Texture Analysis

Chien-Chang Chen
Tamkang University
Department of Computer Science and Information Engineering

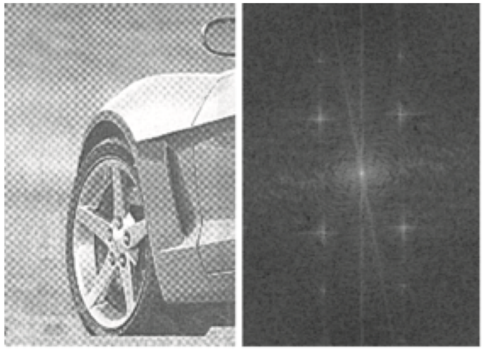
Summarized from Digital Image Processing, 4th ed. - Gonzalez,Woods (Pearson;Global Edition;2017)

Texture Definition

- region description
  - to quantify its texture content
- Texture definition
  - no formal definition
  - properties such as smoothness, coarseness, and regularity
- statistical and spectral approaches for describing the texture region

Texture Definition

- Statistical approaches
  - characterizations of textures as smooth, coarse, grainy, …
- Spectral techniques
  - properties of the Fourier spectrum
  - detect global periodicity in an image
  - identify high-energy, narrow peaks in its spectrum



Texture Images

a b c

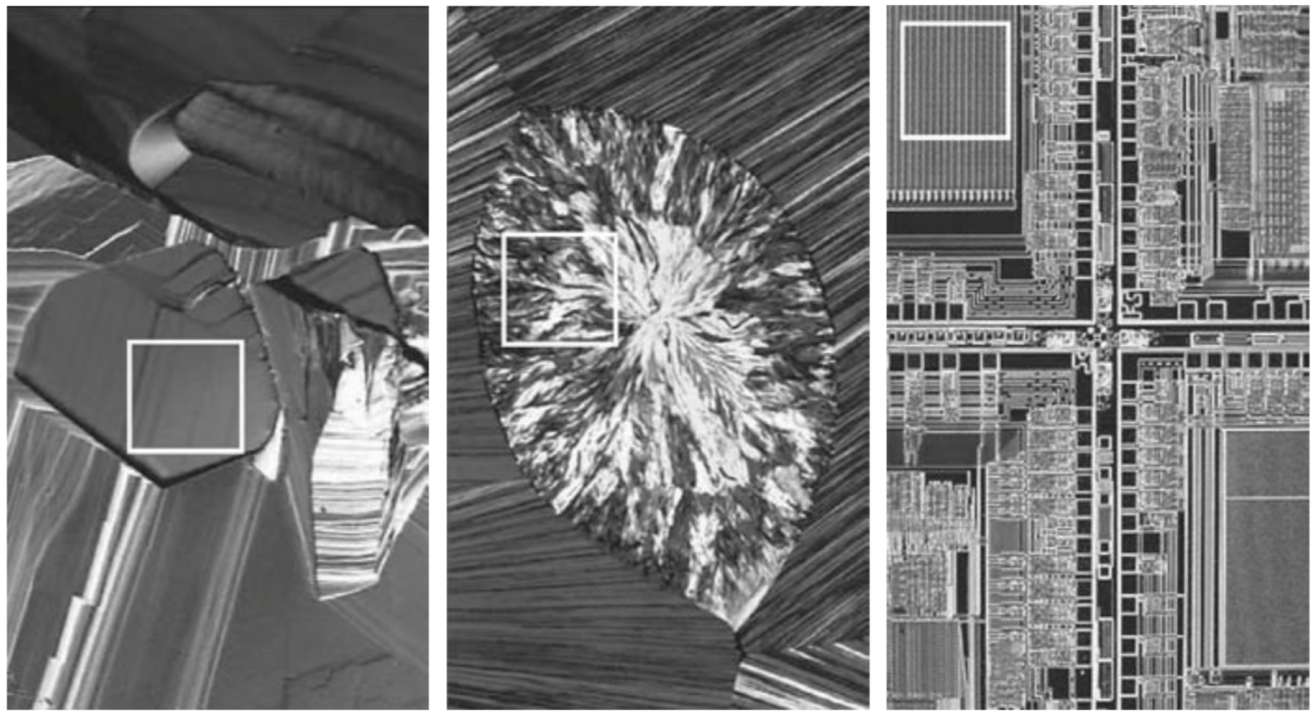


FIGURE 11.29 The white squares mark, from left to right, smooth, coarse, and regular textures. These are optical microscope images of a superconductor, human cholesterol, and a microprocessor. (Courtesy of Dr. Michael W. Davidson, Florida State University.)

Statistical Approaches

Statistical Approaches

- simplest approach
- statistical moments of the intensity histogram of an image or region
- z : intensity,
- $p(z_i)$, $i = 0, 1, 2, \ldots, L-1$, the corresponding normalized histogram
- L is the number of distinct intensity levels

- the *n*th moment of z about the mean

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i)$$

- m is the mean value of z
  - the average intensity of the image or region

$$m = \sum_{i=0}^{L-1} z_i p(z_i)$$

- $\mu_0(z) = 1$
- $\mu_1(z) = 0$

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - m)^n p(z_i)$$

- For example,
  - $z_0=3$, $z_1=5$, $z_2=6$, $z_3=2$

$$m = \sum_{i=0}^{L-1} z_i p(z_i)$$

$$m = \sum_{i=0}^{L-1} i \cdot p(z_i) = 0 \cdot \frac{3}{16} + 1 \cdot \frac{5}{16} + 2 \cdot \frac{6}{16} + 3 \cdot \frac{2}{16} = \frac{23}{16} = 1.4375$$

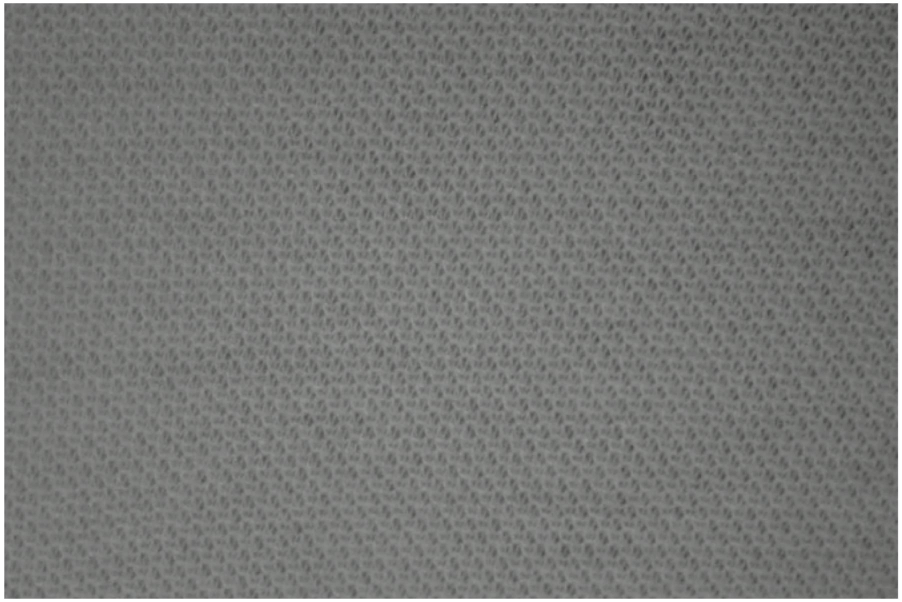
$$\mu_0(z) = \sum_{i=0}^{L-1} (z_i - m)^0 p(z_i) = \sum_{i=0}^{L-1} p(z_i) = 1$$

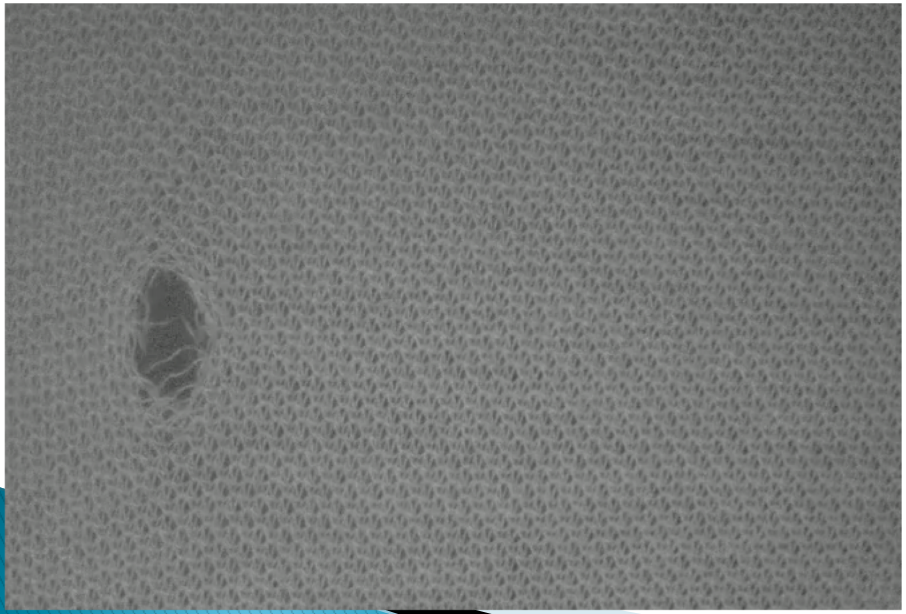
$$\mu_1(z) = \sum_{i=0}^{L-1} (z_i - m)^1 p(z_i) = \sum_{i=0}^{L-1} (z_i p(z_i) - m p(z_i)) = \sum_{i=0}^{L-1} z_i p(z_i) - \sum_{i=0}^{L-1} m p(z_i)$$

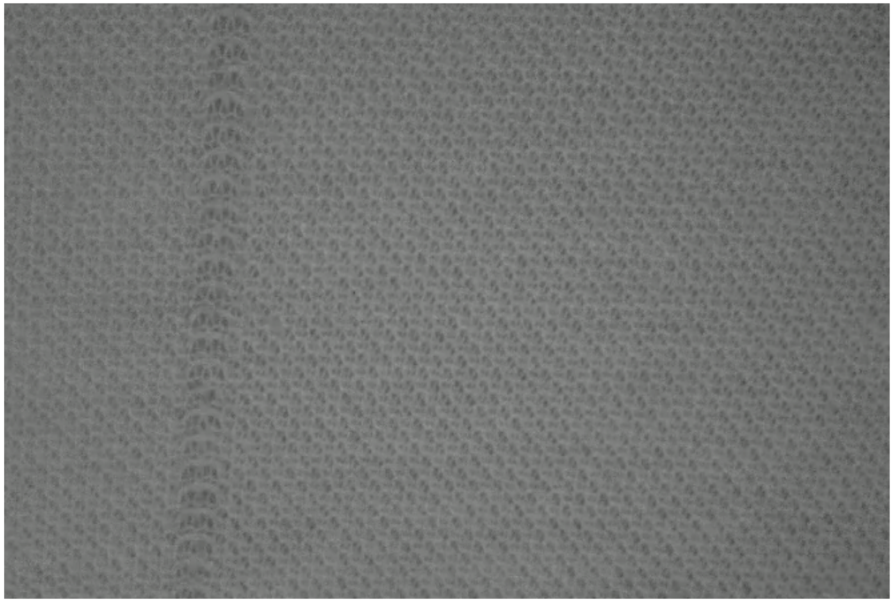
$$= m - m \sum_{i=0}^{L-1} p(z_i) = 0$$

- The second moment [the variance $\sigma^2(z) = \mu_2(z)$] is particularly important in texture description
  - a measure of intensity contrast
  - can establish descriptors of relative intensity smoothness
- the third moment, $\mu_3(z)$
  - a measure of the skewness of the histogram
- the fourth moment, $\mu_4(z)$
  - a measure of relative flatness of the histogram
- The fifth and higher moments
  - not so easily related to histogram shape
  - provide quantitative discrimination of texture content

Final Project







- For example,

$$R(z) = 1 - \frac{1}{1 + \sigma^2(x)}$$

  - the measure is 0 for areas of constant intensity (variance is zero, $\sigma^2(z)$=0)
  - approaches 1 for large values of $\sigma^2(z)$
  - In gray images [0, 255], the variance can be large, therefore, to normalize the variance to the interval [0, 1] is better
- This is done simply by dividing $\sigma^2(z)$ by $(L-1)^2$
- The standard deviation, $\sigma(z)$, is also used as a measure of texture because intuitive

Additional texture features

- Uniformity

$$U(z) = \sum_{i=0}^{L-1} p^2(z_i)$$

- Average entropy

$$e(z) = -\sum_{i=0}^{L-1} p\,(z_i) \log_2 p\,(z_i)$$

- p are in the range [0, 1] and their sum equals 1
- U is maximum for an image in which all intensity levels are equal ($p_0 = 1, p_1 = 0$)
- Entropy is a measure of variability, and is 0 for a constant image.

Texture descriptors based on histograms

a b c

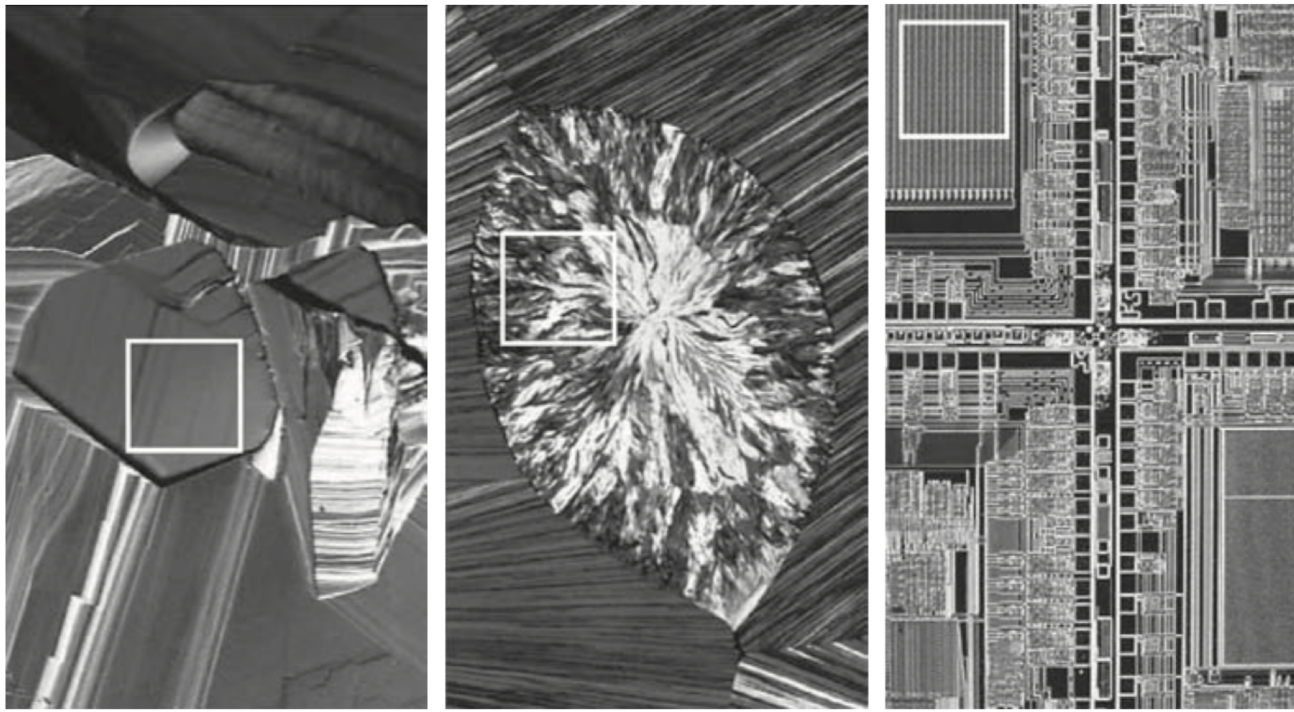


FIGURE 11.29 The white squares mark, from left to right, smooth, coarse, and regular textures. These are optical microscope images of a superconductor, human cholesterol, and a microprocessor. (Courtesy of Dr. Michael W. Davidson, Florida State University.)

TABLE 11.2 Statistical texture measures for the subimages in Fig. 11.29.

Texture	Mean	Standard deviation	R (normalized)	3rd moment	Uniformity	Entropy
Smooth	82.64	11.79	0.002	−0.105	0.026	5.434
Coarse	143.56	74.63	0.079	−0.151	0.005	7.783
Regular	99.72	33.73	0.017	0.750	0.013	6.674

Texture descriptors based on histograms

TABLE 11.2
Statistical texture measures for the subimages in Fig. 11.29.

Texture	Mean	Standard deviation	*R* (normalized)	3rd moment	Uniformity	Entropy
Smooth	82.64	11.79	0.002	−0.105	0.026	5.434
Coarse	143.56	74.63	0.079	−0.151	0.005	7.783
Regular	99.72	33.73	0.017	0.750	0.013	6.674

- mean
  - the average intensity of each region
  - is useful only as a rough idea of intensity
  - not texture
- standard deviation
  - more informative
  - first texture has significantly less variability in intensity (smoother) than the other two textures
- *R* also measures essentially the same thing as the standard deviation

Texture descriptors based on histograms

TABLE 11.2
Statistical texture measures for the subimages in Fig. 11.29.

Texture	Mean	Standard deviation	*R* (normalized)	3rd moment	Uniformity	Entropy
Smooth	82.64	11.79	0.002	−0.105	0.026	5.434
Coarse	143.56	74.63	0.079	−0.151	0.005	7.783
Regular	99.72	33.73	0.017	0.750	0.013	6.674

- 3rd moment
  - useful for determining the symmetry of histograms
  - whether they are skewed to the left (negative value) or the right (positive value)
  - useful only when variations between measurements are large
- Uniformity
  - the first subimage is smoother than others (more uniform than the rest)
  - the most random (lowest uniformity) corresponds to the coarse texture

Texture descriptors based on histograms

TABLE 11.2
Statistical texture measures for the subimages in Fig. 11.29.

Texture	Mean	Standard deviation	*R* (normalized)	3rd moment	Uniformity	Entropy
Smooth	82.64	11.79	0.002	−0.105	0.026	5.434
Coarse	143.56	74.63	0.079	−0.151	0.005	7.783
Regular	99.72	33.73	0.017	0.750	0.013	6.674

- Entropy
  - increase as uniformity decreases
  - first subimage has the lowest variation in intensity levels
  - coarse image the most
  - regular texture is in between the two extremes with respect to both of these measures

co-occurrence matrix

- Texture measurement by histograms
  - no spatial relationships information between pixels
- *co-occurrence matrix G*
  - Image level: 1~L
    - Gray-level image: 0~255 -> 1~256
  - Q : an operator defines the position of two pixels relative to each other
  - **G :** a matrix, element g_{ij} is the number of times with pixel pairs (z_i, z_j) occurring in image f
  - $1 \le i, j \le L$

L=8

Q defined as "one pixel immediately to the right"

FIGURE 11.30 How to construct a co-occurrence matrix.

1	1	7	5	3	2
5	1	6	1	2	5
8	8	6	8	1	2
4	3	4	5	5	1
8	7	8	7	6	2
7	8	6	2	6	2

Image *f*

	1	2	3	4	5	6	7	8
1	1	2	0	0	0	1	1	0
2	0	0	0	0	1	1	0	0
3	0	1	0	1	0	0	0	0
4	0	0	1	0	1	0	0	0
5	2	0	1	0	1	0	0	0
6	1	3	0	0	0	0	0	1
7	0	0	0	0	1	1	0	2
8	1	0	0	0	0	2	2	1

Co-occurrence matrix **G**

The array on the left is a small image and the array on the right is matrix **G** with size LxL

Question

- What is the size of the co-occurrence matrix G in a gray-level image ?

256x256

- If image *f* is 128x128 and *Q* is defined as "one pixel immediately to the right", what is the sum of all elements in G?

127x127

- If Q as, “one pixel to the right and one pixel above(右上方),”
  - position (1,1) in **G** would have been 0
  - positions (1,3), (1,5), and (1,7) in **G** would all be 1

1	1	7	5	3	2
5	1	6	1	2	5
8	8	6	8	1	2
4	3	4	5	5	1
8	7	8	7	6	2
7	8	6	2	6	2

Image f

- co-occurrence matrices sometimes are used in sequences
- One approach for reducing computations is to quantize the intensities into a few bands in order to keep the size of **G** manageable
- Example,
  - 256 level image -> 8 level image
    - 1-32 -> 1
    - 33-64 -> 2
    - …
    - 225-256 -> 8
    - result in a co-occurrence matrix of size 8×8

Q defined as “one pixel immediately to the right”

1	1	7	5	3	2
5	1	6	1	2	5
8	8	6	8	1	2
4	3	4	5	5	1
8	7	8	7	6	2
7	8	6	2	6	2

Image f

- The total number, n, of pixel pairs that satisfy Q is equal to the sum of the elements of **G** (n = 30 in the above example)

$$p_{ij} = \frac{g_{ij}}{n}$$

is an estimate of the probability that a pair of points satisfying Q will have values (z_i , z_j)

- These probabilities are in the range [0, 1] and their sum is 1

$$\sum_{i=1}^{K} \sum_{j=1}^{K} p_{ij} = 1$$

where K is the row and column dimension of square matrix **G**

- **G** depends on Q
  - texture patterns can be detected by choosing an appropriate position operator and analyzing the elements of **G**
- The correlation descriptor are defined as follows:

$$m_r = \sum_{i=1}^{K} i \sum_{j=1}^{K} p_{ij}, \qquad m_c = \sum_{i=1}^{K} j \sum_{j=1}^{K} p_{ij}$$

$$\sigma_r^2 = \sum_{i=1}^{K} (i - m_r)^2 \sum_{j=1}^{K} p_{ij}$$

$$\sigma_c^2 = \sum_{j=1}^{K} (j - m_c)^2 \sum_{j=1}^{K} p_{ij}$$

Descriptors used for characterizing co-occurrence matrices of size $K \times K$. The term p_{ij} is the ij-th term of G divided by the sum of the elements of G

Descriptor	Explanation	Formula
Maximum probability	Measures the strongest response of **G**. The range of values is [0, 1].	$\max_{i,j}(p_{ij})$
Correlation	A measure of how correlated a pixel is to its neighbor over the entire image. The range of values is 1 to –1 corresponding to perfect positive and perfect negative correlations. This measure is not defined if either standard deviation is zero.	$\sum_{i=1}^{K}\sum_{j=1}^{K}\frac{(i-m_r)(j-m_c)p_{ij}}{\sigma_r \sigma_c}$ $\sigma_r \neq 0;\ \sigma_c \neq 0$
Contrast	A measure of intensity contrast between a pixel and its neighbor over the entire image. The range of values is 0 (when **G** is constant) to $(K-1)^2$.	$\sum_{i=1}^{K}\sum_{j=1}^{K}(i-j)^2 p_{ij}$

Descriptors used for characterizing co-occurrence matrices of size $K \times K$. The term p_{ij} is the ij-th term of G divided by the sum of the elements of G

Uniformity (also called Energy)	A measure of uniformity in the range [0, 1]. Uniformity is 1 for a constant image.	$\sum_{i=1}^{K}\sum_{j=1}^{K} p_{ij}^2$
Homogeneity	Measures the spatial closeness to the diagonal of the distribution of elements in **G**. The range of values is [0, 1], with the maximum being achieved when **G** is a diagonal matrix.	$\sum_{i=1}^{K}\sum_{j=1}^{K} \frac{p_{ij}}{1 + \lvert i - j \rvert}$
Entropy	Measures the randomness of the elements of **G**. The entropy is 0 when all p_{ij}'s are 0, and is maximum when the p_{ij}'s are uniformly distributed. The maximum value is thus $2\log_2 K$.	$-\sum_{i=1}^{K}\sum_{j=1}^{K} p_{ij} \log_2 p_{ij}$

- If we let

$$P(i) = \sum_{j=1}^{K} p_{ij}$$

$$P(j) = \sum_{i=1}^{K} p_{ij}$$

- then the m_r and m_c can be written as

$$m_r = \sum_{i=1}^{K} iP(i)$$

$$m_c = \sum_{j=1}^{K} jP(j)$$

and the σ_r^2 and σ_c^2 can be written as

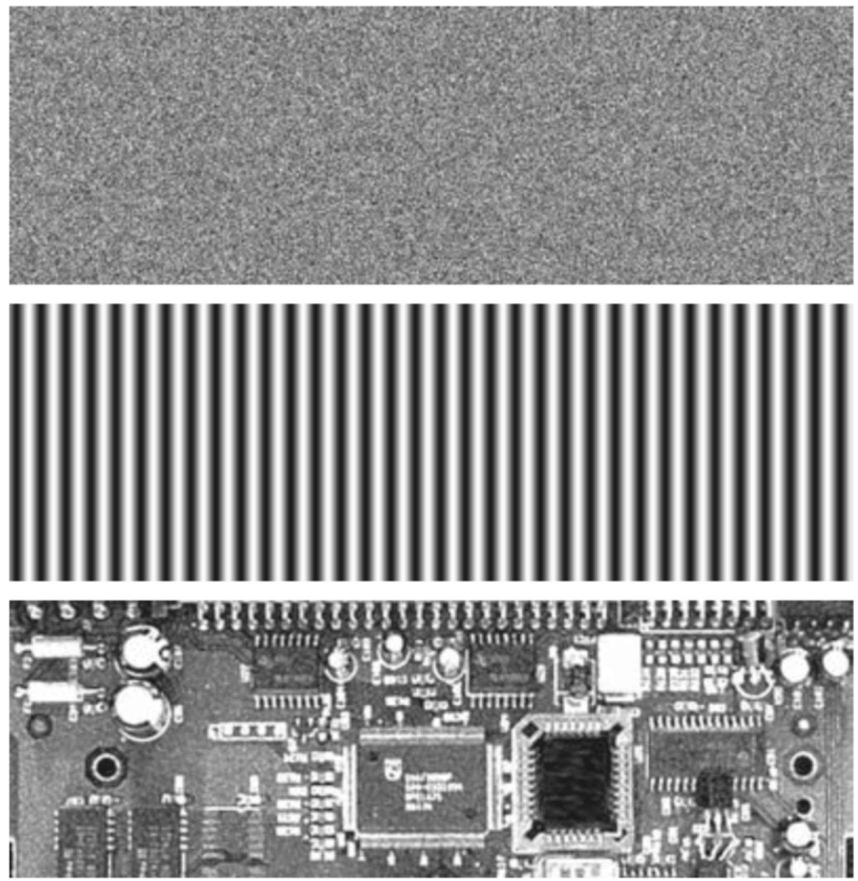
$$\sigma_r^2 = \sum_{i=1}^{K} (i - m_r)^2 \, P(i)$$

$$\sigma_c^2 = \sum_{j=1}^{K} (j - m_c)^2 \, P(j)$$

- m_r is in the form of a mean computed along rows of the normalized **G**
- m_c is a mean computed along the columns
- σ_r and σ_c are in the form of standard deviations (square roots of the variances) computed along rows and columns, respectively.
- Each of these terms is a scalar, independently of the size of **G**
- Remember, neighbors do not necessarily have to be adjacent

Using descriptors to characterize co-occurrence matrice

- Examples



a
b
c

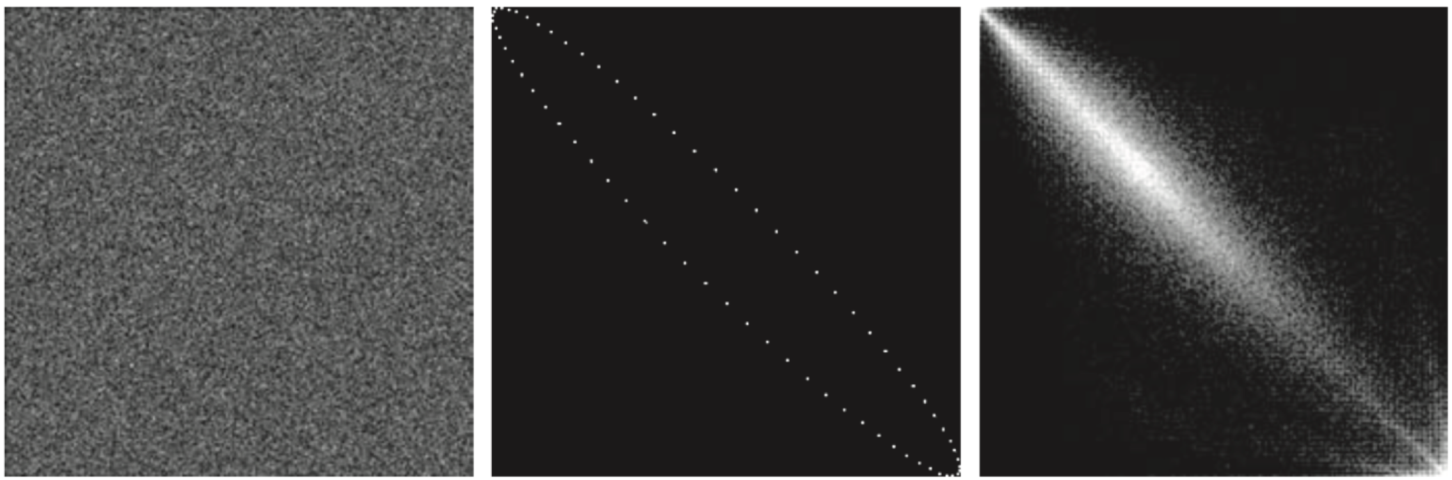
FIGURE 11.31 Images whose pixels have (a) random, (b) periodic, and (c) mixed texture patterns. Each image is of size 263×800 pixels.

Qualitative observation

- This example has two objectives
  - show values of the descriptors for the three co-occurrence matrices, **G**1, **G**2,and **G**3, corresponding (from top to bottom) to these images
  - illustrate how sequences of co-occurrence matrices can be used to detect texture patterns in an image
- $L = 256$, the position operator "one pixel immediately to the right"

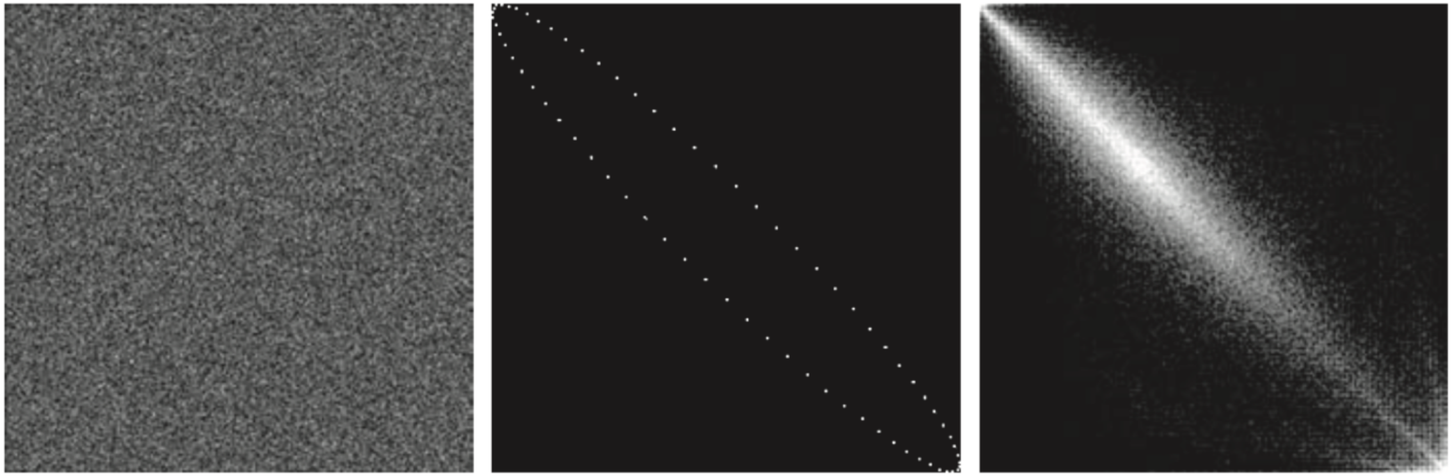
a b c

FIGURE 11.32 256×256 co-occurrence matrices $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{G}_3$, corresponding from left to right to the images in Fig. 11.31.



Qualitative observation

- The value at coordinates (i, j) in these images is the number of times that pixel pairs with intensities z_i and z_j occur in f in the position specified by Q
- Fig. 11.32(a) is a random image, given the nature of the image from which it was obtained.



a b c

FIGURE 11.32 256×256 co-occurrence matrices $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{G}_3$, corresponding from left to right to the images in Fig. 11.31.

Qualitative observation

- Figure 11.32(b)
  - The first obvious feature is the symmetry about the main diagonal
  - Because of the symmetry of the sine wave(Fig 11.31(b)), the number of counts for a pair (z_i , z_j) is the same as for the pair (z_j , z_i), which produces a symmetric co-occurrence matrix
  - The nonzero elements of **G**2 are sparse because value differences between horizontally adjacent pixels in a horizontal sine wave are relatively small
  - It helps to remember in interpreting these concepts that a digitized sine wave is a staircase, with the height and width of each step depending on the frequency of the sine wave and the number of amplitude levels used in representing the function.

Qualitative observation

- Fig. 11.32(c)
  - The structure of co-occurrence matrix **G**3 is more complex
  - High count values are grouped along the main diagonal
  - their distribution is more dense than for **G**2
  - an image with a rich variation in intensity values, but few large jumps in intensity between adjacent pixels
  - large areas characterized by low variability in intensities
  - high transitions in intensity occur at object boundaries, but these counts are low comparing with large areas

Quantification

- Using descriptors
- To use descriptors, the co-occurrence matrices must be normalized by dividing them by the sum of their elements

TABLE 11.4
Descriptors evaluated using the co-occurrence matrices displayed as images in Fig. 11.32.

Normalized Co-occurrence Matrix	Maximum Probability	Correlation	Contrast	Uniformity	Homogeneity	Entropy
$\mathbf{G}_1/n_1$	0.00006	–0.0005	10838	0.00002	0.0366	15.75
$\mathbf{G}_2/n_2$	0.01500	0.9650	00570	0.01230	0.0824	06.43
$\mathbf{G}_3/n_3$	0.06860	0.8798	01356	0.00480	0.2048	13.58

Quantification

- **G3**: Maximum Probability $\max_{i,j}(p_{ij})$
  - The highest probability corresponds to the third co-occurrence matrix, which tells us that this matrix has the highest number of counts (largest number of pixel pairs occurring in the image relative to the positions in Q) than the other two matrices. This agrees with our analysis of **G**3.

TABLE 11.4
Descriptors evaluated using the co-occurrence matrices displayed as images in Fig. 11.32.

Normalized Co-occurrence Matrix	Maximum Probability	Correlation	Contrast	Uniformity	Homogeneity	Entropy
$\mathbf{G}_1/n_1$	0.00006	–0.0005	10838	0.00002	0.0366	15.75
$\mathbf{G}_2/n_2$	0.01500	0.9650	00570	0.01230	0.0824	06.43
$\mathbf{G}_3/n_3$	0.06860	0.8798	01356	0.00480	0.2048	13.58

Quantification

$$\sum_{i=1}^{K}\sum_{j=1}^{K}\frac{(i-m_r)(j-m_c)p_{ij}}{\sigma_r\,\sigma_c}$$

$$\sigma_r \neq 0;\ \sigma_c \neq 0$$

- **G2**: the highest correlation
  - the intensities in the second image are highly correlated (the repetitiveness of the sinusoidal pattern)
  - the correlation for **G1** is essentially zero (random images, no correlation between adjacent pixels

TABLE 11.4
Descriptors evaluated using the co-occurrence matrices displayed as images in Fig. 11.32.

Normalized Co-occurrence Matrix	Maximum Probability	Correlation	Contrast	Uniformity	Homogeneity	Entropy
$\mathbf{G}_1/n_1$	0.00006	–0.0005	10838	0.00002	0.0366	15.75
$\mathbf{G}_2/n_2$	0.01500	0.9650	00570	0.01230	0.0824	06.43
$\mathbf{G}_3/n_3$	0.06860	0.8798	01356	0.00480	0.2048	13.58

Quantification

$$\sum_{i=1}^{K}\sum_{j=1}^{K}(i-j)^2 p_{ij}$$

- Contrast Descriptor: highest **G**1, lowest **G**2
  - less random, lower contrast
  - Random: G1>G3>G2, Contrast: G1>G3>G2
  - The $(i-j)^2$ terms are the same for any **G**.
  - the probabilities are the factors that determine the value of contrast
  - **G**1 has the lowest maximum probability, the other two matrices have many more zero or near-zero probabilities
  -

TABLE 11.4
Descriptors evaluated using the co-occurrence matrices displayed as images in Fig. 11.32.

Normalized Co-occurrence Matrix	Maximum Probability	Correlation	Contrast	Uniformity	Homogeneity	Entropy
$\mathbf{G}_1/n_1$	0.00006	–0.0005	10838	0.00002	0.0366	15.75
$\mathbf{G}_2/n_2$	0.01500	0.9650	00570	0.01230	0.0824	06.43
$\mathbf{G}_3/n_3$	0.06860	0.8798	01356	0.00480	0.2048	13.58

Quantification

$$\sum_{i=1}^{K}\sum_{j=1}^{K} p_{ij}^2$$

- Uniformity
  - increases as a function of the values of the probabilities squared
  - the less randomness there is in an image, the higher the uniformity descriptor will be

TABLE 11.4
Descriptors evaluated using the co-occurrence matrices displayed as images in Fig. 11.32.

Normalized Co-occurrence Matrix	Maximum Probability	Correlation	Contrast	Uniformity	Homogeneity	Entropy
$\mathbf{G}_1/n_1$	0.00006	–0.0005	10838	0.00002	0.0366	15.75
$\mathbf{G}_2/n_2$	0.01500	0.9650	00570	0.01230	0.0824	06.43
$\mathbf{G}_3/n_3$	0.06860	0.8798	01356	0.00480	0.2048	13.58

Quantification

$$\sum_{i=1}^{K}\sum_{j=1}^{K}\frac{p_{ij}}{1+|i-j|}$$

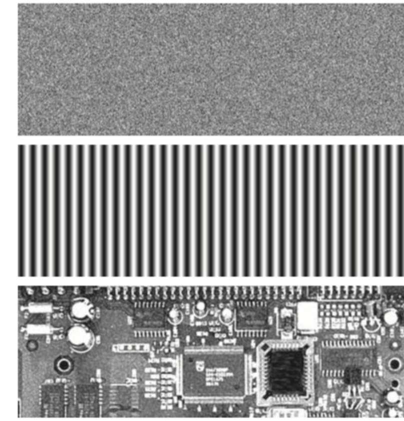
- Homogeneity
  - the concentration of values of **G** with respect to the main diagonal
  - denominator $(1+|i-j|)$ are the same for all co-occurrence matrices
  - $(1+|i-j|)$ decrease as i and j become closer (closer to the main diagonal)
  - the matrix with the highest values of probabilities (numerator terms) near the main diagonal will have the highest value of homogeneity

TABLE 11.4
Descriptors evaluated using the co-occurrence matrices displayed as images in Fig. 11.32.

Normalized Co-occurrence Matrix	Maximum Probability	Correlation	Contrast	Uniformity	Homogeneity	Entropy
$\mathbf{G}_1/n_1$	0.00006	–0.0005	10838	0.00002	0.0366	15.75
$\mathbf{G}_2/n_2$	0.01500	0.9650	00570	0.01230	0.0824	06.43
$\mathbf{G}_3/n_3$	0.06860	0.8798	01356	0.00480	0.2048	13.58

Quantification

$$-\sum_{i=1}^{K}\sum_{j=1}^{K} p_{ij} \log_2 p_{ij}$$



- Entropy,
  - measures of randomness in the corresponding images
  - **G**1 had the highest value because it was totally random
  - The other two entries are self-explanatory
  - **G**1 entropy is near the theoretical maximum of 16 ($2 \log_2 256 = 16$)

TABLE 11.4

Descriptors evaluated using the co-occurrence matrices displayed as images in Fig. 11.32.

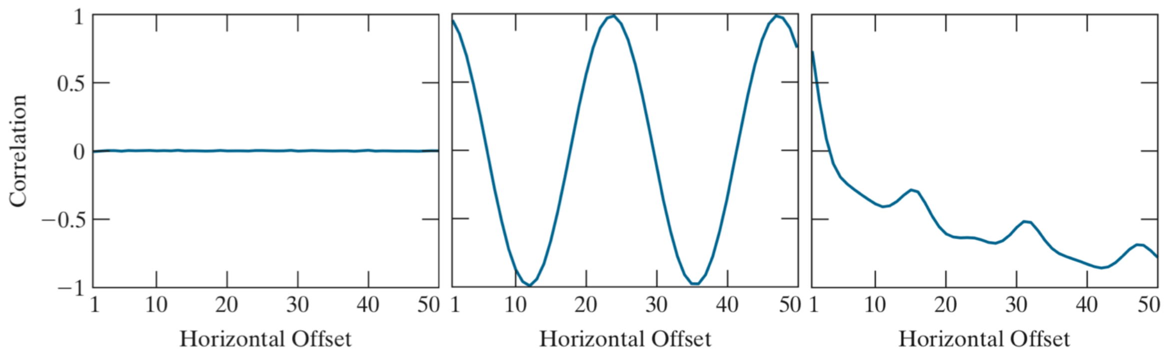
Normalized Co-occurrence Matrix	Maximum Probability	Correlation	Contrast	Uniformity	Homogeneity	Entropy
$\mathbf{G}_1/n_1$	0.00006	−0.0005	10838	0.00002	0.0366	15.75
$\mathbf{G}_2/n_2$	0.01500	0.9650	00570	0.01230	0.0824	06.43
$\mathbf{G}_3/n_3$	0.06860	0.8798	01356	0.00480	0.2048	13.58

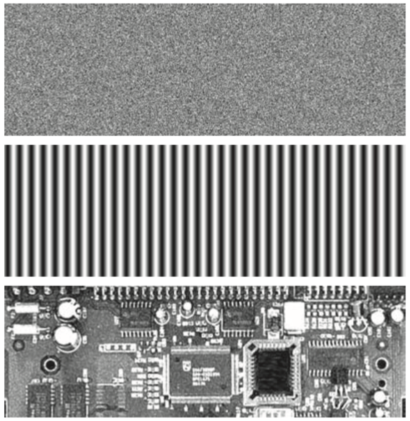
$$\sum_{i=1}^{K}\sum_{j=1}^{K}\frac{(i-m_r)(j-m_c)p_{ij}}{\sigma_r\ \sigma_c}$$

$$\sigma_r \neq 0;\ \sigma_c \neq 0$$

- any sections in these images that contain repetitive components ? (i.e., periodic textures)
- One way is to examine the correlation descriptor for sequences of co-occurrence matrices, derived from these images by increasing the distance between neighbors
- it is customary when working with sequences of co-occurrence matrices to quantize the number of intensities in order to reduce matrix size and corresponding computational load
- The following results were obtained using $L = 8$

- Figure 11.33 shows plots of the correlation descriptors as a function of horizontal "offset" (i.e., horizontal distance between neighbors) from 1 (for adjacent pixels) to 50.
- (a) shows that all correlation values are near 0, indicating that no such patterns were found in the random image
- (b) is a clear indication that the input image is sinusoidal in the horizontal direction
- the correlation function starts at a high value, then decreases as the distance between neighbors increases, and then repeats itself





a b c

FIGURE 11.33 Values of the correlation descriptor as a function of offset (distance between "adjacent" pixels) corresponding to the (a) noisy, (b) sinusoidal, and (c) circuit board images in Fig. 11.31.

- (c) shows that the correlation descriptor associated with the circuit board image decreases initially, but has a strong peak for an offset distance of 16 pixels
- upper solder joints form a repetitive pattern approximately 16 pixels apart
- The next major peak is at 32 (by the same pattern), but the amplitude of peak is lower because the number of repetitions at 32 is less than at 16 pixels
- A similar observation explains the even smaller peak at an offset of 48 pixels

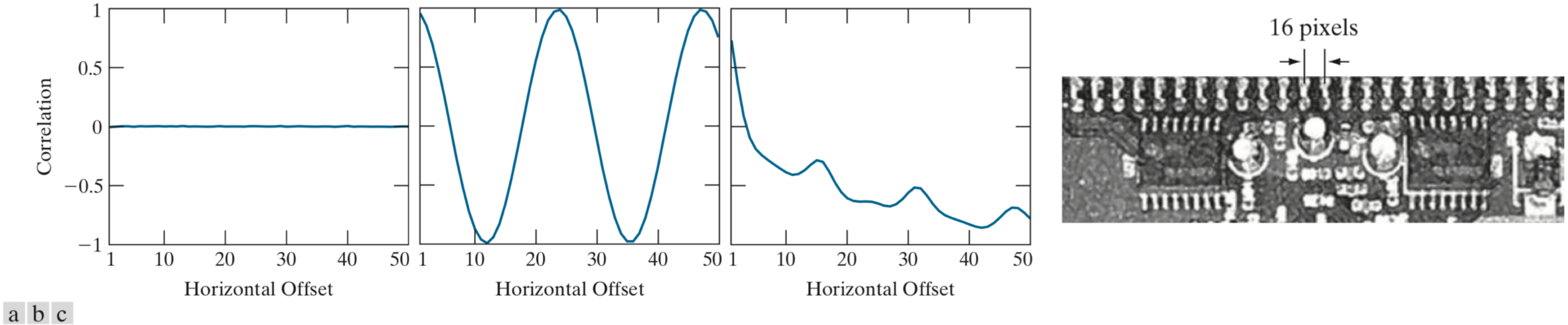


FIGURE 11.33 Values of the correlation descriptor as a function of offset (distance between "adjacent" pixels) corresponding to the (a) noisy, (b) sinusoidal, and (c) circuit board images in Fig. 11.31.

Spectral techniques

Spectral Approaches

- Fourier spectrum is suited for describing the directionality of periodic or semiperiodic 2-D patterns in an image
- global texture patterns are distinguished as concentrations of high-energy bursts in the spectrum
- the Fourier spectrum are useful for texture description
  - prominent peaks in the spectrum give the principal direction of the texture patterns
  - location of the peaks in the frequency plane gives the fundamental spatial period of the patterns
  - eliminating any periodic components via filtering leaves nonperiodic image elements, which can then be described by statistical techniques
  - spectrum is symmetric about the origin, so only needs half of the frequency plane -> every periodic pattern is associated with only one peak in the spectrum, not two

- spectrum features
  - expressing the spectrum in polar coordinates to yield a function $S(r, \theta)$, where S is the spectrum function, and r and (r, θ) are the variables in this coordinate system
  - For each direction θ, $S(r, \theta)$ may be considered a 1-D function $S_\theta(r)$
  - For each frequency r, $S_r(\theta)$ is a 1-D function
  - Analyzing $S_\theta(r)$ for a fixed value of (θ) yields the behavior of the spectrum (e.g., the presence of peaks) along a radial direction from the origin, whereas analyzing $S_r(\theta)$ for a fixed value of r yields the behavior along a circle centered on the origin

- A more global description is obtained by integrating (summing for discrete variables) these functions:

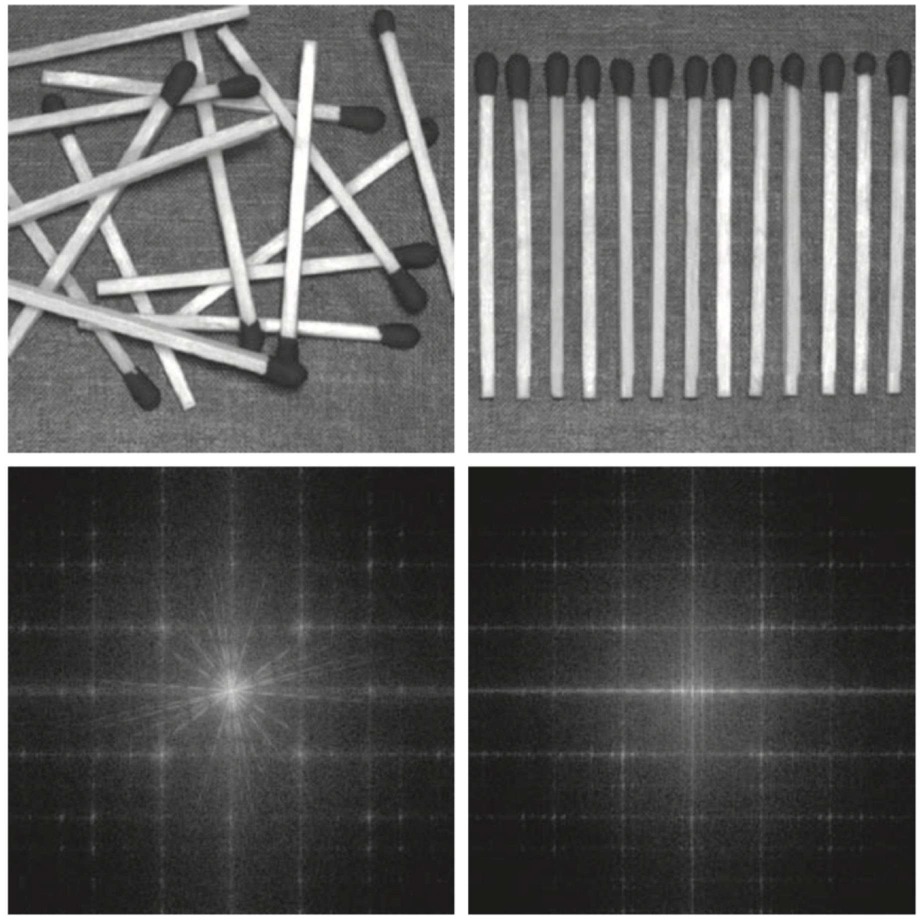
$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r)$$

$$S(\theta) = \sum_{r=1}^{R_0} S_r(\theta)$$

where R_0 is the radius of a circle centered at the origin

- a pair of values $[S(r), S(\theta)]$ for each pair of coordinates (r, θ)
- generate two 1-D functions, $S(r)$ and $S(\theta)$, that constitute a spectral-energy description of texture for an entire image or region
- Descriptors useful for this purpose are
  - the location of the highest value
  - the mean and variance of both the amplitude and axial variations
  - the distance between the mean and the highest value of the function

Spectral texture



a	b
c	d

FIGURE 11.35 (a) and (b) Images of random and ordered objects. (c) and (d) Corresponding Fourier spectra. All images are of size 600 × 600 pixels.

Spectral texture

- randomly distributed objects vs. periodically arranged objects
- periodic bursts of energy extending quadrilaterally in two dimensions in both Fourier spectra are due to the periodic texture of the coarse background material on which the objects rest.
- randomly distributed objects => random orientation
- periodically arranged objects => along the horizontal axis (strong vertical edges in image)

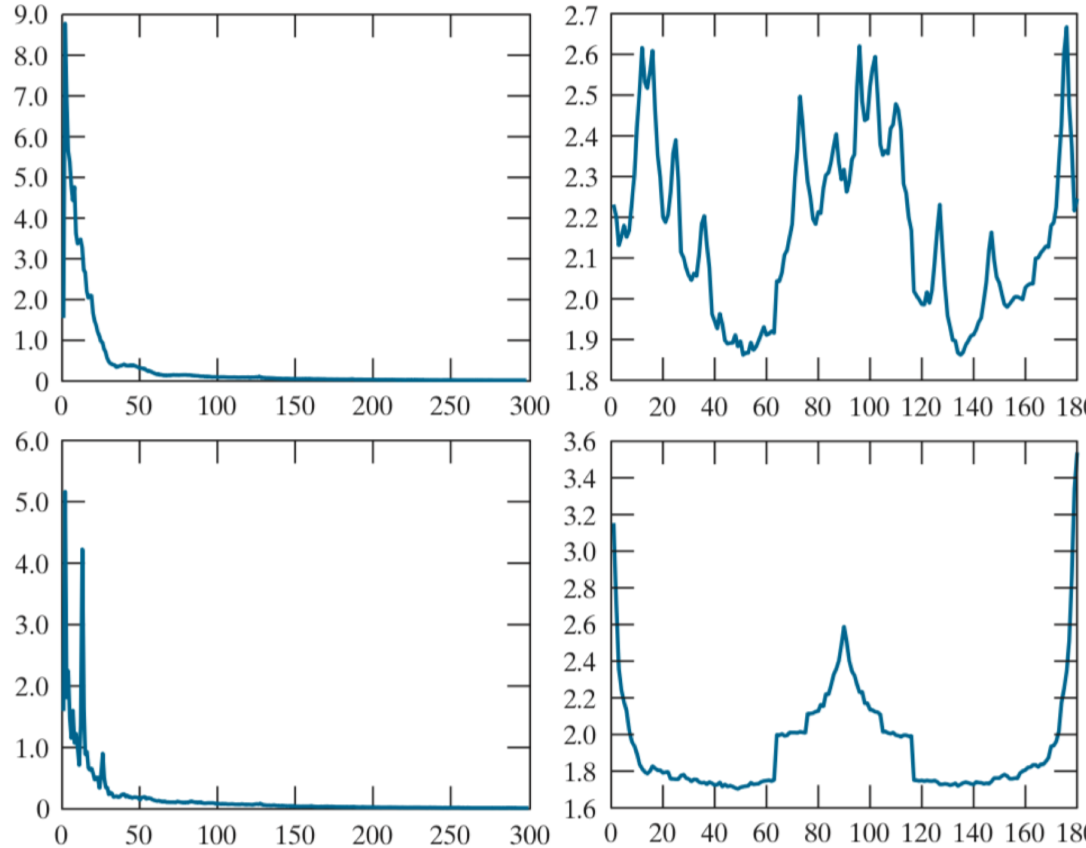
Spectral texture

- The plot of $S(r)$ for the random objects shows no strong periodic components
  - no dominant peaks in the spectrum besides the peak at the origin (dc component)
- The plot of $S(r)$ for the ordered objects shows a strong peak near $r = 15$ and a smaller one near $r = 25$
  - corresponding to the periodic horizontal repetition of the light (objects) and dark (background) regions
- the energy bursts in Fig. 11.35(c) is quite apparent in the plot of $S(\theta)$ in Fig. 11.36(b)
- Fig. 11.36(d) shows strong energy components in the region near the origin and at $90°$ and $180°$, consistent with the energy distribution of the spectrum in Fig. 11.35(d)

Spectral texture

a b
c d

FIGURE 11.36 (a) and (b) Plots of $S(r)$ and $S(\theta)$ for Fig. 11.35(a). (c) and (d) Plots of $S(r)$ and $S(\theta)$ for Fig. 11.35(b). All vertical axes are $\times 10^5$.



灰階共生矩陣

Grey Level Co-occurrence Matrix(GLCM)

Definition

- Definition: The GLCM is a tabulation of how often different combinations of pixel brightness values (grey levels) occur in an image

Framework for the GLCM

選擇兩個相鄰的點，稱作reference pixel與neighbour pixel，計算他們的值出現的次數後記錄在表格上

0	1	0	2
0	2	1	1
3	1	0	0
0	0	2	3

4 x 4 image

2	1	3	0
2	1	0	0
0	1	0	1
0	1	0	0

GLCM

0.16	0.08	0.25	0
0.16	0.08	0	0
0	0.08	0	0.08
0	0.08	0	0

Nomalized GLCM

Expressing the GLCM as a probability:

Normalization equation:

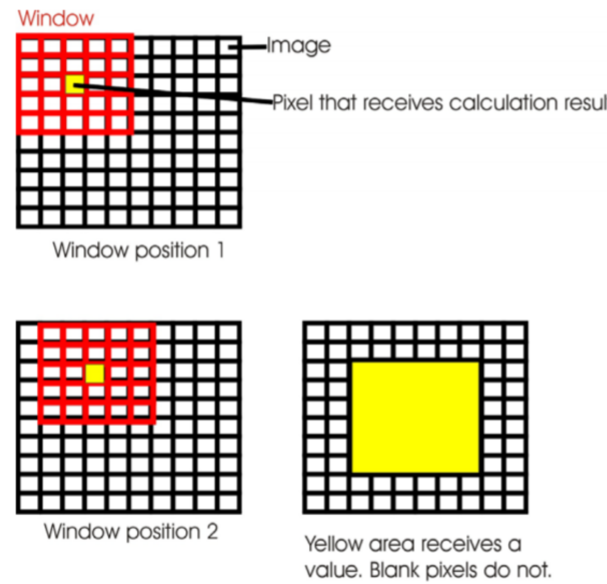
$$P_{i,j} = \frac{V_{i,j}}{\sum_{i,j=0}^{N-1} V_{i,j}}$$

4	2	1	0
2	4	0	0
1	0	6	1
0	0	1	2

.166 (4/24)	.083 (2/24)	.042 (1/24)	0 (0/24)
.083	.166	0	0
.042	0	.250	.042
0	0	.042	.083

THE TEXTURE MEASURE CALCULATIONS

- Defining the neighbourhood area
- 在描述圖像時我們想看到pixel與pixel間的關係，像是岩石我們會形容是鋸齒狀的，這時就得將圖像區分區域來判斷紋理



THE TEXTURE MEASURE CALCULATIONS

- A. Contrast

1. *Contrast* group

Measures related to *contrast* use weights related to the *distance from the GLCM diagonal*.

Principle: To emphasize a large amount of contrast, create weights so that the calculation results in a larger figure when there is great contrast between adjacent pixels. Values on the GLCM diagonal show no contrast, and contrast increases away from the diagonal. So, create a weight that increases as distance from the diagonal increases.

$$\sum_{i,j=0}^{N-1} P_{i,j}(i-j)^2$$

Contrast equation

Explanation:

When i and j are equal, the cell is on the diagonal and (i-j)=0. These values represent pixels entirely similar to their neighbour, so they are given a weight of 0 (no contrast).
If i and j differ by 1, there is a small contrast, and the weight is 1.
If i and j differ by 2, contrast is increasing and the weight is 4.
The weights continue to increase exponentially as (i-j) increases.

THE TEXTURE MEASURE CALCULATIONS

Calculation example: for the horizontal GLCM

Contrast weights:				X	horizontal GLCM				=	Multiplication result			
0	1	4	9		0.166	0.083	0.042	0		0	0.083	.168	0
1	0	1	4		0.083	0.166	0	0		0.083	0	0	0
4	1	0	1		0.042	0	.249	0.042		.168	0	0	.042
9	4	1	0		0	0	0.042	0.083		0	0	.042	0

Sum of all elements in the multiplication result table = **0.586.** Below is the actual computation is it wold be entered in a spreadsheet or carried out by hand.

$$.166*(0-0)^2 + .083*(0-1)^2 + .042*(0-2)^2 + 0*(0-3)^2 +$$
$$.083*(1-0)^2 + .166*(1-1)^2 + 0*(1-2)^2 + 0*(1-3)^2 +$$
$$.042*(2-0)^2 + 0*(2-1)^2 + .250*(2-2)^2 + .042*(2-3)^2 +$$
$$0*(3-0)^2 + 0*(3-1)^2 + .042*(3-2)^2 + .083*(3-3)^2$$

$$= .166(0) + .083(1) + .042(4) + .083(1) + .166(0) + .042(4) + .25(0) +$$
$$.042(1) + .042(1) + .083(0)$$

$$= .083 + .168 + .083 + .168 + .042 + .042$$

$$\mathbf{= .586}$$

THE TEXTURE MEASURE CALCULATIONS

- C. Correlation

$$\sum_{i,j=0}^{N-1} P_{i,j}\left[\frac{(i-\mu_i)(j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}}\right]$$

Correlation equation

What does correlation mean?

Correlation between pixels means that there is a predictable and linear relationship between the two neighbouring pixels within the window, expressed by the regression equation.

Example: Suppose there is a very high correlation between the reference and neighbour pixel, expressed by n=2r+2, where n is the value of the neighbour and r of the reference. Therefore, if r=1, n is very likely to equal 4; if r=4, n=10, etc.

Python code

- 使用skimage.feature套件中的greycoprops函數判斷區塊的contrast與correlation值

```
for i in range(0, end_r, offset):
    for j in range(0, end_c, offset):
        f = image[ i : i + size_filter , j : j + size_filter ]
        result = greycomatrix( f, [1],[0] )
        co_contrast[int(i/offset),int(j/offset)] = float(greycoprops(result, 'contrast'))
        #co_homogeneity[int(i/offset),int(j/offset)] = float(greycoprops(result, 'homogeneity'))
        co_correlation[int(i/offset),int(j/offset)] = float(greycoprops(result, 'correlation'))
```

Python code

- 計算平均

```
#print mean(no error)
'''
homogeneity_mean = np.mean(co_homogeneity)
contrast_mean = np.mean(co_contrast)
correlation_mean = np.mean(co_correlation)
print("homogeneity mean: %f seconds" % homogeneity_mean)
print("contrast mean: %f seconds" % contrast_mean)
print("correlation mean: %f seconds" % correlation_mean)
'''
```

- 高於threshold就將區塊標記

```
for i in range(x_filter):
    for j in range(y_filter):
        if co_correlation[i,j] > 0.825:
            if co_contrast[i,j] > 40:
                cv2.rectangle(image,( j*offset , i*offset ),( j*offset+50 , i*offset+50 ) ,(255,0,0),2)
```

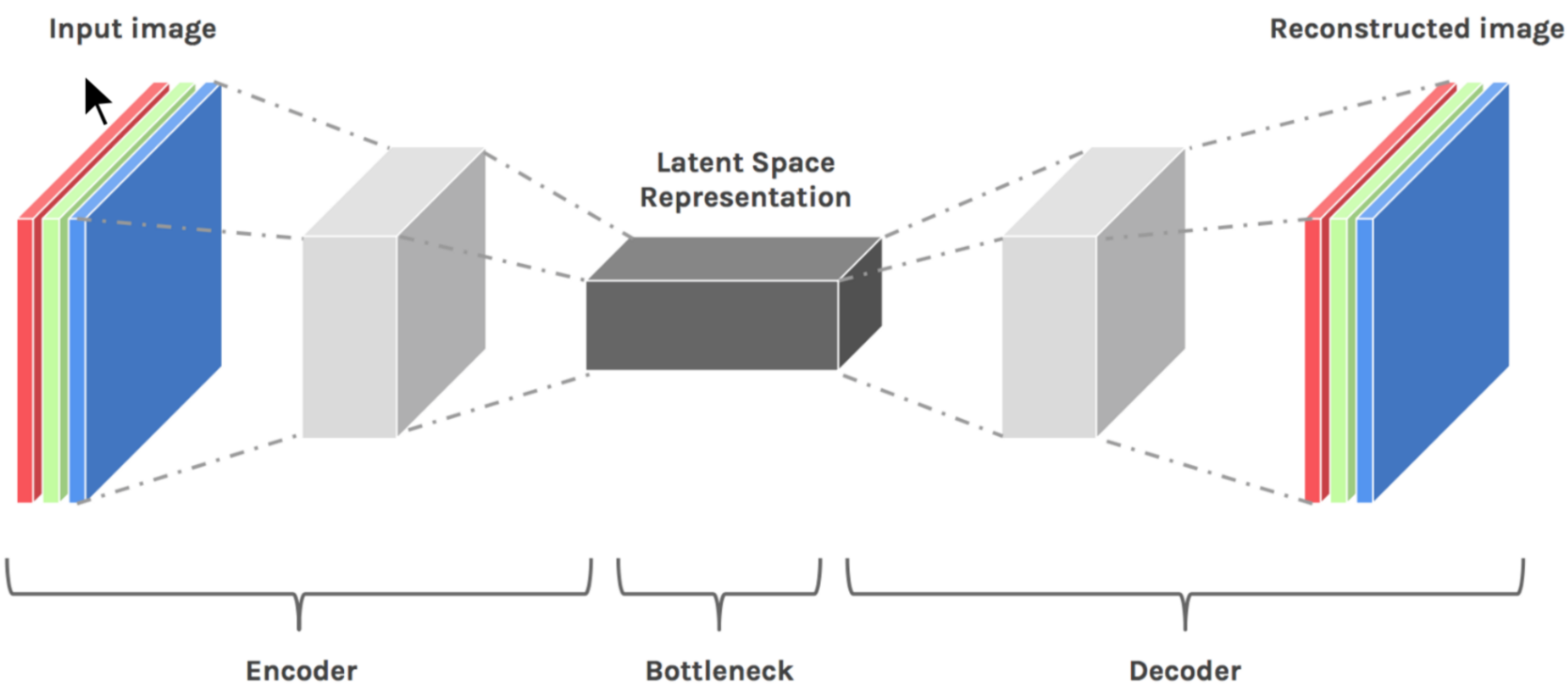
來源

- https://prism.ucalgary.ca/bitstream/handle/1880/51900/texture%20tutorial%20v%203_0%20180206.pdf?sequence=11&isAllowed=y
- https://juliaimages.org/ImageFeatures.jl/stable/tutorials/glcm/

Auto-Encoder

AutoEncoder的概念

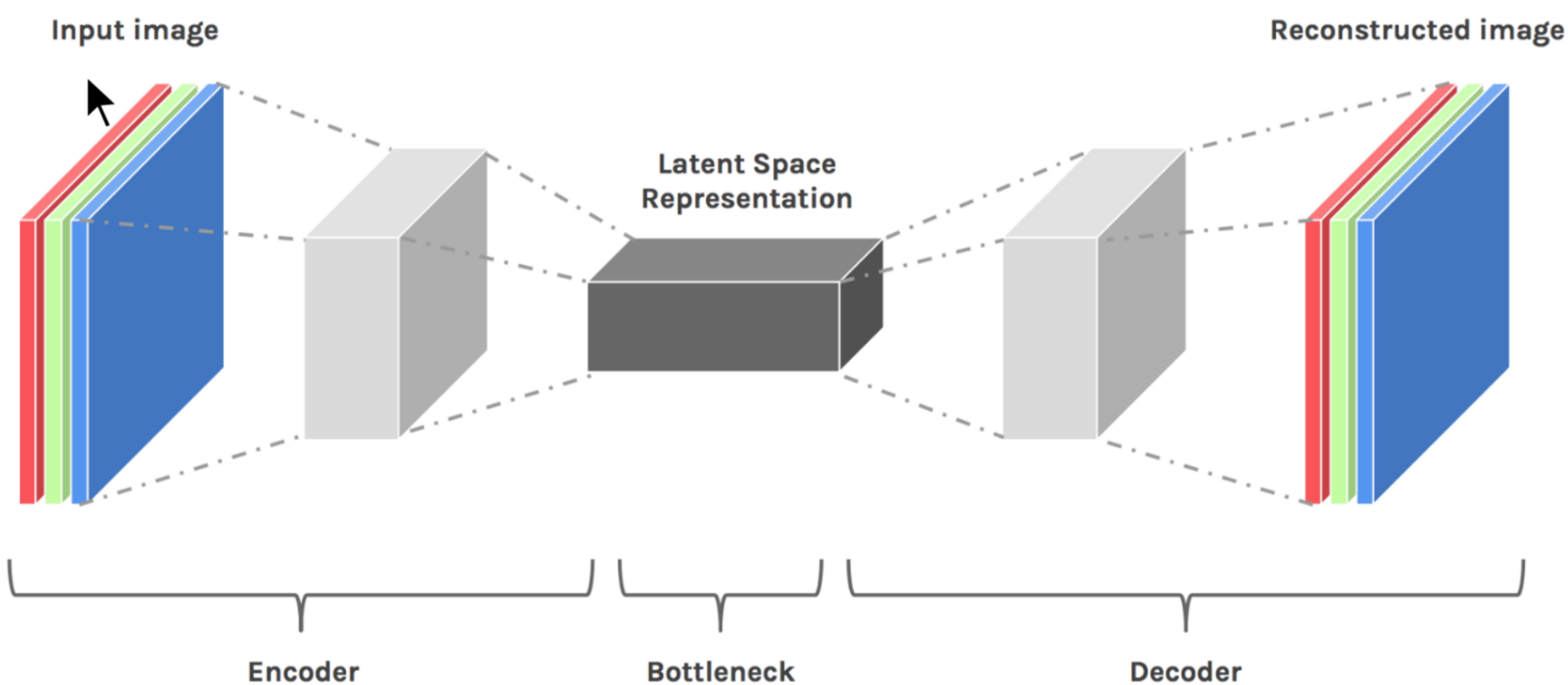
- 輸入一張影像資料後，經過Encoder後變成一個低維度Latent Vector後，還原時經過Decoder變回原本大小的圖樣
- 架構如下圖



https://medium.com/ai-academy-taiwan/what-are-autoencoders-175b474d74d1

AutoEncoder的實際應用

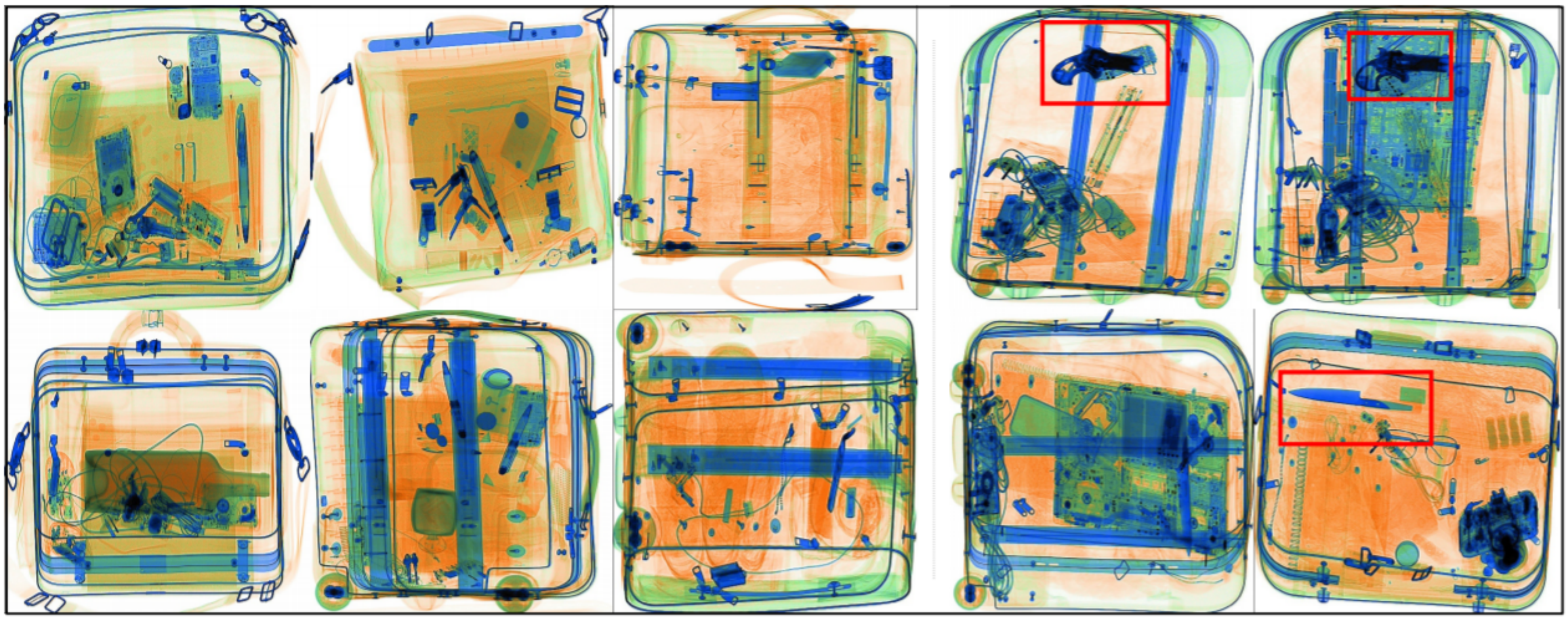
- 雖然AutoEncoder可以將影像縮小再放大還原，但畢竟降低了維度，很多時候影像的還原是經過神經網路內的過往訓練資料所填補上



https://medium.com/ai-academy-taiwan/what-are-autoencoders-175b474d74d1

AutoEncoder的實際應用

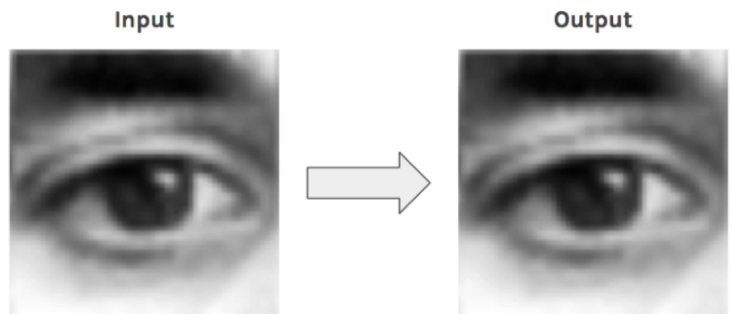
- AutoEncoder在還原與訓練資料相似的影像時，效果良好。但假如輸入影像與訓練資料有差異時，效果就會很差，因此我們可以拿來用作異常偵測



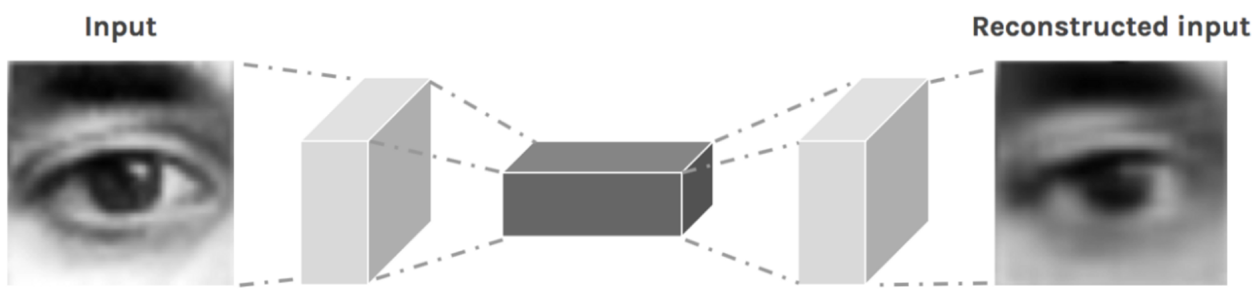
(a) Normal Data (X-ray Scans)　　(b) Normal + Abnormal Data (X-ray Scans)

https://medium.com/ai-academy-taiwan/what-are-autoencoders-175b474d74d168

AutoEncoder的實際應用



The CAE is trained to reconstruct its input



The reconstructed image is blurry

https://medium.com/hackernoon/autoencoders-deep-learning-bits-1-11731e200694

Basic autoencoder

- 最簡單的autoencoder，encoder與decoder都是一層全連結層所做成

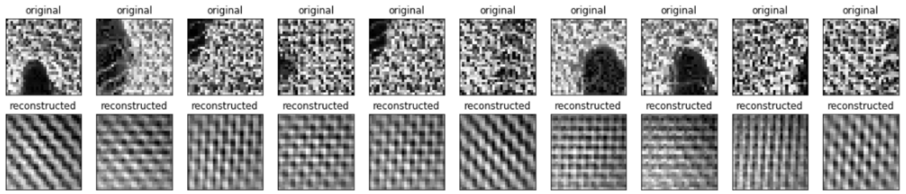
```
latent_dim = 64

class Autoencoder(Model):
  def __init__(self, latent_dim):
    super(Autoencoder, self).__init__()
    self.latent_dim = latent_dim
    self.encoder = tf.keras.Sequential([
      layers.Flatten(),
      layers.Dense(latent_dim, activation='relu'),
    ])
    self.decoder = tf.keras.Sequential([
      layers.Dense(784, activation='sigmoid'),
      layers.Reshape((28, 28))
    ])

  def call(self, x):
    encoded = self.encoder(x)
    decoded = self.decoder(encoded)
    return decoded

autoencoder = Autoencoder(latent_dim)
```

Source: tensorflow tutorials



Deep autoencoder

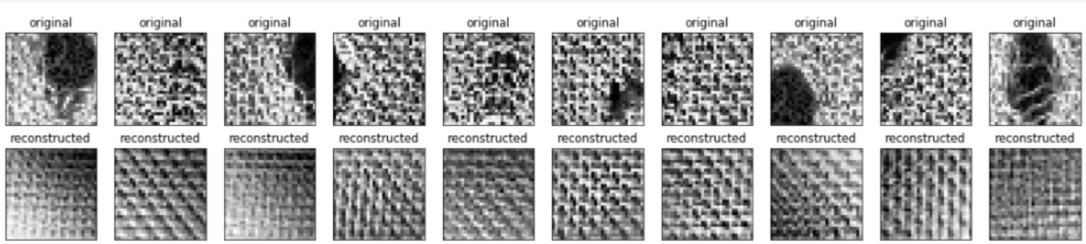
Encoder與decoder由多層神經網路所組成

```python
class AnomalyDetector(Model):
  def __init__(self):
    super(AnomalyDetector, self).__init__()
    self.encoder = tf.keras.Sequential([
      layers.Dense(32, activation="relu"),
      layers.Dense(16, activation="relu"),
      layers.Dense(8, activation="relu")])

    self.decoder = tf.keras.Sequential([
      layers.Dense(16, activation="relu"),
      layers.Dense(32, activation="relu"),
      layers.Dense(140, activation="sigmoid")])

  def call(self, x):
    encoded = self.encoder(x)
    decoded = self.decoder(encoded)
    return decoded

autoencoder = AnomalyDetector()
```



[Source: tensorflow tutorials]

convolutional autoencoder

- 使用convolutional但沒有使用全連結層的autoencoder

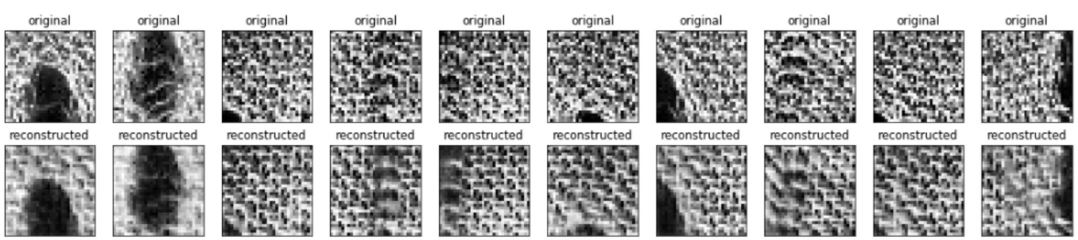
```
class Denoise(Model):
  def __init__(self):
    super(Denoise, self).__init__()
    self.encoder = tf.keras.Sequential([
      layers.Input(shape=(28, 28, 1)),
      layers.Conv2D(16, (3, 3), activation='relu', padding='same', strides=2),
      layers.Conv2D(8, (3, 3), activation='relu', padding='same', strides=2)])

    self.decoder = tf.keras.Sequential([
      layers.Conv2DTranspose(8, kernel_size=3, strides=2, activation='relu', padding='same'),
      layers.Conv2DTranspose(16, kernel_size=3, strides=2, activation='relu', padding='same'),
      layers.Conv2D(1, kernel_size=(3, 3), activation='sigmoid', padding='same')])

  def call(self, x):
    encoded = self.encoder(x)
    decoded = self.decoder(encoded)
    return decoded

autoencoder = Denoise()
```

Source: tensorflow tutorials



convolutional autoencoder

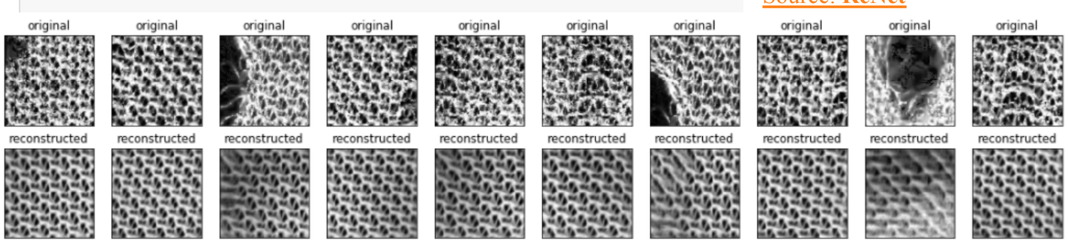
- encoder與decoder都是由CNN所構成，重建圖形的效果相較於前幾種來說最好

```
class CNN(Model):
  def __init__(self):
    super(CNN, self).__init__()
    self.encoder = tf.keras.Sequential([
      layers.Conv2D(8, 3, activation='relu', padding='same', strides=2),
      layers.Conv2D(16, 3, activation='relu'),
      layers.Conv2D(16, 3, activation='relu', padding='same', strides=2),
      layers.Conv2D(16, 3, activation='relu', padding='same', strides=2),
      layers.Conv2D(32, 3, activation='relu', padding='same', strides=2),
      layers.Conv2D(64, 3, activation='relu'),
      layers.Flatten(),
      layers.Dense(256, activation='relu'),
      layers.Dense(128, activation='relu'),

    ])

    self.decoder = tf.keras.Sequential([
      layers.Dense(256, activation='relu'),
      layers.Dense(1024, activation='relu'),
      layers.Reshape((4,4,64)),
      layers.Conv2DTranspose(64, 3, activation='relu'),
      layers.Conv2DTranspose(32, 3, strides=2, activation='relu', padding='same'),
      layers.Conv2DTranspose(16, 3, strides=2, activation='relu', padding='same'),
      layers.Conv2DTranspose(16, 3, strides=2, activation='relu', padding='same'),
      layers.Conv2DTranspose(16, 3, activation='relu'),
      layers.Conv2DTranspose(8, 3, strides=2, activation='relu', padding='same'),
      layers.Conv2D(1, 3, activation='relu', padding='same'),
    ])
```

Source: ReNet



Auto-Encoder的改善方向

- 使用不同loss function
- 將網路的階層與神經元數目增加或減少
- 調整判斷為異常的threshold值
- 使用其他類型的AutoEncoder架構
  - U-net
  - Adversarial Autoencoder
  - …