

基於人類視覺敏感性的改良式資訊嵌入技術

潘哲倫* 周佳籐** 洪維恩*** 劉志威****

育達商業科技大學多媒體與遊戲發展科學系講師*

育達商業科技大學多媒體與遊戲發展科學系助理教授**

育達商業科技大學資訊管理研究所副教授***

育達商業科技大學資訊管理研究所研究生****

摘要

Lou 等學者在 2010 年提出了一個基於人類視覺特性的資訊嵌入技術 LCHVS。此方法將影像分成多個區塊，利用人類對於複雜區塊較不敏感的特性，在較複雜的區塊嵌入較大量的資訊，較平滑的區塊嵌入較少的資訊。此技術不但有相當好的載量，又考慮到人類視覺的特性，所以能得到相當好的影像品質。但是此方法嵌入資訊後可能會造成取出資訊上的錯誤，因此 Lou 等學者提出一個調整區塊像素的方法，來改進此問題，他們針對發生問題的區塊，依序取出像素進行調整。雖然此方法能讓取出資訊時不會發生錯誤，但是需要調整的像素較多，所以對影像的品質會有較大的破壞。本文提出一個方法，用以改進 Lou 等學者需要調整較多像素的問題。我們的方法是針對需要調整的區塊，尋找最需要調整的像素進行調整，如此一來我們不但有效地減少調整的像素，同時也能使影像品質有所提升。

關鍵字:人類視覺，LSB，OPAP，HVS

一、簡介

現今網路科技的蓬勃發展下，資訊的傳輸也日益頻繁，許多重要的資訊在網路上傳輸。為了防止資料遭竊，所以發展出各式各樣的密碼學。但是加密過的資料，同時也表示了該資料的重要性，使得有心人士更想去破解該文件的內容。基於不讓加密過的文件受到偵測，於是便發展出了資料藏匿(data hiding)的技術[1][2]。此技術是將資料隱藏在數位媒體中，而藏入資訊後的數位媒體也不容易被發覺有藏資料。因此將藏有資訊的媒體傳送至接收方，而接收到此媒體的一方再將資訊取出，便能安全的傳送資訊了。

Data hiding 過去有許多技術發展出來，其中最為人知的技術應是 LSB replacement[3]，LSB 是將機密資訊直接取代像素中較不重要的位元，來達到資訊隱藏的效果。此方法雖然能簡單且有效的嵌入資訊，但是對影像的破壞也較大。因此，基於修改 LSB 的方法也就相繼發展出來[4][5][6]。在 2004 年 Chan and Cheng 提出 Optimal pixel adjustment process(OPAP)的方法[7]，大幅改進了 LSB 的方法。此方法一樣是置換像素中較不重要的位元，但是對於沒更改的位元進行調整，使得調整後的像素值較接近原始像素。由於沒更動到嵌入的資訊，所以取出資訊時，只需要取出像素 LSB 的 bit 便可。OPAP 不但能有效的嵌入資訊，對影像的品質更是大幅的提升。

LSB 或 OPAP 都是對每個像素嵌入等量的資訊。嵌入資訊時對影像必會造成破壞，然而人類的視覺對影像敏感度並不相同。人類的視覺系統(human visual system, HVS)，對於一張圖片較凌亂的部份比較不敏感，平滑的地區比較敏感。如一張滿滿都是葉子的圖片中，少了一片葉子人類視覺很難察覺，但是在一張白紙上加了一個黑點卻很容易被發現。因此利用 HVS 概念而發展出不少資訊嵌入的方法[8]。HVS 較早的技術為 2003 年由 Wu and Tai 提出的 pixel-value differencing(PVD)[9]，此技術是將像素兩兩一組相減，當這兩個像素差值較大時，意味著該組像素經過影像的邊緣，也就是較複雜的地區。當兩個像素的差值較小，意味著該組像素為較平滑的地區。PVD 是利用 HVS 的特性，對於較複雜的地區較不敏感，因而嵌入較多的訊息，較平滑的地區便嵌入較少的訊息。不過 PVD 嵌入資訊前會參照事先定義好的 Range Table，用以決定該組像素差值所要嵌入的 bits 數 n ，然後將 n 個 bits 嵌入至這兩個像素中。PVD 的設定嵌入完的像素差值，參照 Range Table 能取得相同的 n 值，因此可以正確取出資訊。2008 年 Wang 等學者利用 PVD 的概念配合 modulus function(MF-PVD)[10]，該方法是運用 PVD 來決定這兩個像素可藏入的 bits 數，然後以 modulus function 計算出這個兩像素值的餘數，最後利用修改像素值的餘數來嵌入資訊。比起 Wu and Tsai 的方法，此方法可以保有更好的影像品質。在 2010 年 Lou 等學者提出了一個 Local Complexity and Human Vision Sensitivity (LC-HVS) 的資訊隱藏技術[11]。他們先將影像分為數個區塊，然後定義兩個閾值。接著計算區塊的變異數，當區塊的變異數大於閾值時，便將此區塊判斷為複雜地區，如果

小於閾值便判斷此區塊為平滑地區。接著將較複雜的區塊嵌入較多的訊息，較平滑的區塊嵌入較少的訊息。Lou 等學者利用 OPAP 的方法將資訊嵌入到影像中，如果嵌入後的區塊變異數與未嵌入區塊的變異數落在不同的閾值範圍時，則會造成取出資訊的錯誤。因此 Lou 等學者提出一個調整的方法來改進此問題。針對發生問題的區塊，將像素值依序取出進行調整，直到調整完的變異數與未嵌入資訊的變異數落在相同的閾值範圍為止。此方法不但能有效的嵌入資訊，且影像品質大幅提升，更考慮到 HVS 的概念，所以相較過去的方法更有效率且更好。但是 Lou 等學者調整區塊時，需要依序取出值進行調整，容易造成影像較大的破壞。本文提出一個新的方法，利用 Lou 等學者提出的方法嵌入資訊，可是 Lou 等學者提出的方法，在嵌入資訊後如果該區塊會有取出資訊錯誤時，需要對該區塊的像素依序調整，此方法需要調整較多的像素，才能保持取出資訊時不發生錯誤。而我們提出一個新的方法是找出該區塊最需要調整的像素進行調整。本研究的方法不但能有效地減少需要調整的像素，影像品質也能夠有所提升，後續本文將針對我們提出的方法進行詳細介紹。本文安排後續如下：第二節我們將對 Lou 等學者的方法進行詳細介紹。第三節我們將提出一個新的方法用以解決需要調整過多像素的問題。第四節則是實驗結果與討論。最後第五節提出我們的結論與看法。

二、文獻探討

本節中，我們將探討 OPAP 及 Lou 等學者提出的資訊嵌入法。我們所提出的方法將以這兩個方法為基礎進行資訊的嵌入。

(一) OPAP

Chan 等學者在 2004 年提出了 OPAP 的嵌入技術，這個技術改進了 LSB 的方法，使得影像的品質得以大幅提升。OPAP 的方法簡述如下：假設要嵌入資訊的像素為 p ，可以嵌入 k 個 bit 的資訊。而 p 的 rightmost k 個 LSBs 的十進位值為 $p^{(k)}$ ，以 LSB 嵌入 k 個 bit 的十進位值為 s ，嵌入完資訊後 p 會變成 p' 。OPAP 則利用下面公式調整 p' ，使得嵌入後的像素誤差值更小：

$$p'' = \begin{cases} p' + 2^k & p^{(k)} - s > 2^{k-1} \text{ and } p' + 2^k \leq 255, \\ p' - 2^k & p^{(k)} - s < -2^{k-1} \text{ and } p' - 2^k \geq 0, \\ p' & \text{otherwise,} \end{cases}$$

其中 p'' 是經由 OPAP 嵌入後的結果。但是 p'' 的 k 個 LSB 依然與 p' 相同，因此可以正確的取出嵌入的機密資訊。下面我們以一個簡單的範例來說明。設要嵌入資訊的像素值 $p = 96 = 01100000_2$ ，需要嵌入個 3 個 bit 值為 101_2 ，因此 $k = 3$ ， $s = 5$ ， $p^{(k)} = 0$ 。以 LSB 的方法嵌入後得到 $p' = 101$ 。由於 $p^{(k)} - s = 0 - 5 < -2^{3-1}$ ，因此 $p'' = p' - 2^k = 101 - 2^3 = 93$ ，93 的二進位值為 01011101_2 ，所以取出 93 的 3 個 LSB 便可將機密資訊 101_2 取出。

(二) LCHVS

Lou 等學者在 2010 年提出 local complexity and human vision sensitivity(LCHVS)，此技術利用人類視覺的特性，將影像平滑地區嵌入較少資訊，複雜的地區嵌入較多的資訊。LCHVS 先將影像分為多個區塊，計算區塊的變異數。將較複雜的區塊嵌入較多的訊息，較平滑地區嵌入較少的訊息。其步驟如下：

假設 8-bit 的原始影像 I ，大小為 $M \times N$ ，要嵌入的機密資訊為 S ，我們定義兩個閾值分別為 T_1, T_2 ，區塊大小為 $n \times n$ 。

Step 1 首先把原始影像 I 分成多個 $n \times n$ 大小的區塊為 $b_1, b_2, b_3, \dots, b_t$ ，其中

$$t = \lfloor (M \times N) / (n \times n) \rfloor。$$

Step 2 計算每個區塊的變異數為 v ，其中 v 的公式如下：

$$v = \sqrt{\frac{\sum_{j=1}^{n \times n} (b_{ij} - m_i)^2}{(n \times n - 1)}}$$

其中 m 為該區塊所有像素的平均值， j 為該區塊所有的像素。

Step 3 由 Step2 計算出變異數 v 之後，接著我們則需要判斷此區塊要嵌入的參數 k ，而 k 的公式如下：

$$k = \begin{cases} k = 1, & v < T_1 \\ k = 2, & v \geq T_1 \text{ and } v < T_2 \\ k = 3, & v \geq T_2 \end{cases}$$

Step 4 決定 k 值之後，我們從機密資訊 S 裡取出 $k \times n \times n$ 個嵌入值，利用 OPAP 將機密資訊嵌入到該區塊中。

Step 5 嵌入的區塊，我們需要用 Eq.1 重新計算新的變異數為 v' ，有了 v' 則利用 Eq.2 計算出取值參數 k' ，如果計算出來的 $k' \neq k$ ，則會造成取值的錯誤。因此需要用 Step 6 進行調整。

Step 6 因為 $k' \neq k$ 會造成取值的錯誤，所以需要針對此區塊進行調整，調整方法如下：

```

if  $k'_i < k_i$ 
    While  $k'_i \neq k_i$ 
        if  $b'_{ij} + 2^{k_i} \leq 255$  and  $b'_{ij} \geq m_i$ 
             $b'_{ij} = b'_{ij} + 2^{k_i}$ 
        else if  $b'_{ij} - 2^{k_i} \geq 0$  and  $b'_{ij} < m_i$ 
             $b'_{ij} = b'_{ij} - 2^{k_i}$ 
        end
        計算變異數 $v'$ 
        計算嵌入參數 $k'$ 
    else if  $k'_i > k_i$ 
        While  $k'_i \neq k_i$ 
            if  $b'_{ij} + 2^{k_i} \leq 255$  and  $b'_{ij} < m_i$ 
                 $b'_{ij} = b'_{ij} + 2^{k_i}$ 
            else if  $b'_{ij} - 2^{k_i} \geq 0$  and  $b'_{ij} \geq m_i$ 
                 $b'_{ij} = b'_{ij} - 2^{k_i}$ 
            end
            計算變異數 $v'$ 
            計算嵌入參數 $k'$ 
        end
    end
end
    
```

取出資訊時，我們只需要計算該區塊的變異數，決定取值參數 k' 之後，將每個像素 k' 個 LSB 取出便可。

下面我們舉一個簡單的範例作為說明。設原始影像 I 如圖 1-1 所示，我們定義兩個閾值分別為 $T_1 = 5$ ， $T_2 = 10$ ，區塊大小 $n = 4$ ，要嵌入的資訊 S 為 0000000101110101101101011111010。首先將原始影像 I 分為 $n \times n$ 大小的區塊，利用 eq.1 計算此區塊的變異數 v 為 5.5072，因為 $v \geq T_1$ and $v < T_2$ ，所以此區塊的嵌入參數 $k = 2$ ，每個像素將嵌入 2 個 bit。我們取出第一個像素值 $p = 24$ ，利用 OPAP 嵌入機密資訊 00，得到像素值 $p' = 24$ 。接著取出第二個像素值 $p = 19$ ，利用 OPAP 嵌入機密資訊 00，得到像素值 $p' = 20$ 。然後依序將值嵌入後得到的結果如圖 1-2 所示。接著需要計算取值參數是否等於嵌入參數，因此利用 Eq.1 計算 $v' = 4.9866$ ，因為 $v' < T_1$ ，所以嵌入參數 $k' = 1$ 。因為 $k' \neq k$ ，所以需要針對此區塊進行調整。首先我們計算該區塊的平均值 $m = 16.25$ ，然後利用 Step 6 調整完的結果如圖 1-3 所示。取值時，我們只需先計算此區塊 $v' = 5.9634$ ，然後判斷取值參數 $k' = 2$ ，這時只需要把該區塊所有像素的 2 個 LSB 取出，便能得到機密資訊 S 。

24 16 13 17	24 17 14 19	28 17 14 19
19 8 16 10	20 7 15 11	20 7 15 11
20 23 16 8	20 21 17 10	20 21 17 10
22 13 23 9	21 13 21 10	21 13 21 10
(1)	(2)	(3)

圖 1.(1)為原始影像，(2)為嵌入資訊後的影像，(3)為調整完的影像

LCHVS 可得非常好的影像品質，但是在調整區塊時，需要從第一個像素開始調整，直到該區塊 $k' = k$ 。此方法雖然可以讓取出的值不會發生錯誤，但是需要調整的值會較多，並且造成影像的品質降低。因此我們提出一個調整最佳值的方法來改進，依據新的方法只需調整較少的像素，對於影像的品質也能有所提升，我們將在下節詳細說明改進的方法。

三、提出的方法

本文我們提出一個基於 LCHVS 的嵌入方法，而嵌入值後的區塊可能會造成 $k' \neq k$ ，所以 Lou 等學者提出一個方法改進。此方法需要從區塊中的像素值依序調整，直到 $k' = k$ 為止。其缺點是需要更動較多的像素值，因此我們提出一個更改最佳值的改進方法，解決 LCHVS 需要更動較多像素的問題。我們的方法首先利用 LCHVS 的 Step 1~Step 5 嵌入資訊，而 Step 6 的調整方法則需用以下公式進行修改：

```

if  $k'_i < k_i$ 
    While  $k'_i \neq k_i$ 
         $p' = b'_{ij}(\min |b'_{ij} - m|)$ 
        if  $p' + 2^{k_i} \leq 255$  and  $p' \geq m_i$ 
             $p'' = p' + 2^{k_i}$ 
        else if  $p'_{ij} - 2^{k_i} \geq 0$  and  $p' < m_i$ 
             $p'' = p' - 2^{k_i}$ 
        end
        計算變異數  $v'$ 
        計算嵌入參數  $k'$ 
    else if  $k' > k$ 
        While  $k'_i \neq k_i$ 
             $p' = b'_{ij}(\min |b'_{ij} - m|)$ 
            if  $p' + 2^{k_i} \leq 255$  and  $p' < m_i$ 
                 $p'' = p' + 2^{k_i}$ 
            else if  $p' - 2^{k_i} \geq 0$  and  $p' \geq m_i$ 
                 $p'' = p' - 2^{k_i}$ 
            end
            計算變異數  $v'$ 
            計算嵌入參數  $k'$ 
        end
    end
end
    
```

由上式計算中，我們可以更改較少的像素達到 $k' = k$ 的效果。下面我們舉個簡單的範例近行說明。假設原始影像 I 如圖 2-1，我們定義兩個閾值分別為 $T_1 = 5$ ， $T_2 = 10$ ，區塊的大小為 $n = 4$ ，要嵌入的機密資訊 $S = [1\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 0]$ 。首先我們利用 Eq.1 計算出 $v = 4.7430$ ，因為 $v < T_1$ ，所以嵌入參數 $k = 1$ ，然後從機密資訊 S 中取出 $k \times n \times n$ 個 bit，利用 OPAP 將機密資訊嵌入到原始影像 I 中得到偽裝影像如圖 2-2 所示，接著計算偽裝影像的 $v' = 5.0183$ ，因為 $v' \geq T_1$ and $v' < T_2$ ，所以取值參數 $k' = 2$ ，由於 $k' > k$ 所以該區塊需要進行調整。首先計算所有像素的平均值 $m = 68.3750$ ，然後利用 Eq.3 計算出要調整的像素為 $p' = 56$ ，因為 $p' < m$ ，所以 $p'' = p' + 2^k = 56 + 2^1 = 58$ ，然後重新計算 $v' = 4.7046$ 與 $k' = 1$ ，此時 $k' = k$ ，便不需要再進行調整，而結果如圖 2-3 所示。而如果用 Lou 等學者的方法調整完的結果如圖 2-4 所示，由此可見我們提出的方法只需調整一個像素值，而 Lou 等學者的方法需要調整到五個像素。而我們調整的像素較少，所以得到的影像品質也較好。

69 76 72 68	69 76 72 69	69 76 72 69	67 74 72 69
71 70 72 67	70 71 73 66	70 71 73 66	68 71 73 66
69 72 66 59	68 73 67 59	68 73 67 59	70 73 67 59
70 68 67 57	70 68 67 56	70 68 67 58	68 68 67 56
(1)	(2)	(3)	(4)

圖 2.(1)為原始影像，(2)為嵌入資訊後的影像，(3)我們提出的方法

(4)為 Lou 等學者的方法。

四、實驗結果

本節將比較我們提出的方法，與 LCHVS 的方法進行比較，我們選用四張 8-bit 的灰階影像，大小皆為 512×512 ，分別為 Lena、Baboon、Sailboat 跟 Peppers，如圖 3 所示。嵌入的機密資訊都是以 pseudo random number generator 產生。我們使用 MSE 作為影像品質的評估標準。MSE 為原始影像與偽裝影像相減的平方差，設原始影像的大小為 $M \times N$ ，而 MSE 的計算得到的數值越小代表影像品質越好，其 MSE 的定義如下：

$$MSE = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (p_{ij} - p'_{ij})^2,$$

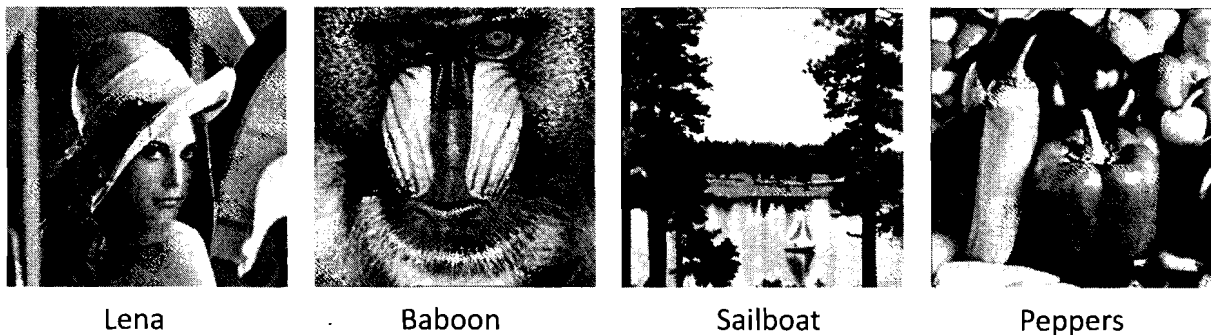


圖 3 選用的四張實驗影像

我們定義兩個閾值分別為 $T_1 = 3, T_2 = 6$ ，區塊大小為 $n = 4$ 。由於嵌入方法與變異數評估的方法皆相同下，我們提出的方法與 LCHVS 的載量將會相同，所以本文將不進行載量的討論。本文只針對需要調整的區塊計算我們提出的方法與 LCHVS 所需更改之像素的數目，與更改完的 MSE 進行比較。其結果如表 1。

表 1 選用 $T_1 = 3, T_2 = 6$ 嵌入資訊後的結果

cover image	LCHVS		Proposed	
	調整像素	MSE	調整像素	MSE
Lena	1,773	2.60	910	2.26
Baboon	556	4.82	328	4.49
Peppers	2,070	2.98	1,092	2.60
Sailboat	1,582	3.64	768	3.03

由上表可以看出我們提出的方法，不但更動的像素較少，而影像品質也較 LCHVS 來的好。因為我們針對最需要更改的像素值進行調整，與原先 LCHVS 需要由第一個像素依序調整的情況下，我們需要更改的像素較少，所以也能得較好的影像品質。下面我們將選用較小的閾值來進行嵌入的動作，我們選用 $T_1 = 2, T_2 = 3$ ，嵌入完的結果如表 2 所示。

表 2 選用 $T_1 = 2, T_2 = 3$ 嵌入資訊後的結果

cover image	LCHVS		Proposed	
	調整像素	MSE	調整像素	MSE
Lena	6,440	3.6078	2,032	1.9172
Baboon	341	6.0699	89	3.2278
Peppers	4,862	4.5710	1,352	2.2695
Sailboat	2,602	4.1528	778	2.1883

由上表看出，當選用較小範圍值嵌入時，需要調整的像素變多了，但唯獨 Baboon 這張需要調整的像素明顯的變少了，因為這張圖片較複雜的部份相對較多，所以計算出的變異數大多都已經超過 3 以上，因此嵌入完後變異數大多還是在 3 以上，相對的需要調整的區塊也就較少。而其它的影像都較平滑，所以計算出來的變異數相對容易在 2 到 3 之間，嵌入完後超出閾值的機率也就大多了，所以需要調整的區塊也會較多。下面我們將使用 Lena 這張圖片嵌入資訊，選用閾值為 $T_1 = 2, T_2 = 3$ 與 $T_1 = 3, T_2 = 6$ ，嵌入完資訊後，有相當多的區塊需要進行調整，我們將這些區塊用黑色標示出來，結果如圖 4 所示。



(1) $T_1 = 2, T_2 = 3$



(2) $T_1 = 3, T_2 = 6$

圖 4 (1)為閾值 $T_1 = 2, T_2 = 3$ ，(2)為閾值 $T_1 = 3, T_2 = 6$

由圖 4 我們可以清楚發現，當我們選用的閾值較小時，Lena 這張圖需要調整的區塊明顯比較多。當閾值較大時，需要調整的區塊卻變少了。原因是 Lena 這張圖片相對的平滑，所以每個區塊算出來變異數相對的也較小，所以嵌入之後會發生需要調整的區塊也變多了，所以圖 4(1)才會出現較多需要調整的區塊，而圖 4(2)相對的比較少了

五、結論

本文我們提出一個基於 LCHVS 的改進方法。LCHVS 是利用人類視覺的原理，將影像較複雜的區塊嵌入較多的資訊，較平滑的區塊嵌入較少的資訊。用此方法可以讓嵌入資訊後的影像，不容易被人類的視覺察覺出。而 LCHVS 嵌入資訊後的區塊，可能會因為變異數的差異，造成取出資訊時的錯誤，所以 Lou 等學者提出一個方法改進此問題。他們提出的方法是將此區塊的像素依序調整，直到調整完的變異數能夠正確計算出取值的參數為止。可是他們的方法需要更改較多的像素才能達到取值的正確，因此我們提出一個方法，採用計算區塊中最需要被更改的像素進行調整，直到能夠正確計算出取值的參數為止。由實驗結果也證明了，本研究的方法不但能有效地減少像素被更改的次數，同時也能得到較好的影像品質。

六、引用文獻

1. Hong, W., Chen, T.S., and Shiu, C.W., "A Minimal Euclidean Distance Searching Technique for Sudoku Steganography," in Proceedings of International Symposium on Information Science and Engineering, vol. 1, pp. 515-518, 2008.
2. Provos, N. and Honeyman, P., "Hide and Seek: An Introduction to Steganography," IEEE Security and Privacy Magazine, vol. 1, no. 3, pp. 32-44, 2003.
3. Cheddad, A., Condell, J., Curran, K., and Kevitt, P.M., "Digital Image Steganography: Survey and Analysis of Current Methods," Signal Processing, vol. 90, no. 3, pp. 727-752, 2010
4. R.Z. Wang, C.F. Lin, and J.C. Lin, "Image Hiding by Optimeal LSB Substitution and Genetic Algorithm," Pattern Recognition, vol. 34, pp. 671-683, 2001.
5. C.C Chang, J.Y.Hsiao, and C.S. Chan, "Finding Optimal Least-Significant-Bit Substitution in Image Hiding by Dynamic Programming Strategy," Pattern Recognition, vol. 36, on. 7, pp.1583-1595, 2003.
6. J. Mielikainen, "LSB Matching Revisited," IEEE Signal Processing Letters, vol. 13, on. 5, pp.285-287, 2006.
7. Chan, C.K. and Cheng, L.M., "Hiding Data in Images by Simple LSB Substitution," Pattern Recognition, vol. 37, no. 3, pp. 469-474, 2004.
8. Yang, C.H., Weng, C.Y., Wang, S.J., and Sun, H.M., "Adaptive Data Hiding in Edge Areas of Images with Spatial LSB Domain Systems," IEEE Transactions on Information Forensics and Security, vol. 3, no. 3, pp. 488-497, 2008.
9. D.C Wu and W.H. Tsai, "A Stegoanographic Method for Images by Pixel-Value Differencing," Pattern Recognition Letters, vol. 24, pp.1613-1626, 2003
10. C.M. Wang, N.I Wu, C.S Tsai, and M.S. Hwang, "A High Quality Steganographic Method with Pixel-Value Differencing and Modulus Function," Journal of Systems and Software, vol. 81, on. 1, pp.150-158, 2008.
11. D.C. Lou, N.I. Wu, C.M. Wang, Z.H. Lin, C.S. Tsai, "A Novel Adaptive Steganography Based on Local Complexity and Human Vision Sensitivity," Journal of Systems and Software, vol. 83 pp.1236-1248