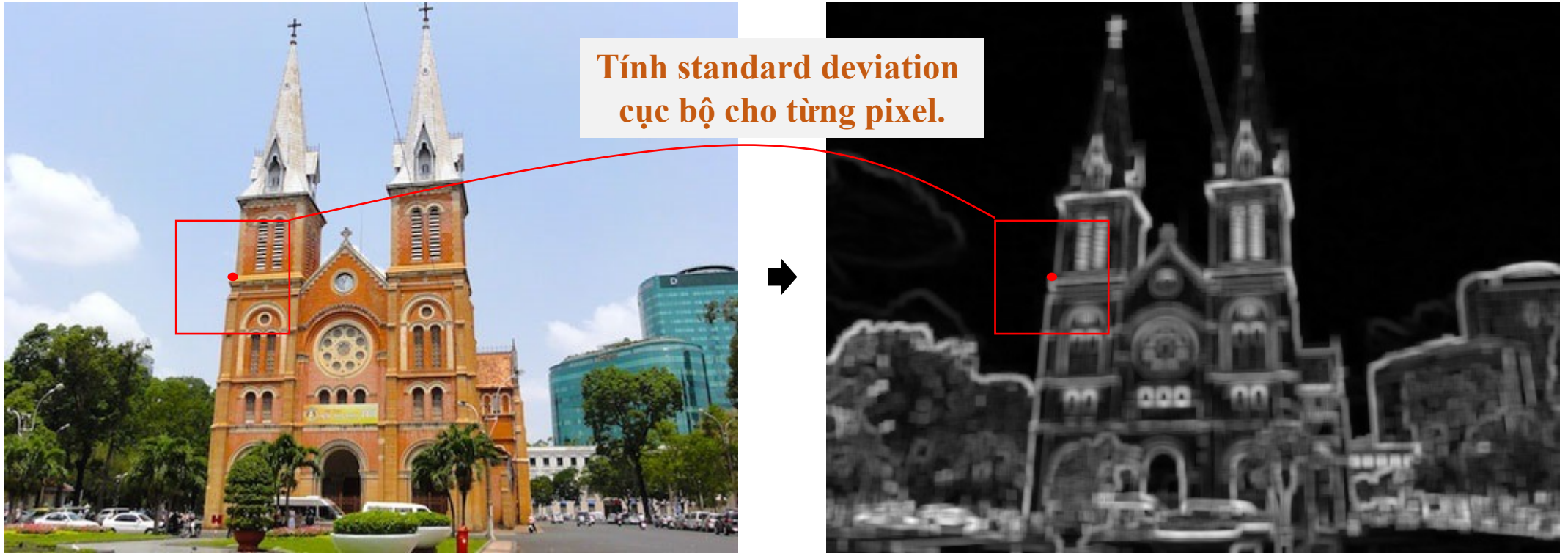
Standard deviation $\sigma = \sqrt{var(X)}$

```python
# variance
def calculate_mean(numbers):
    s = sum(numbers)
    N = len(numbers)
    mean = s/N
    return mean

def caculate_variance(numbers):
    mean = calculate_mean(numbers)

    diff = []
    for num in numbers:
        diff.append(num-mean)

    squared_diff = []
    for d in diff:
        squared_diff.append(d**2)

    sum_squared_diff = sum(squared_diff)
    variance = sum_squared_diff/len(numbers)

    return variance
```

AI VIET NAM
@aivietnam.edu.vn

Variance

Ứng dụng tính chất của variance (~standard deviation) để tìm texture cho một hình



Image

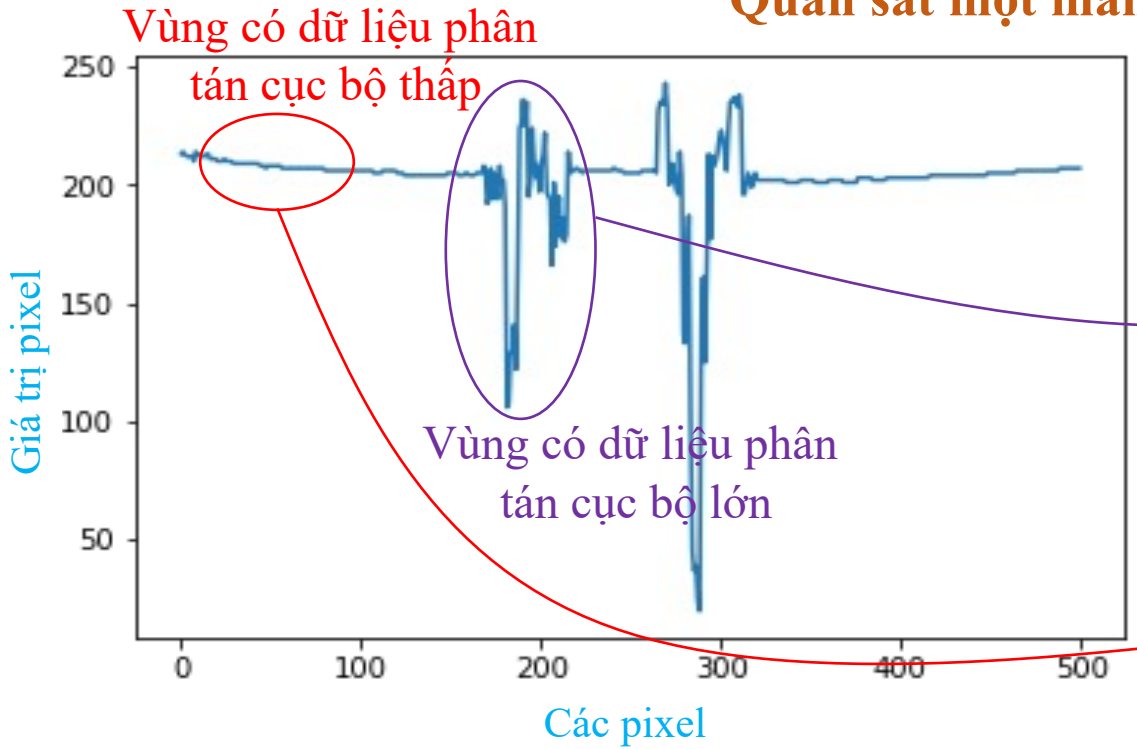
Image Texture

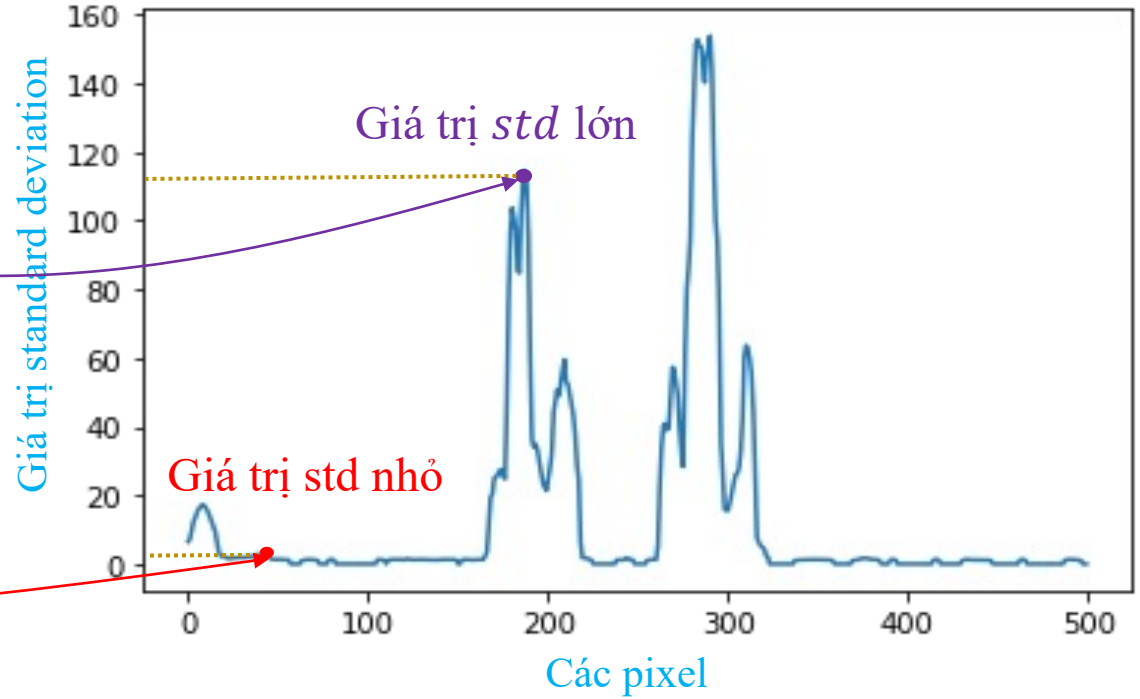
Tính standard deviation cục bộ cho từng pixel.

Ảnh gốc

Ảnh thông tin texture

Quan sát một mảng các giá trị pixel.



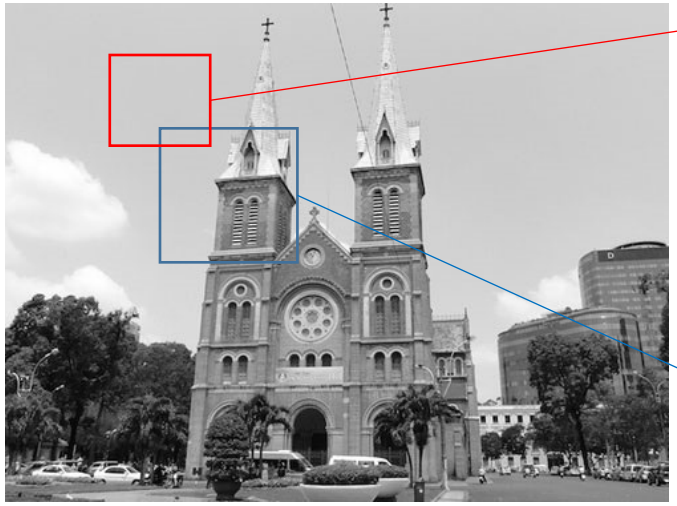


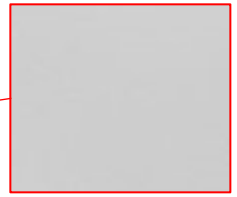
Variance

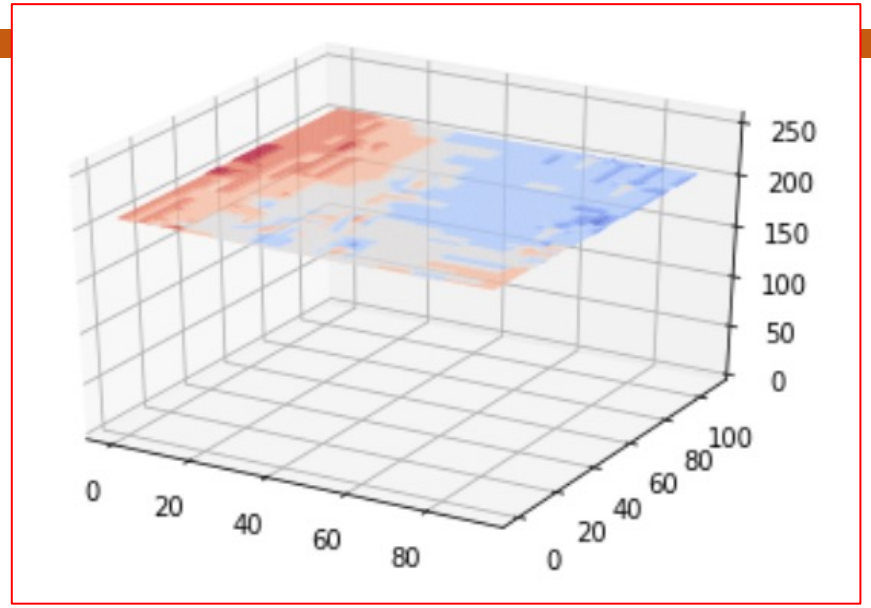
❖ Implementation

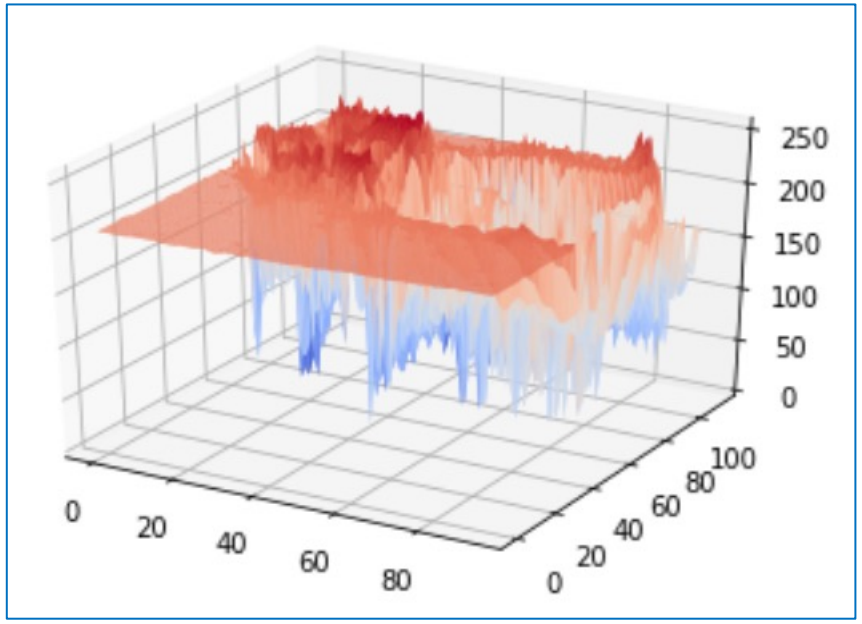
Variance

❖ Implementation





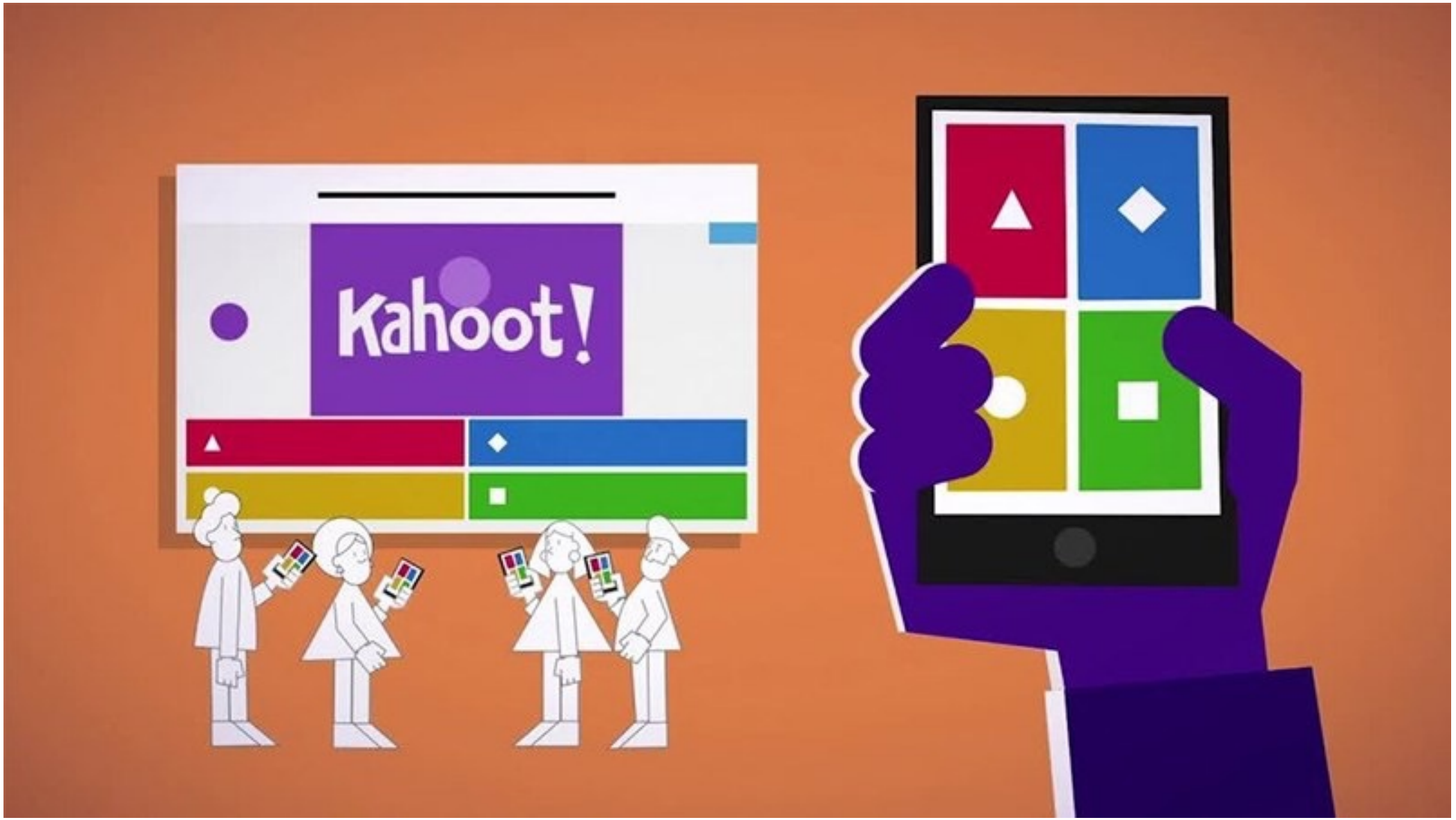




Variance

❖ Implementation

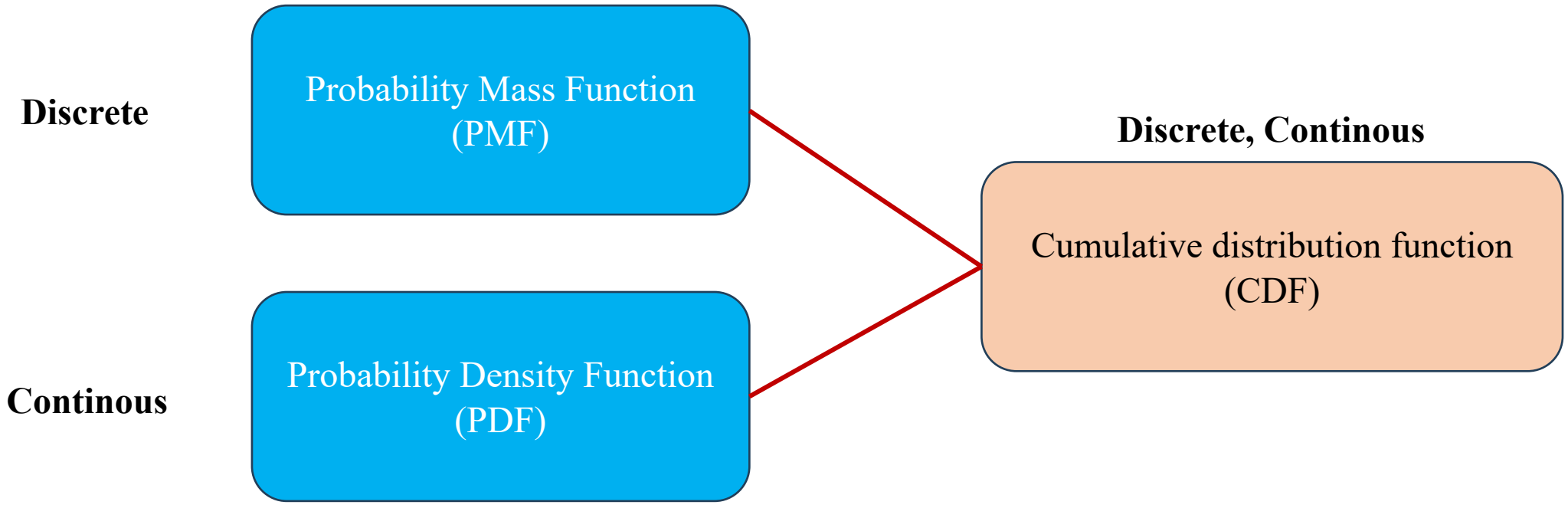
```python
import numpy as np
import cv2
import math
from scipy.ndimage.filters import generic_filter

img  = cv2.imread('img.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imwrite('edge_s1.jpg', gray)

x = gray.astype('float')
x_filt = generic_filter(x, np.std, size=7)
cv2.imwrite('edge_s2.jpg', x_filt)

x_filt[x_filt < 20] = 0
cv2.imwrite('edge_s3.jpg', x_filt)

maxv = np.max(x_filt)
print(maxv)

x_filt = x_filt*2.5
cv2.imwrite('edge_s4.jpg', x_filt)
```


Kahoot!

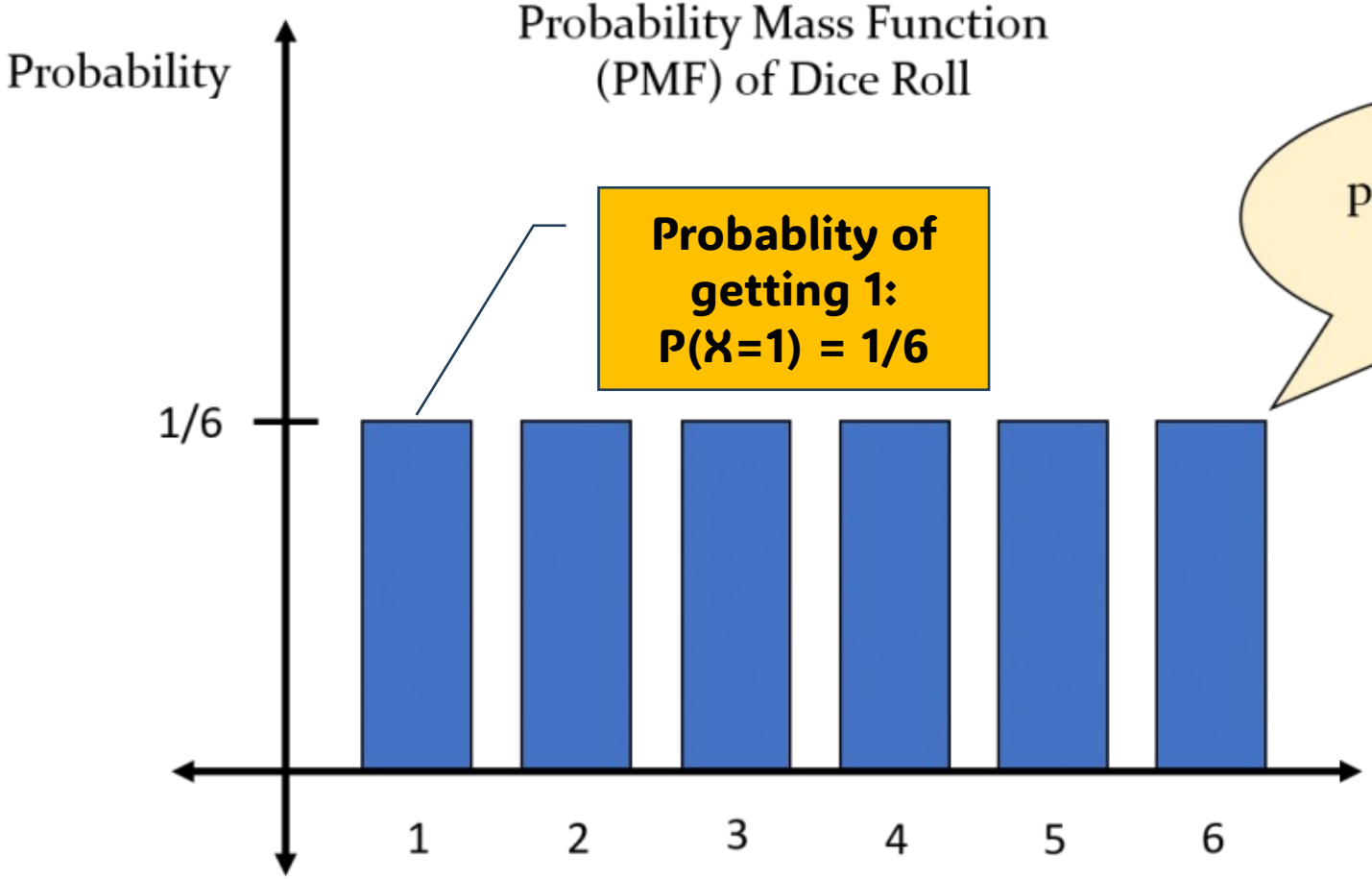
Outline

- Mean and Its Application
- Median and Its Application
- Variance and Its Application
- What is a PMF, PDF, and CDF?
- Histogram
- Histogram Equalization
- Summary

Probability Distribution-Phân phối xác suất



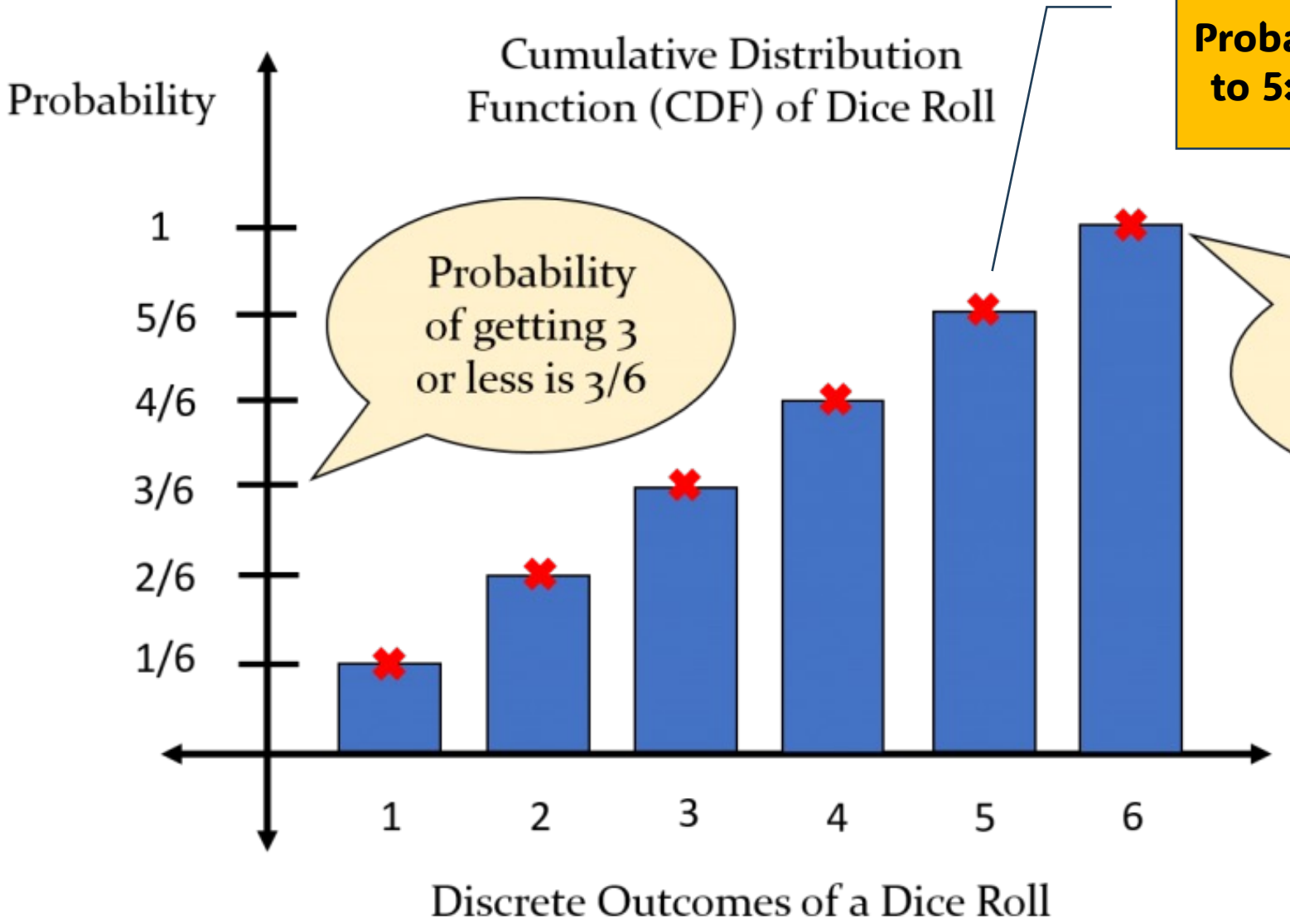
PMF - Discrete



P(X) measures the probability of a given single value x occuring

CDF - Discrete

Probablity of getting less than or equal to 5: P (X <=5) = (1+1+1+1+1)/6 = 5/6



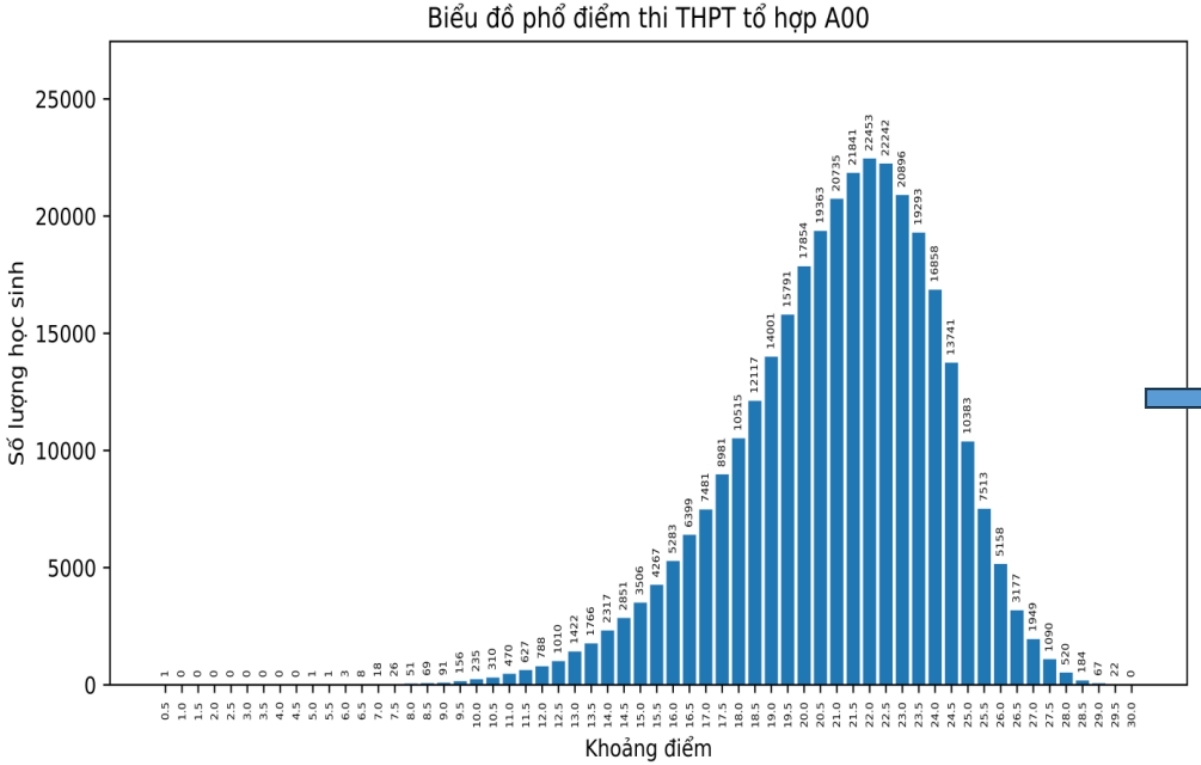
$$F_X(x) = P(X \leq x)$$

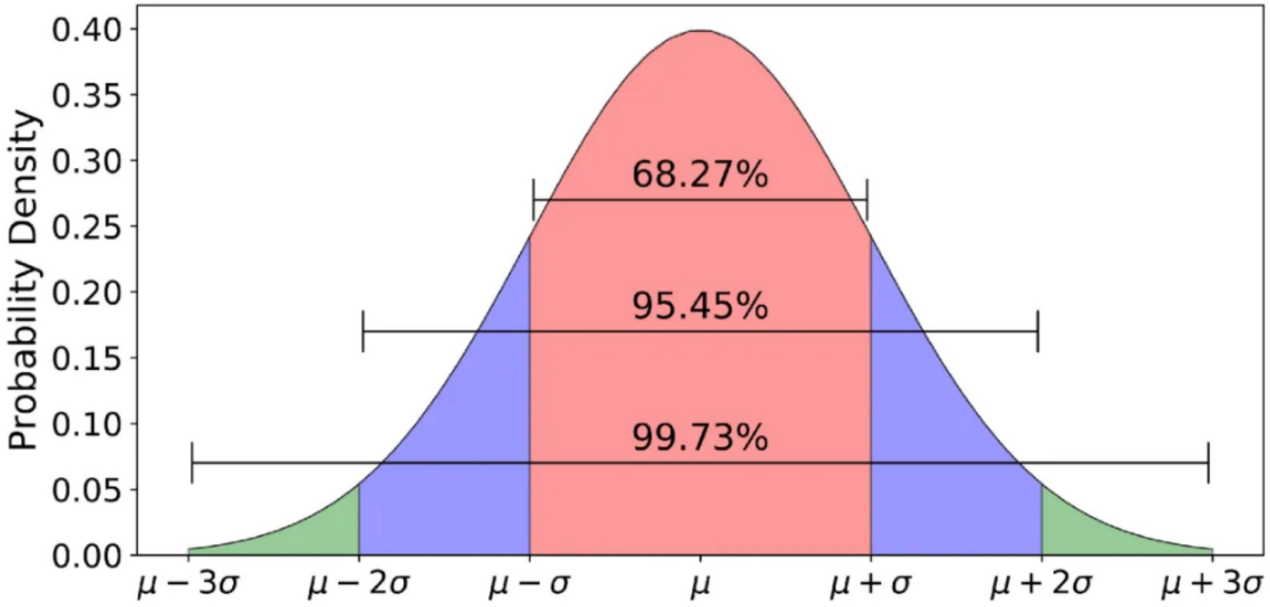
$F_X(x)$ = function of X

X = real value variable

P = probability that X will have a value less than or equal to x

PDF - Continous





Gaussian Distribution

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

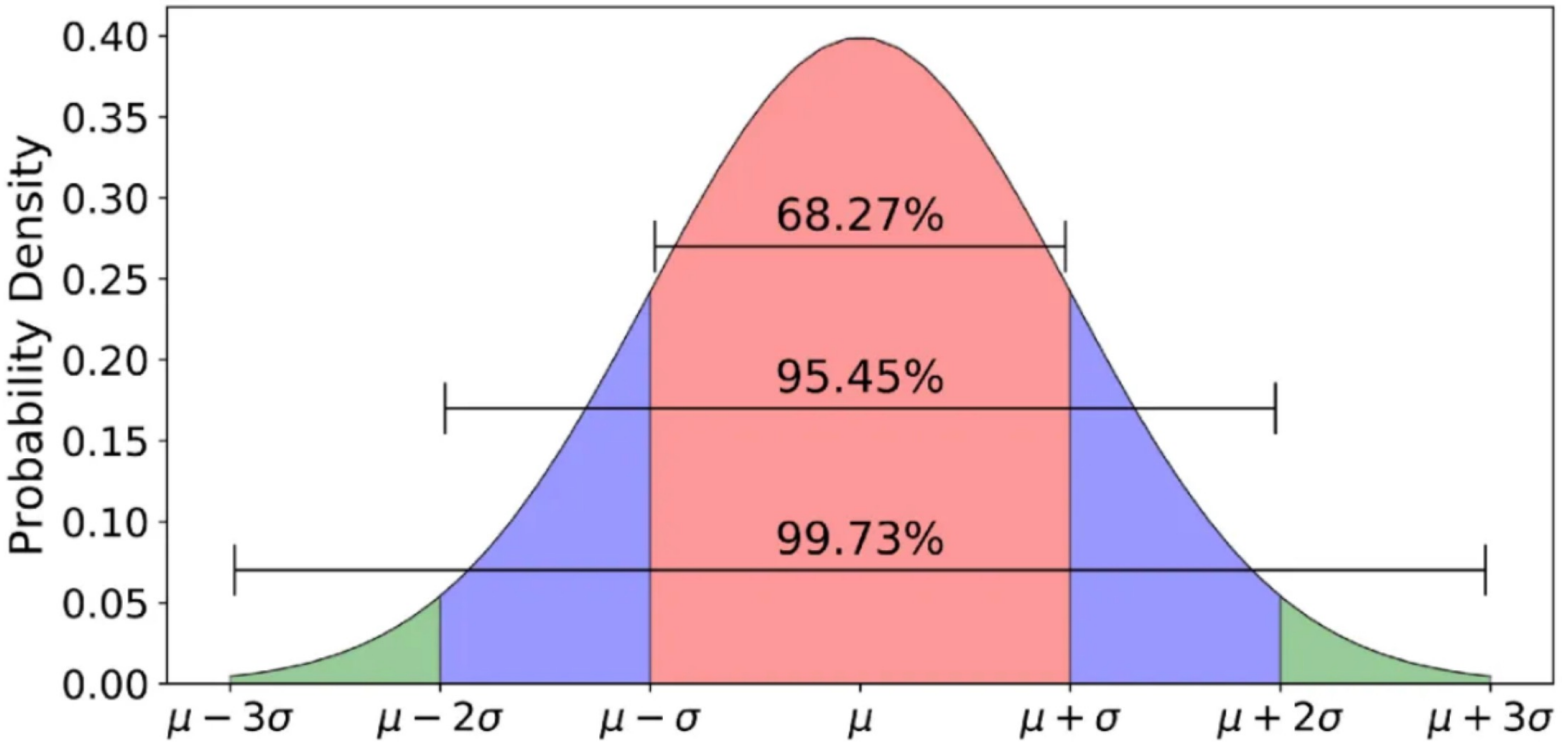
$f(x)$ = probability density function

σ = standard deviation

μ = mean

Phổ điểm các tổ hợp xét tuyển đại học năm 2023, nguồn https://laodong.vn/

PDF - Continous



$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

$f(x)$ = probability density function

σ = standard deviation

μ = mean

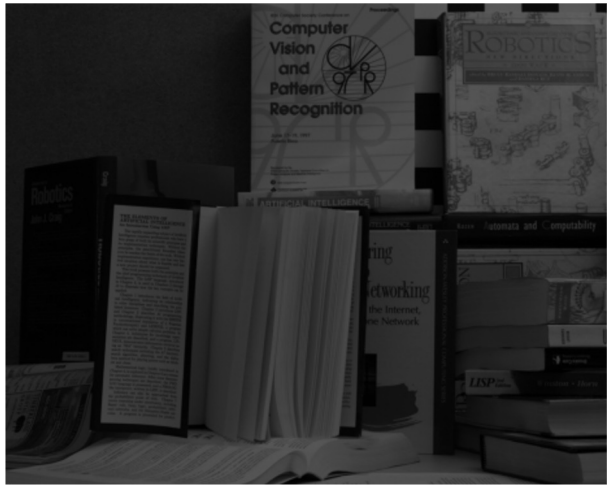
The empirical rule, also known as the 68–95–99.7 rule, is a guideline that applies to data following a normal distribution

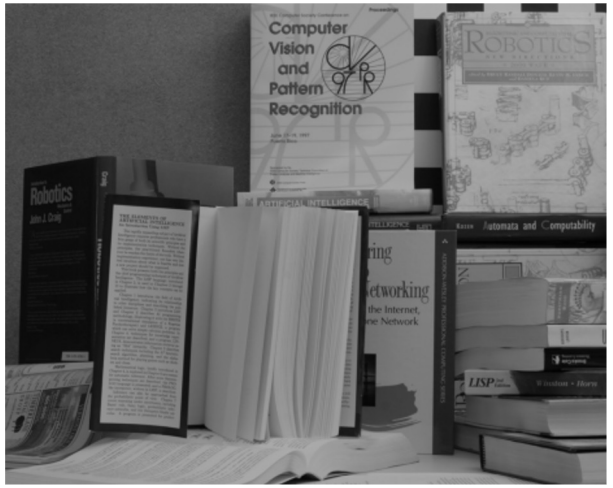
Outline

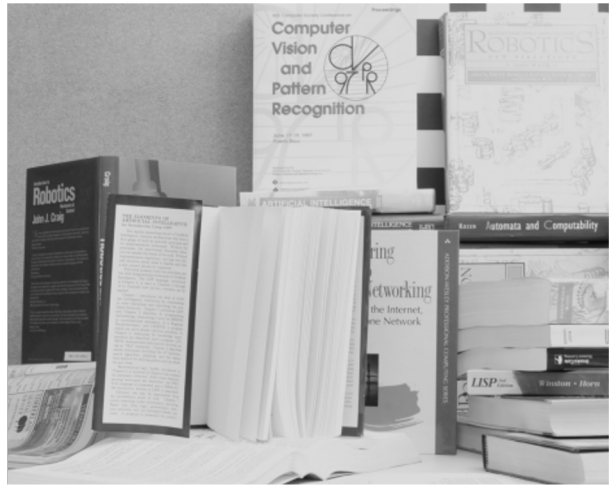
- Mean and Its Application
- Median and Its Application
- Variance and Its Application
- What is a PMF, PDF, and CDF?
- Histogram
- Histogram Equalization
- Summary

Histogram cho ảnh grayscale

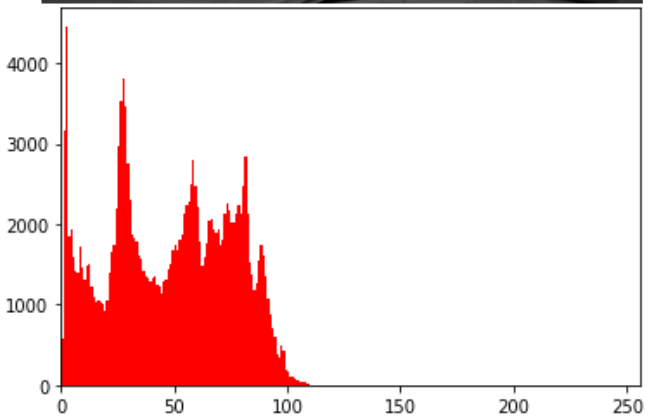
Ảnh





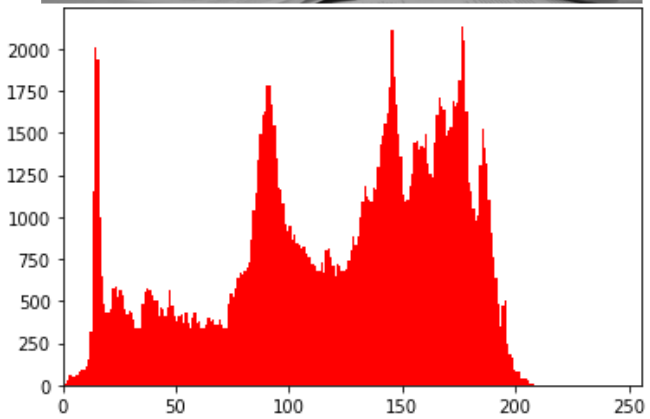


Histogram



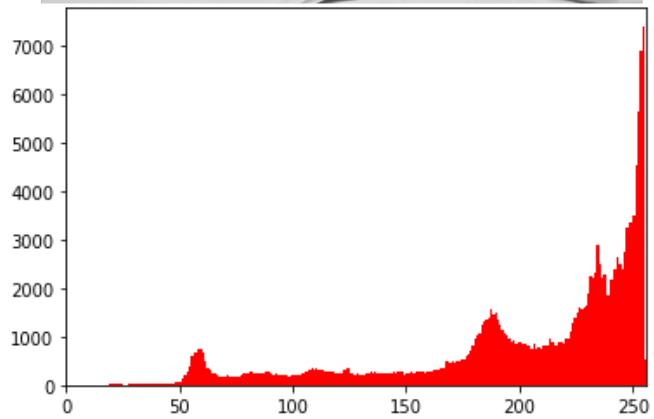
256 bin

Frequency Histogram



256 bin

Frequency Histogram

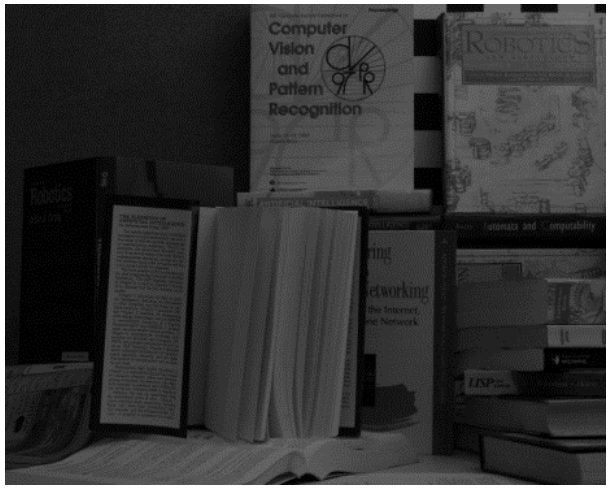


256 bin

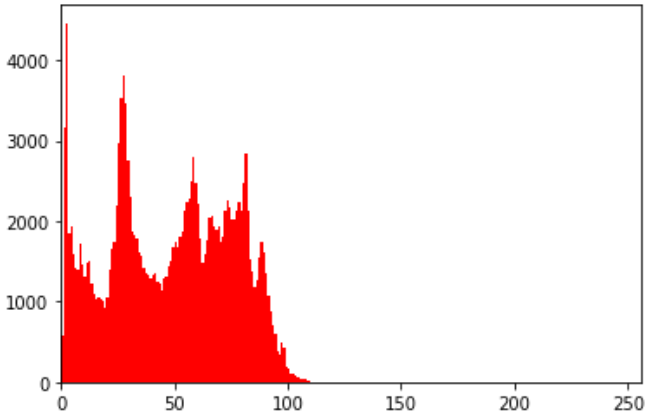
Frequency Histogram

AI VIET NAM
@aivietnam.edu.vn

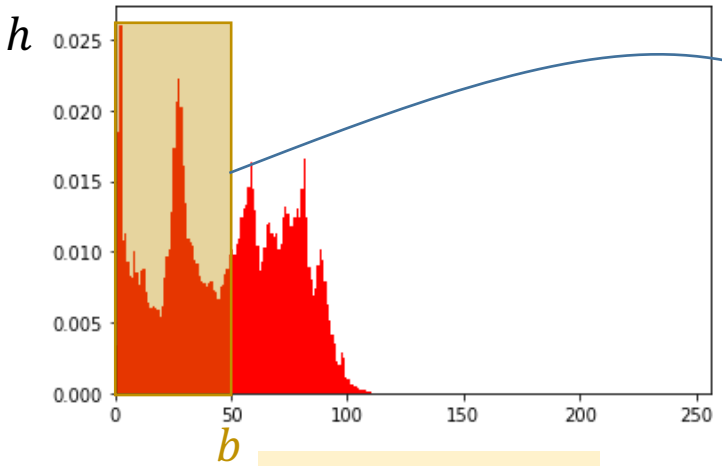
Histogram cho ảnh grayscale



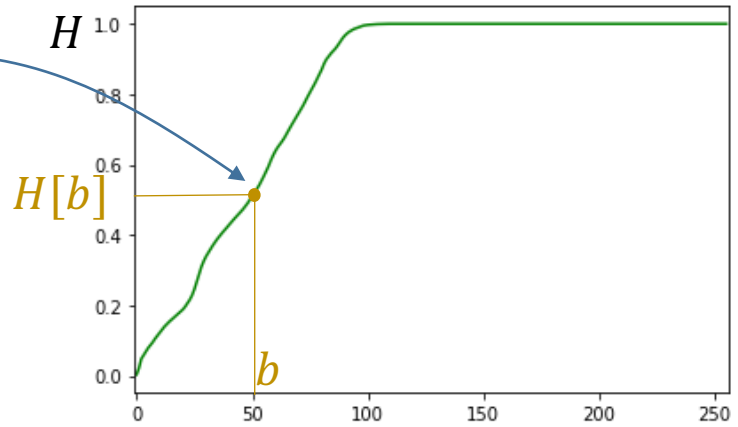
Ảnh



Histogram



Density Histogram



Cumulative Histogram

Gọi N là tổng số pixel của ảnh và n_b là số lượng pixel ở bin thứ b.

Giá trị density histogram ở bin thứ b được tính như sau:

$$h[b] = \frac{n_b}{N}$$

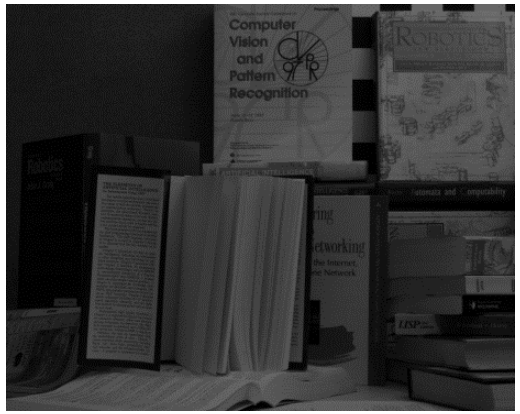
Giá trị cumulative histogram ở bin thứ b được tính như sau:

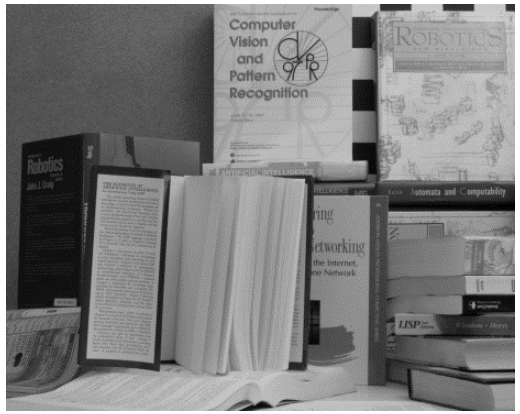
$$H[b] = \sum_{k=0}^{b} h[k]$$

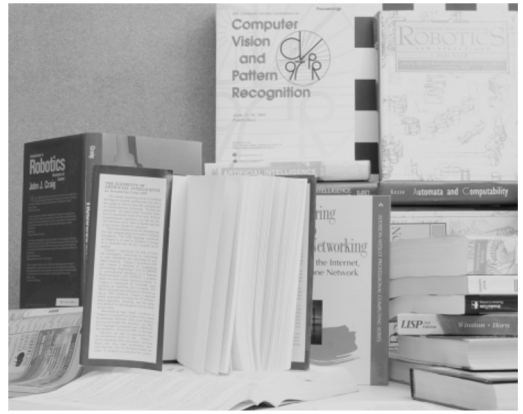
Cộng dồn giá trị $h[k]$ từ 0 đến b

Histogram cho ảnh grayscale

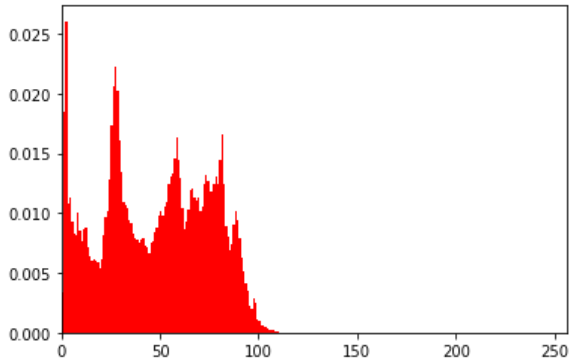
Ảnh

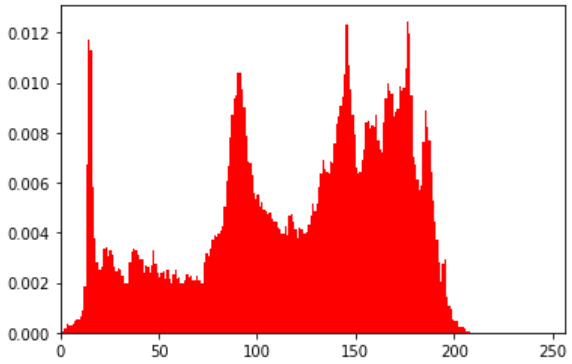


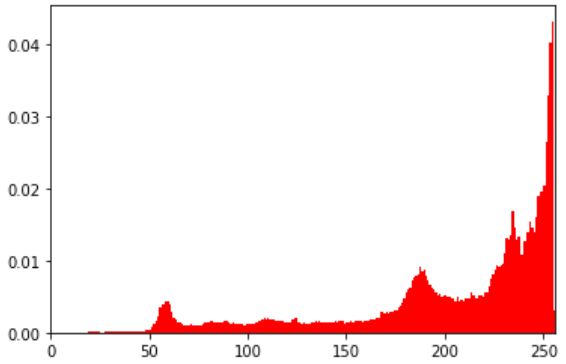




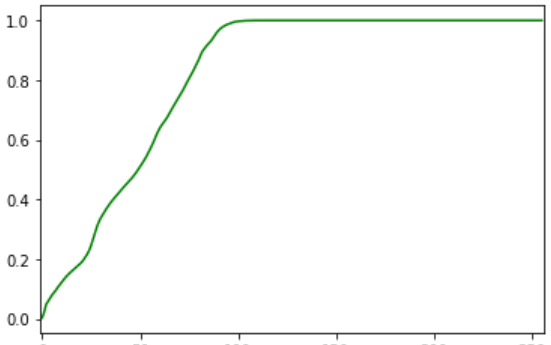
Density Histogram

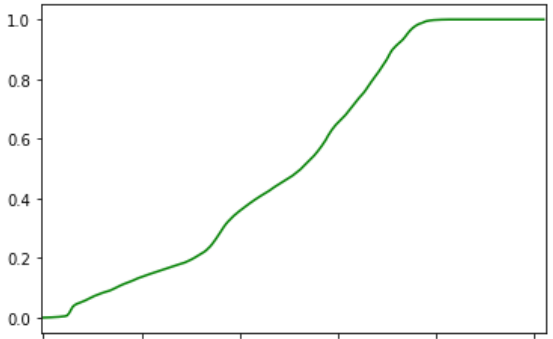


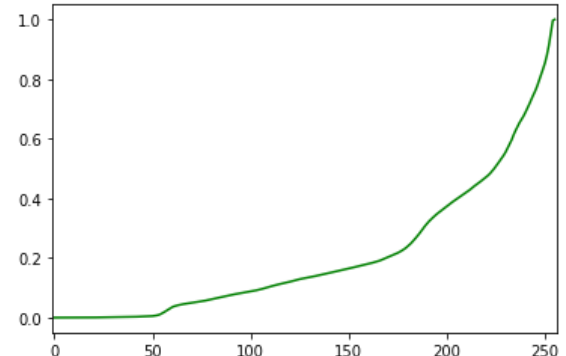




Cumulative Histogram







Outline

- Mean and Its Application
- Median and Its Application
- Variance and Its Application
- What is a PMF, PDF, and CDF?
- Histogram
- Histogram Equalization
- Summary

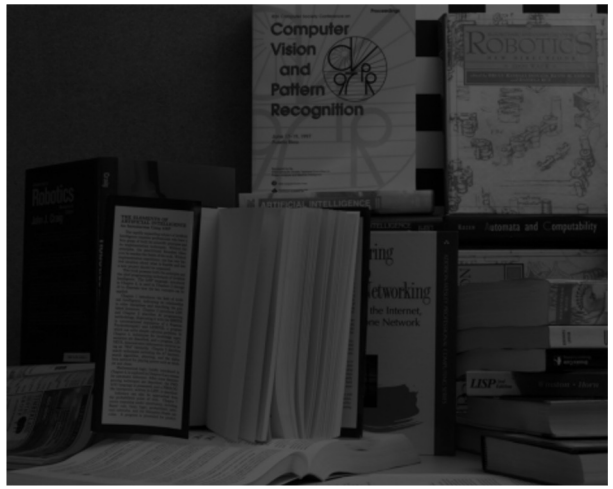
Histogram equalization

Được dùng để tăng độ tương phản của ảnh

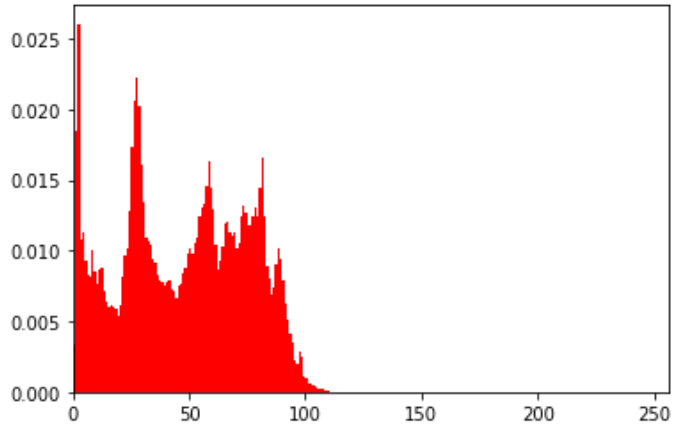
Idea: Kéo phân bố của density histogram sao cho xấp xỉ với uniform distribution.

$$\text{Floor} = \left\lfloor \frac{n}{m} \right\rfloor \times m,$$

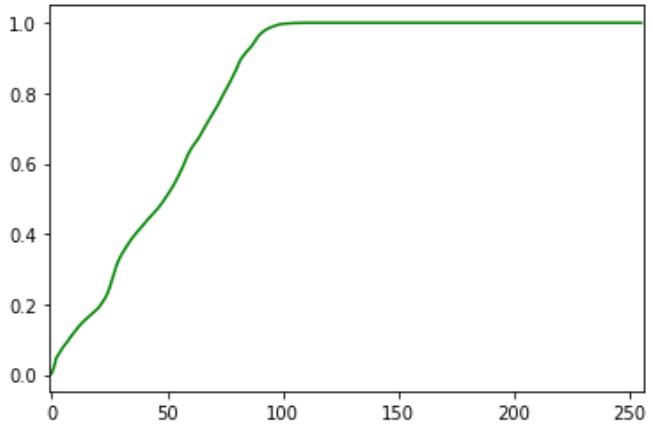
$$\text{Ceiling} = \left\lceil \frac{n}{m} \right\rceil \times m,$$



Ảnh



Density Histogram



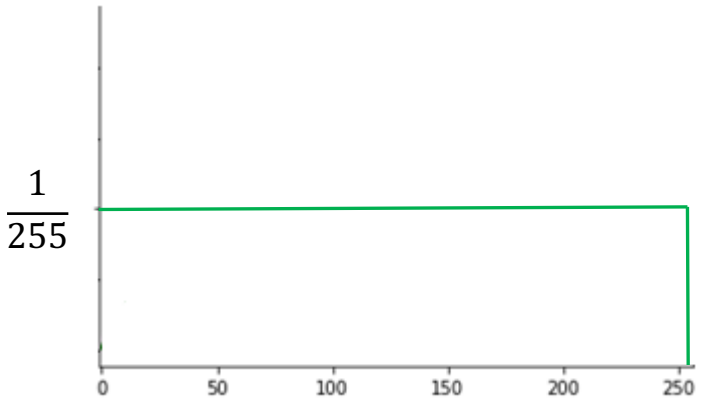
Cumulative Histogram

Công thức

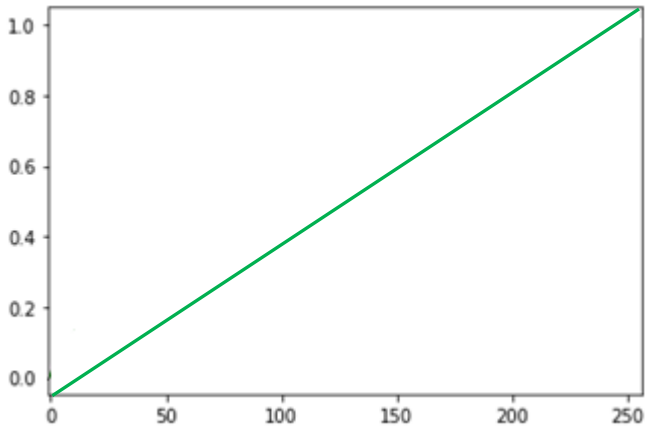
$$b_{new} = \lfloor (L-1) * H[b] + 0.5 \rfloor$$

Các bin (giá trị pixel) được thay đổi theo $H[b]$

L *is the number of gray levels*



Uniform density function trong khoảng [0, 255]



Cumulative density function cho uniform distribution

HE Example: Step by Step

Input image

Histogram Equalization

Output image

HE Example: Step by Step

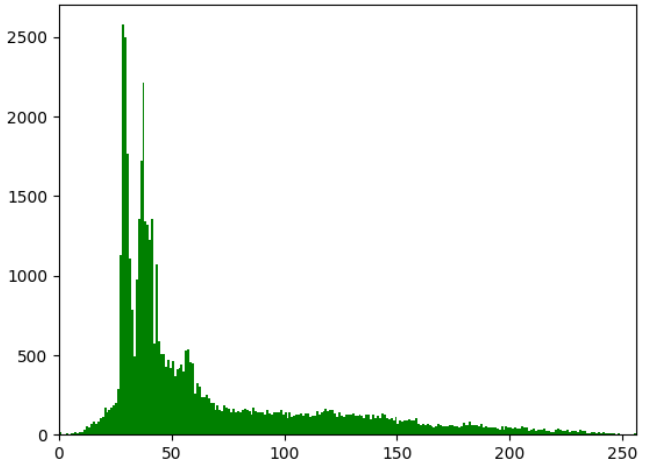
Find PMF

Input image

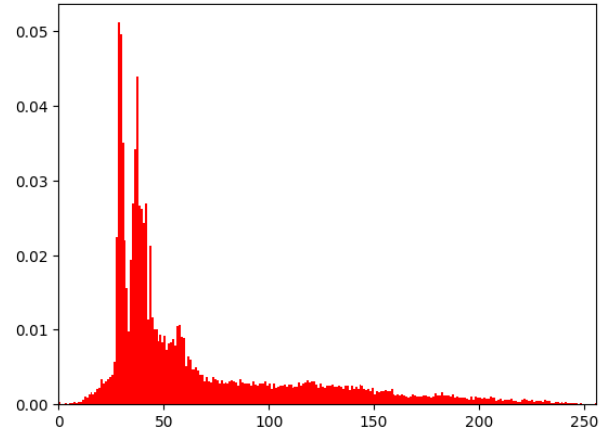
1	2	7	5	6
7	2	3	4	5
0	1	5	7	3
1	2	5	6	7
6	1	0	3	4

Simple Image

Frequency Histogram



Density Histogram



PMF

Intensity	Frequency	Probablity
0	2	2/25
1	4	4/25
2	3	3/25
3	3	3/25
4	2	2/25
5	4	4/25
6	3	3/25
7	4	4/25

HE Example: Step by Step

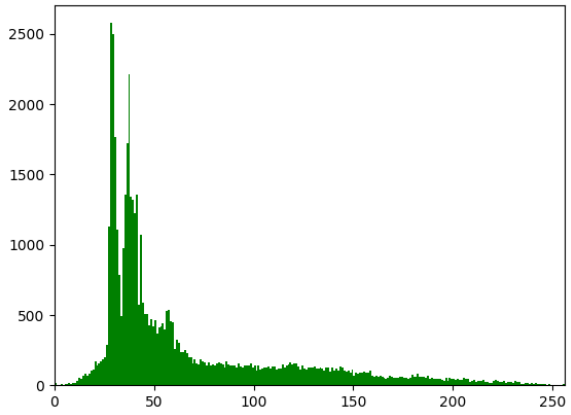
Find CDF

Gray Level Value	Frequency	PMF	CDF
0	2	0,08	0,08
1	4	0,16	0,24
2	3	0,12	0,36
3	3	0,12	0,48
4	2	0,08	0,56
5	4	0,16	0,72
6	3	0,12	0,84
7	4	0,16	1

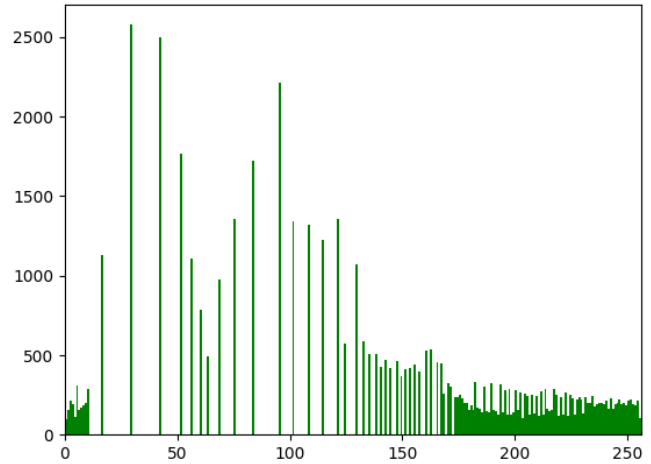
Gray Level Value	CDF	CDF * (Levels-1)	New Gray Level Value
0	0,08	1,11	1.00
1	0,24	2,23	2.0
2	0,36	3,07	3.0
3	0,48	3,91	3.0
4	0,56	4,47	3.0
5	0,72	5,59	5.0
6	0,84	6,43	6.0
7	1	7,55	7.0

Update new value

HE Example: Step by Step

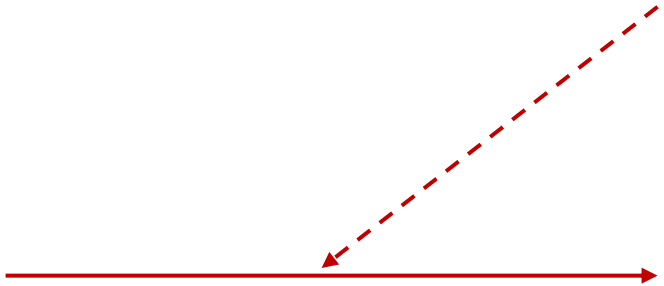


Gray Level Value	CDF	CDF * (Levels-1)	New Gray Level Value
0	0,08	1,64	1.00
1	0,24	2,92	2.0
2	0,36	3,88	3.0
3	0,48	4,84	4.0
4	0,56	5,48	5.0
5	0,72	6,76	6.0
6	0,84	7,72	7.0
7	1	7.5	7.0



1	2	...	...	...
...	2	...	...	...
...	1	...	...	...
1	2	...	...	...
...	1	...	...	...

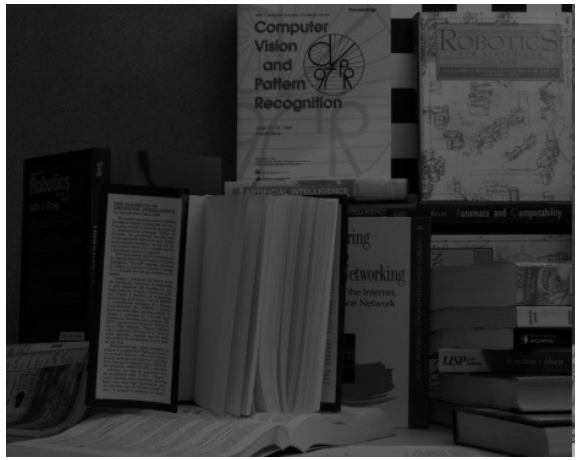
Original Image



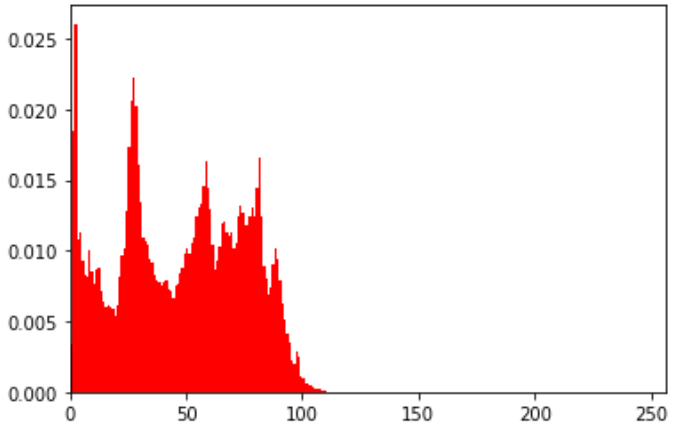
2	3	...	...	...
	3	...	...	...
	2	...	...	...
2	3	...	...	...
	2	...	...	...

New Image

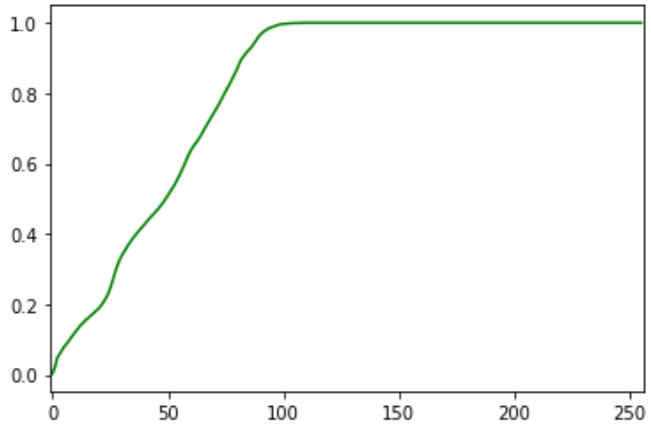
Histogram equalization để tăng độ tương phản



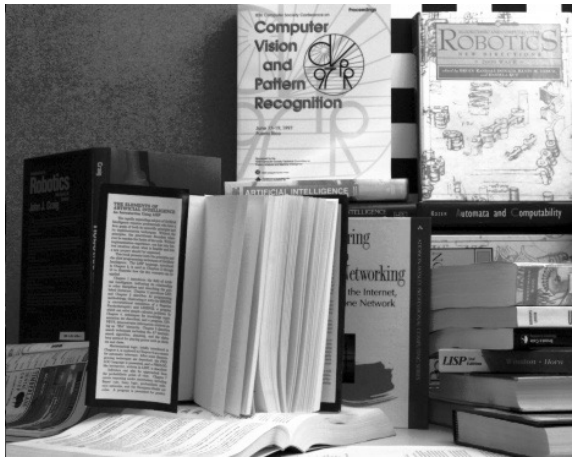
Ảnh gốc



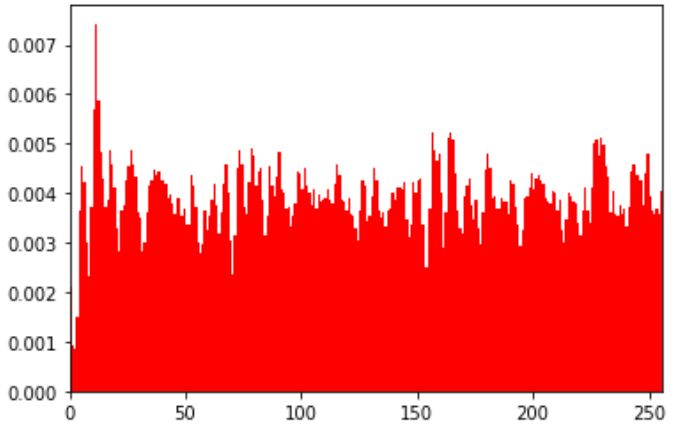
Density Histogram



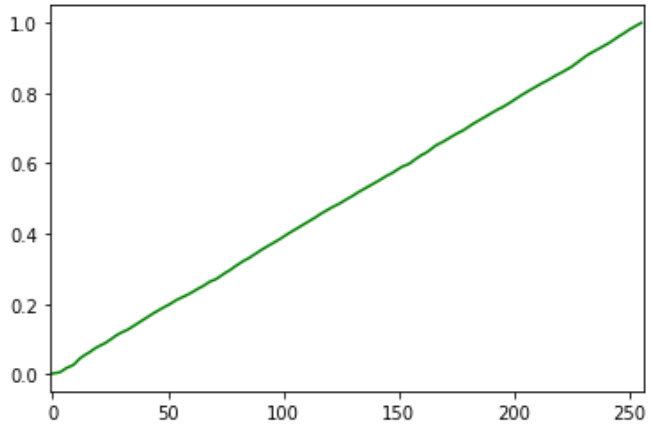
Cumulative Histogram



Ảnh kết quả



Density Histogram



Cumulative Histogram

Histogram equalization để tăng độ tương phản

Cho ảnh grayscale

Ảnh gốc | Ảnh kết quả

Cho ảnh màu

Ảnh gốc | Ảnh kết quả

Expected Values

	Love Math	Not Love Math	Total
# of students	60	40	100
Probability	0.6	0.4	1.0
Outcome	1$	-1.5$	

Tôi sẽ trả cho bạn 10$ nếu sinh viên kế tiếp tham gia AIO2025 thích Toán, Nhưng nếu ko phải bạn phải thua tôi 1$?

What do you think about this game, are you willing to play?

$$E(\mathrm{X}) = (0.6 * 1) + (0.4 * -1.5) = 0\ \$ = \sum \mathrm{x} * \mathrm{P}(\mathrm{X} = \mathrm{x})$$