

## 多尺度方法應用於影像處理與機器學習

林政均<sup>†</sup>

<sup>†</sup>國立臺灣大學 電信工程學研究所

<sup>1</sup>E-mail: d11942011@ntu.edu.tw

指導教授：丁建均 博士

關鍵詞：多尺度分析、小波轉換、影像金字塔、影像處理、深度神經網路

### 摘要

多尺度方法已在訊號處理領域中廣泛應用，其在設計自適應演算法中扮演重要的角色，實務上，幾乎沒有任何一個演算法是足以符合任何狀況或解決任何問題，因此，在設計演算法前必須分析資料的性質，針對不同種類或等級的訊號特徵進行萃取，從而選擇最適等級的特徵，並進行後續的處理或應用，本文將簡單介紹傳統影像處理中較為廣泛使用的兩種方法—小波轉換及影像金字塔，以及多尺度在深度神經網路的應用。

### 前言

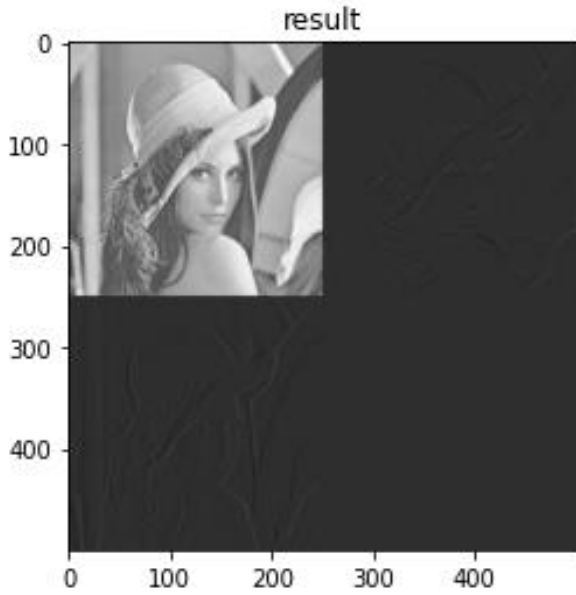
由於 1940 年代電腦的興起，自 1950 年代，人們開始思考是否能使用電腦模擬或仿造人的思維與行為，即仿生工程 (Bionic Engineering) 與人工智慧 (Artificial Intelligence)。人類的行為來自於人透過自身的感官得到環境的刺激 (Stimulation)，藉由刺激進行相對應的行為 (Action)，以利於人類的生存，人類的感官尤以視覺最為依賴，多數人對環境的感知皆以視覺 (Vision) 為主，因此延伸出了「電腦視覺」的領域。

電腦視覺 (Computer Vision) 是一門研究電腦眼中世界的科學，利用鏡頭做為人眼的模擬，電腦透過鏡頭得到來自環境的視覺感知後，再利用影像處理 (Image Processing) 演算法萃取出資訊的特徵，進而得到環境的資訊，以便電腦進行後續的分析 (Analysis) 或學習 (Learning)，1960 年代，Roberts 透過電腦自數字影像提取出三維結構 [1]，並分別對其與空間的關係進行描述，1980 年代，電腦視覺技術逐漸興起，不少學者開始從心理學、神經科學等等，對人類視覺進行研究，如今，電腦視覺領域主要項目即為針對影像中的特徵，進行三維目標的辨識，因此，辨識在影像處理為最典型的問題。

辨識 (Recognition) 即在一組影像資料中，判斷是否有特定物體、特徵或行為，簡易的任務如辨識幾何圖形、數字等等，困難的任務如人臉辨識、文本辨識等等，其皆屬於針對輸入影像進行特徵提取 (Feature Extraction)，而後將各個特徵進行分群 (Cluster) 或分類 (Classification)，達到辨識的效果。

由上述可得知特徵提取的重要性，若特徵提取並未達到預期的效果，便無法進行後續的分群或分類，亦無法達到辨識的效果，因此特徵提取衍生出許多演算法，諸如邊緣偵測 (Edge Detection) [2]、角點偵測 (Corner Detection) [3]、脊偵測 (Ridge Detection) [4] 等等。特徵提取是一種透過降低維度，將帶有資訊量的資料保留，捨棄其他冗餘部分的方法，其方法利於資料被歸納至更容易管理的群，同時不失描述原始資料的完整性。然而，特徵的維度，取決於目標物件的特性，例如，辨識三角形與辨識人臉，需要降低的維度不同，當資料中存在各種不同的特徵時，特徵降低到什麼程度，才能符合吾人的需求，因此衍生出了多尺度 (Multi-scale) 的方法。

多尺度意即將資料分成多個尺度，每一尺度皆為資料的特徵在不同維度下的映射，吾人即可藉由相對應的尺度特徵描述特定物件，可



圖一、原圖經哈爾小波轉換後之結果

在較低運算量的情況下，仍能保持資料的完整性。本文將介紹有關使用到「多尺度」方法的演算法。

## 1. 小波轉換

小波轉換 (Wavelet Transform) [5]，不同於傳統傅立葉轉換 (Fourier Transform) 使用無限長的三角函數做為基底，係一種利用有限長且會衰減的母小波 (Mother Wavelet) 做為基底。相較於傅立葉轉換變量只有頻率，小波轉換同時有尺度 (Scale) 和平移量 (Translation)，既可得知訊號的頻率成分，亦可得到在時域上的位置。

### 1.1 哈爾小波轉換

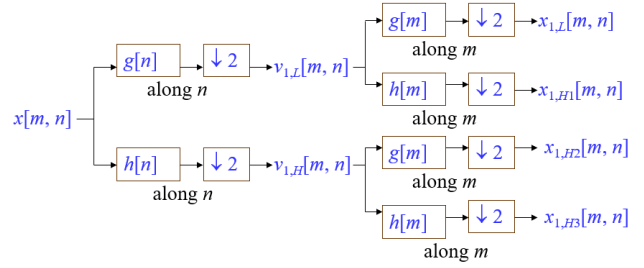
小波轉換是來自於小波分析 (Wavelet Analysis) 的結果，最早可追溯至 1909 年，由數學家 Haar Alfred 所提出的哈爾轉換 (Haar Transform) [6]，其母小波函數  $\psi(t)$  可描述為

$$\psi(t) = \begin{cases} 1, 0 \leq t < \frac{1}{2} \\ -1, -\frac{1}{2} \leq t < 0 \\ 0, \text{otherwise} \end{cases} \quad (1)$$

縮放函數  $\varphi(t)$  可描述為

$$\varphi(t) = \begin{cases} 1, 0 \leq t < 1 \\ 0, \text{otherwise} \end{cases} \quad (2)$$

濾波器 (Filter)  $h[n]$  被定義為



圖二、多貝西小波轉換之流程圖 [7]

$$h[n] = \begin{cases} \frac{1}{\sqrt{2}}, n = 0, 1 \\ 0, \text{otherwise} \end{cases} \quad (3)$$

由於影像屬於二維訊號，因此這裡僅針對二維訊號做介紹，二維 Haar Transform 是利用哈爾函數 (Haar Function) 做為調變函數 (Scaling Function)，表示為

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (4)$$

$$H_{2N} = \begin{bmatrix} H_N \otimes [1, 1] \\ I_N \otimes [1, -1] \end{bmatrix}, N = 1, 2, \dots \quad (5)$$

其優勢在於，大部分的運算皆為 0，僅需要進行加減，計算複雜度較小，因此簡單且快速，存在正交性，因此具可逆性，頻率只分為低頻與高頻成分，可用於分析訊號的局部特徵 (Localized Feature)，而在所有的正交 (Orthonormal) 小波轉換中，哈爾小波轉換是最簡單的一種轉換。

在 Python 中，pywt 函式庫可使用 `pywt.dwt2(img, 'haar')[8]`，輕易地針對影像進行哈爾小波轉換，其結果如圖一所示

由圖一可得知，哈爾小波轉換可輕易地將低頻與高頻成分分離。此外，除了利於降低資料維度外，同時可用來進行影像壓縮，原圖經過哈爾小波轉換後，像素值會落在 0 的附近，此時即可達到壓縮的效果。



圖三、原圖經多貝西小波轉換後之結果，左上為 $x_{1,L}$ ，右上為 $x_{1,H1}$ ，左下為 $x_{1,H2}$ ，右下為 $x_{1,H3}$

## 1.2 多貝西小波轉換

多貝西小波轉換 (Daubechies Wavelet Transform) [9]是由比利時的女性科學家 Ingrid Daubechies 所提出，主要用於離散小波轉換，多貝西小波也是一種具正交性的小波，且容易利用快速小波轉換 (Fast Wavelet Transform) [10]實現，吾人通常以濾波器的長度  $N$  ( $N$  為偶數)來描述濾波器屬於  $DN$ ，如  $N=2$  的多貝西小波為  $D2$ 。

多貝西小波是以消失動量 (Vanishing Moment)作為分類依據，其小波長度為消失動量的兩倍，調變函數及小波函數的平滑則皆會因消失動量的增加而增加，例如，若某一多貝西小波的消失動量為 1，則多貝西小波的小波濾波器及調變濾波器的長度  $N$  皆為 2，實際上，哈爾小波轉換即為多貝西小波  $N=2$  的特例。

### 1.2.1 多貝西小波轉換的建立

多貝西小波具有調變函數  $g[n]$ 及小波函數  $h[n]$ 兩個函數，其分別用於低通濾波 (Low Pass Filter)及高通濾波 (High Pass Filter)，若假設某一多貝西小波為  $2p$ -points case，建立調變函數與小波函數的係數步驟參見附錄。



圖四、經多貝西小波轉換後之影像，再進行多貝西小波逆轉換之結果

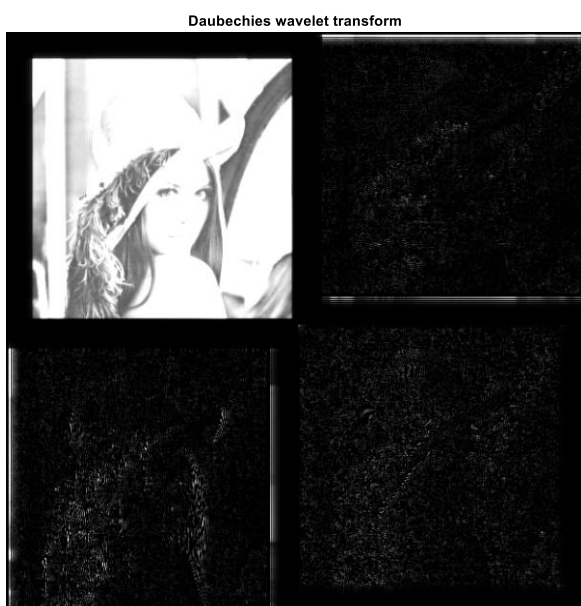
### 1.2.2 多貝西小波正轉換

有了調變函數與小波函數的係數後，即可進行多貝西小波轉換，其流程圖如圖二所示。

將某一輸入影像分別對調變函數  $g[n]$ 與小波函數  $h[n]$ 進行卷積後，再進行降採樣 (Downsampling)，降採樣維度為 2，可分別得到  $v_{1,L}$ 及 $v_{1,H}$ ，再分別將 $v_{1,L}$ 及 $v_{1,H}$ 對調變函數  $g[n]$ 與小波函數  $h[n]$ 進行卷積後，再進行降採樣 (Downsampling)，降採樣維度為 2，最終得到  $x_{1,L}$ 、 $x_{1,H1}$ 、 $x_{1,H2}$ 及 $x_{1,H3}$ ，其結果圖如圖三所示， $x_{1,L}$ 、 $x_{1,H1}$ 、 $x_{1,H2}$ 及 $x_{1,H3}$ 即由低頻到高频的結果。

### 1.2.3 多貝西小波逆轉換

逆轉換則將流程圖顛倒，將 $x_{1,L}$ 、 $x_{1,H1}$ 、 $x_{1,H2}$ 及 $x_{1,H3}$ 分別進行升採樣 (Upsampling)，升採樣維度為 2，再分別對調變函數與小波函數的逆轉換函數  $g_l[n]$ 及  $h_l[n]$ 進行卷積，得到 $v_{1,L}$ 及 $v_{1,H}$ ，再對 $v_{1,L}$ 及 $v_{1,H}$ 分別進行升採樣 (Upsampling)，升採樣維度為 2，再分別對調變函數與小波函數的逆轉換函數  $g_l[n]$ 及  $h_l[n]$ 進行卷積，最後得到原圖  $x[m,n]$ ，其逆轉換結果如圖四所示。



圖五、60-points case 下，多貝西小波轉換後之結果

由於產生大小比原圖小且具備 Multi-scale 的特徵圖 (Feature Map)，因此吾人可先針對低頻成分搜索 (Search) 是否有特定物件，若有，再往下一尺度進行搜索，已達到降低辨識時間的效果。

而依據經驗法則，p 值在 2~5 之間 (即 4~10-points case)，在實務上已經夠用，若 p 值過大，則會發生低頻成分亮度過亮的問題，逆轉換後的影像已不復當初，但仍可透過降低亮度回復，吾人將 p 值設為 30，即 60-points case 的多貝西小波轉換，其轉換結果與逆轉換結果如圖五和圖六所示

與哈爾小波轉換相同，除了進行 Multi-scale 利於圖形辨識之外，多貝西小波亦可應用於影像壓縮，JPEG 2000 即為使用離散小波轉換的壓縮方法。

## 2. 影像金字塔

影像金字塔 (Image Pyramid) [11] 是一種對影像進行多尺度描述的結構，藉由逐漸降低影像的尺度並以金字塔的形式排列而組成，越接近頂部尺度越低，主要可用於影像分割、壓縮、強化、特徵點匹配等等。



圖六、60-points case 下，多貝西小波逆轉換後之結果

## 2.1 高斯金字塔

由於影像經高斯濾波 (Gaussian Filter) 後會愈加模糊，致使較低頻成分不斷地被濾除，僅留下高頻成分，高斯金字塔 (Gaussian Pyramid) [11] 即利用高斯濾波和降採樣不斷地降低影像尺度，並將所有影像按照尺度由低往高排列而得。

### 2.1.1 高斯金字塔的構建

首先對原影像進行  $5 \times 5$  的高斯濾波，若原圖為  $G_0(x,y)$ ，即

$$L_1(x,y) = G(x,y) * G_0(x,y) \quad (6)$$

其中  $G(x,y)$  高斯卷積函數為

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (7)$$

而後對經過高斯濾波後的影像進行降採樣，則下一張影像即可表示為

$$G_1(x,y) = \text{Down}(L_1(x,y)) \quad (8)$$

利用上述規則，其後可推廣為，第 i 次降採樣的影像為  $G_i(x,y)$ ，則  $G_i(x,y)$  可表示為

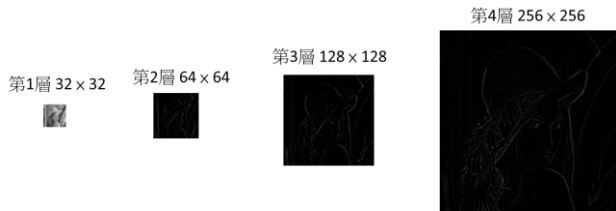
$$L_i(x,y) = G(x,y) * G_{i-1}(x,y) \quad (9)$$

$$G_i(x,y) = \text{Down}(L_i(x,y)) \quad (10)$$

其結果如圖七所示。



圖七、高斯金字塔之各影像尺度分布



圖八、拉普拉斯金字塔之各影像尺度分布

### 2.1.2 尺度空間

特徵點在影像中是帶有較大資訊量的點，其不易受光照、雜訊、線性轉換等影響，若任意兩張圖當中，皆具有相同的物件，其特徵點位置亦相同，僅因攝影時位置不同，而產生尺度、方向、大小等差別，而尺度空間理論 (Scale-Space Theory in Computer Vision) [12] 中提到，高斯核是唯一可以產生多尺度空間的核函數。

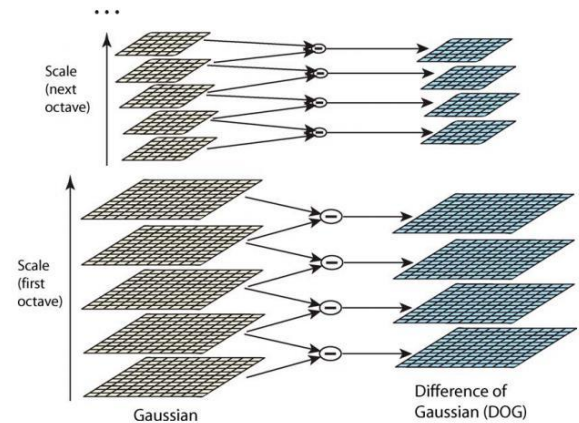
### 2.2 拉普拉斯金字塔

由於高斯金字塔是藉由降採樣的方式，捨棄部分資訊降低影像尺度，倘若吾人欲將低尺度的影像還原至高尺度的影像，便無法輕易地使用升採樣的方式還原，因升採樣係利用留下的元素插值得出捨棄的元素，然而捨棄的元素通常不會與留下的相同，因此必須記錄降採樣後再次上採樣時，與原始圖片的差異。

拉普拉斯金字塔 [11] 即記錄高斯金字塔每次降採樣後再進行升採樣時的結果與降採樣前的差異，以便後續能完整恢復每一層在降採樣前的影像，若  $L_i$  為兩結果的差異，可表示為

$$L_i = G_i - \text{Up}(\text{Down} * (G_i)) \quad (11)$$

其結果如圖八所示。



圖九、差分金字塔運算流程 [13]

### 2.3 差分金字塔

差分金字塔係由金字塔的某一組尺度相同但高斯濾波次數不同的前一層減去該層而得，以此得到差分金字塔，若使用於高斯金字塔，則此差分金字塔則為高斯差分 (Difference of Gaussian, DoG) 金字塔 [13]，其架構如圖九所示。

### 2.4 尺度不變特徵轉換

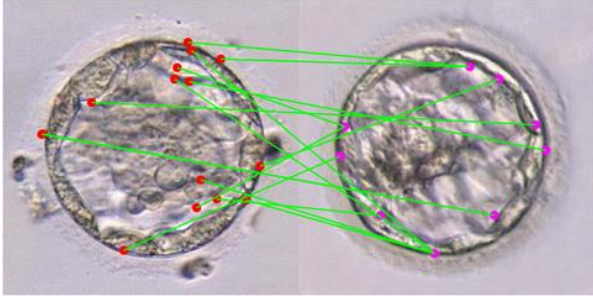
尺度不變特徵轉換 (Scale Invariant Feature Transform, SIFT) [13] 是一種搜索局部特徵的演算法，是 1999 年由 Lowe 提出的，其演算法分成四個步驟：

- (1) 尺度空間極值檢測
- (2) 關鍵點定位
- (3) 關鍵點方向
- (4) 關鍵點描述

#### 2.4.1 尺度空間極值檢測

如 2.1.2 所言，特徵點在影像中含有較大資訊量，當原影像經過一系列的高斯卷積後，影像愈加模糊，由於特徵點不易受干擾的性質，且高斯核中心點權重為最大值，鄰近像素值較權重較小，進行多次加權平均後，相鄰的像素皆與目標像素差異逐漸增大，使得特徵點得以被保留。





圖十、自 GAN 所生成胚胎影像與真實胚胎影像之特徵點匹配

## 2.4.2 關鍵點定位

### 2.4.2.1 定位

由於數位影像係由離散的形式表現，求得的極值未必為真實極值，為確保吾人所篩選出的關鍵點皆為正確，首先利用尺度空間的泰勒展開式 (Taylor Expansion)，進行擬合，其擬合函數為

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X, X = (x, y, \sigma) \quad (12)$$

其中  $X$  可表示為與插值中心的偏移量，當  $X$  大於某一個吾人所設定的閾值 (Threshold)，意即該插值中心已偏離甚遠，此時需改變目前的關鍵點位置，並在新的位置進行擬合，如此反覆直到收斂為止。

### 2.4.2.2 去除不穩定極值

影像的邊緣亦會有極值出現，因此需要將其去除，由於圖像中的物體的邊緣位置的點的主曲率一般會比較高，因此我們可以通過主曲率來判斷該點是否在物體的邊緣位置。某像素點位置處的主曲率可以由二維的 Hessian 矩陣  $H$  計算得到

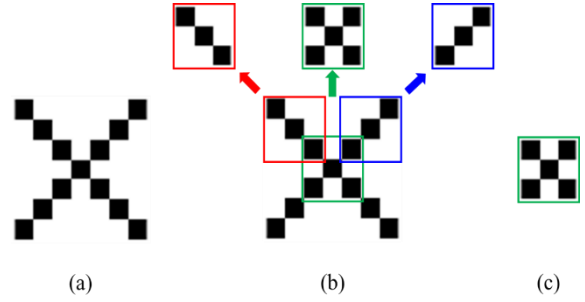
$$H = \begin{bmatrix} D_{xx}(x, y) & D_{xy}(x, y) \\ D_{yx}(x, y) & D_{yy}(x, y) \end{bmatrix} \quad (13)$$

若此矩陣有兩個特徵值  $\alpha$  及  $\beta$ ， $Tr(H)$  為  $H$  的對角元素和， $Det(H)$  為行列式值，則

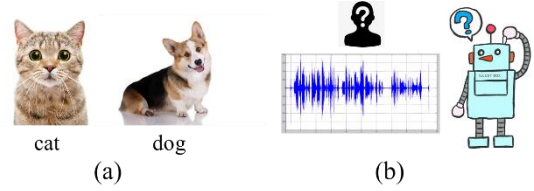
$$Tr(H) = \alpha + \beta \quad (14)$$

$$Det(H) = \alpha\beta \quad (15)$$

首先去除行列式為負的點，而後去除主曲率較大的點，可用以下規則去除



圖十一、(a)「X」圖形。(b) 經由卷積後得到「X」圖形之特徵圖。(c) 最符合「X」圖形之特徵圖。



圖十二、(a)影像分類應用於貓狗分類。(b)語音辨識應用於識別使用者。

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\gamma\beta + \beta)^2}{\gamma\beta^2} = \frac{(\gamma + 1)^2}{\gamma} \quad (16)$$

$\gamma$  越大，則該點有越大的機率位在影像邊緣上，因此需設置一閾值  $\gamma_0$ ，即

$$\frac{Tr(H)^2}{Det(H)} < \frac{(\gamma_0 + 1)^2}{\gamma_0} \quad (17)$$

### 2.4.3 關鍵點方向

為實現關鍵點的旋轉不變性，尚須納入關鍵點的角度，可由高斯尺度圖中得出，若高斯尺度為  $p$ ，則

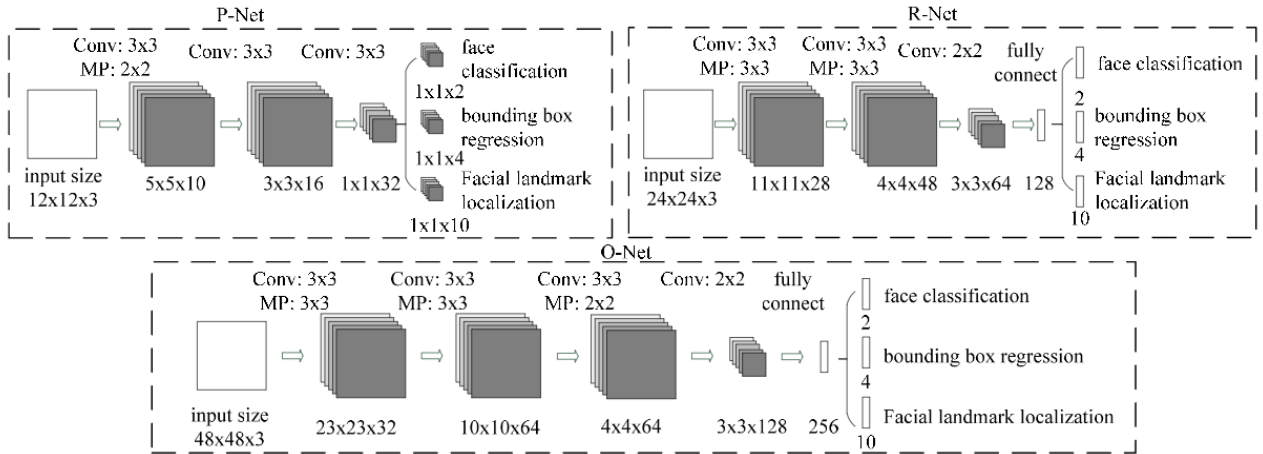
$$r = 3 \times 1.5p \quad (18)$$

而梯度振幅為

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (19)$$

梯度相位為

$$\theta(x, y) = \tan^{-1} \frac{(L(x, y+1) - L(x, y-1))}{(L(x+1, y) - L(x-1, y))} \quad (20)$$



圖十三、MTCNN 之架構圖，由 P-Net、R-Net 及 O-Net 組成，每一種網路輸入尺度皆不相同，由前者大致推論目標物件可能的位置，再將推論結果插值放大並向後傳遞，除了使輸入影像的感受野變大，也可提高篩選效率。[15]

最後利用直方圖對所有關鍵點做相位記錄，數位影像中則可使用角度離散化的方式分割成多個方向，建立好後再對直方圖  $h(i)$  進行平滑化，最終得到  $H(i)$

$$H(i) = \frac{h(i-2) + h(i+2)}{16} + \frac{4 \times h((i-1) + h(i+1))}{16} + \frac{6 \times h(i)}{16} \quad (21)$$

#### 2.4.4 關鍵點描述

經由上述步驟，吾人已得到關鍵點及其位置、相位，將關鍵點附近的區域劃分為  $d \times d$  個子區域，每一個子區域由 1 個種子及 8 個方向組成，為保證關鍵點具方向不變性，因此需將關鍵點及週遭區域進行旋轉，旋轉角度即為關鍵點相位，為使整個子區域旋轉，則旋轉半徑應為此子區域正方形對角線的一半，即

$$r = \frac{3\sigma(d+1)\sqrt{2}}{2} \quad (22)$$

像素點旋轉則使用旋轉矩陣進行，最後再使用三線性插值 (Trilinear Interpolation)，將像素點依序填入直方圖中。

SIFT 結果如圖十所示 (以 GAN 生成的胚胎細胞影像為例)。

### 3. 多尺度深度神經網路

近年由於人工智慧的興起以及科學技術的飛速發展，機器學習演算法幾乎滲透所有領域，

其中又以深度學習 (Deep Learning) 最為熱門，無論是學術抑或是各個應用領域，皆可看到其蹤影。

多尺度深度神經網路 [14] 架構，大致可分為兩大類：多尺度特徵學習 (Multiscale Feature Learning) 和多尺度特徵融合 (Multiscale Feature Fusion)，如圖所示。其優勢係來自於適應不同的任務複雜度，由於每次任務皆不盡相同，處理或辨識不同的訊號時，皆有其適合的尺度，舉例來說，圖十一(a) 為一「X」圖形，吾人可使用較簡易的方式，找到與該物件最匹配的特徵，如圖十一 (b)，最終找到此特徵，可利用推論目標物件是否為我們所尋找的「X」圖形，如圖十一 (c)。然而，近年來越來越多的任務異常複雜或困難，例如自然影像分類 (Image Classification)、自然語言處理 (Natural Language Processing) 等 (參見圖十二)，這些任務的資料本身特徵大量且複雜，以致於必須使用較細微尺度的特徵進行處理。因此，吾人可利用多尺度架構，提取出適當尺度的特徵來解決特定的任務。

#### 3.1 多尺度特徵學習

總括來說，多尺度特徵學習是藉由同時構建多個具有不同特徵尺度輸入與輸出的神經網路 (Neural Network) 模型，從而將多個模型的特徵組合並連接至同一全連接層，達到多尺度分析的效果。





- [12] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” In *International Journal of Computer Vision*, pp.1-28, 2004.
- [13] Lowe, David G, “Object recognition from local scale-invariant features,” In *Proceedings of the International Conference on Computer Vision 2*, pp. 1150–1157, 1999.
- [14] E. Elizar, M. A. Zulkifley, R. Muharar, M. H. M. Zaman, and S. M. Mustaza, “Review on Multiscale-Deep-Learning Applications,” In *Sensors*, 22.19: 7384, 2022.
- [15] X., Jia, and G. Zhu, “Joint face detection and facial expression recognition with MTCNN,” In *2017 4th international conference on information science and control engineering (ICISCE). IEEE*, 2017.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 770-778, 2016.

## 附錄

以下為建立多貝西小波轉換係數之步驟，以 MATLAB 作為範例。

$$(1) \text{ 建立 } P(y) = \sum_{k=0}^{p-1} C_k^{p-1+k} y^k$$

$$C_k^{p-1+k} = \frac{(p-1+k)!}{k!(p-1)!}$$

```
%% step 1: P(y)
P_y = zeros(2,p);
polynomial = ''; % 用於多項式檢查

% P(y)
for k = 0:p-1
    C = factorial(p-1+k) / factorial(k) / factorial(p-1);
    P_y(1,k+1) = C; % y係數C(p-1+k k)
    P_y(2,k+1) = k; % y幕次k
    % 多項式檢查
    if k == 0
        polynomial = [num2str(C), 'y^', num2str(k)];
    else
        polynomial = [polynomial, ' + ', num2str(C), 'y^', num2str(k)];
    end
end
end
```

圖七、步驟一，建立  $P(y)$

其中  $P_y()$  第一列用來表示各項  $y$  的係數，第二列用來表示各項  $y$  的幕次。  
Polynomial 用來可視化  $P(y)$  多項式。

$$(2) P_1(z) = P\left(\frac{2-z-z^{-1}}{4}\right)$$

$\frac{2-z-z^{-1}}{4}$  可寫作  $(0.5 - 0.25z - 0.25z^{-1})$   
 $(0.5 - 0.25z - 0.25z^{-1})^k$  可使用  $[-0.25, 0.5, -0.25]$  Convolution 得到。  
 各幕次計算後進行加總，得到  $P_1(z)$

```
%% Step 2: P1(y)
P1 = [-.25, -.5, -.25; %係數 -0.25z^-1 + 0.5z^0 - 0.25z^1
      -1, 0, 1]; %幕次 -1 0 1

P1_temp = P1(1,:);
P1_list = zeros(p,2*p-1); % 0矩陣 對齊各幕次產生的多項式

middle = int8(floor(length(P1_temp)/2)); % 算數項位置 位在list正中央
P1_list(2,p-middle:p+middle) = P1_temp; % 填入0矩陣 達到多的部分補0的效果

if p-1 > 1 %確認是否有次方 避免二次方多做一次卷積
    for k = 2:p-1 % k-1次方
        P1_temp = conv(P1(1,:),P1_temp);
        middle = int8(floor(length(P1_temp)/2)); % 修正算數項位置
        P1_list(k+1,p-middle:p+middle) = P1_temp; % 填入0矩陣 達到多的部分補0的效果
    end
end

P1_list(1,int8(length(P1_list)/2)) = 1; % 正中央為1 用於對齊算數項 以便後續可與係數直接相乘
p1oy_len = length(P1_list); % 用於多項式檢查

% 乘上P(y)係數
for i = 1:length(P_y)
    P1_list(i,:) = P1_list(i,:) * P_y(1,i);
end

% 各幕次相加 得到P1(z)
P1_sum = sum(P1_list);

% 多項式檢查
% P1_result = '';
% for i = 1:p1oy_len
%     P1_result = [P1_result, num2str(P1_sum(1,i)), 'z^', num2str(i-p), ' + '];
% end
%
% P1_result = P1_result(1:end-3);
```

圖八、步驟二，建立  $P_1(z)$

其中  $P1\_sum$  即為計算後的  $P_1(z)$  係數， $P1\_result$  用來可視化  $P_1(z)$  多項式。

(3) 計算  $z^k P_1(z)$  的根，可使用 MATLAB 的 roots 函式得到

```
%% Step 3: root
r = roots(P1_sum); % 求z^k×P1(z)解 以係數解即可
r_list = [];
```

圖九、步驟三，求出  $z^k P_1(z)$  解

$r$  即為  $P_1(z)$  所有的根 (包括複數根)。

(4) 得到根後，分別取絕對值，小於 1 者做為  $P_2(z) = (z - z_1)(z - z_2) \dots$  的

```
z1, z2, ...
%% step 4: P2(z)
for i = 1:length(r)
    if abs(r(i)) < 1 % 絕對值小於1 才存入根列表
        r_list = [r_list,r(i)];
    end
end

% P2(z)的(z-z1)(z-z2)...形式
for i = 1:length(r_list)
    g_z(i,1) = 1;
    g_z(i,2) = -1 * r_list(i);
end

% P2(z)展開
P2 = g_z(1,:);
if length(g_z(:,1)) >= 2 % 避免只有1個根的狀況進行多項式乘法運算
    for i = 2:length(g_z(:,1))
        P2 = conv(g_z(i,:),P2);
    end
end
```

圖十、步驟四，建立  $P_2(z)$

$P2$  即為  $(z - z_1)(z - z_2)$  展開後的形式

(5) 求出  $G_0(z) = (1 + z)^p P_2(z)$ ， $(1 + z)^p$  及  $P_2(z)$  可分別由 convolution 得到後，兩者再使用 convolution 相乘

```
g0[n] = Z^-1{G0(z)}
%% step 5: g0[n]
G_z = [1 1];
G_z_tmep = G_z;

% 求(1+z)^p的係數
for k = 1:p-1
    G_z_tmep = conv(G_z,G_z_tmep);
end
G_z_result = G_z_tmep;

% (1+z)^p × P2(z)
g0 = conv(G_z_result,P2);
```

圖十一、步驟五，求出  $g_0[n]$

$g_0$  則為最後求出的  $g_0[n]$ 。

- (6) 標準化  $g_1[n] = \frac{g_0[n]}{\|g_0\|}$ ，由於有複數問題，因此需取 real()，即  $g_1 = \text{real}(g_0 / \text{norm}(g_0))$ ;

```
%% step 6: Normalization
g1 = real(g0 / norm(g0));
```

圖十二、步驟六，標準化

- (7) 最後按照  $g[n] = g_1[-n]$ 、 $h[n] = (-1)^n g[2p-1-n]$ 、 $h_1[n] = h[-n]$  的規則求出  $g[n]$ 、 $h[n]$ 、 $g_1[n]$  及  $h_1[n]$

```
%% step 7: Time reverse
g = flip(g1); % g[n] = g1[-n]

% h[n] = (-1)^n * g[2p-1-n]
for i = 0:length(g)-1
    h(i+1) = (-1)^i * g(length(g) - i);
end

h1 = flip(h); % h1[n] = h[-n]
Daubechies_wavelet_list = [g; h; g1; h1];
```

圖十三、步驟七，求出  $g[n]$ 、 $h[n]$ 、 $g_1[n]$  及  $h_1[n]$

- (8) 最後依據 2p-point，輸入 p 值，即可得到(7)的 Daubechies wavelet list，包括正逆轉換的係數