

● [Training]

這裡我將所有的training data 切成 pixels * pixels (pixels的大小有測試 8 與 12), 且 pixel steps 為 2 , 放入 blkMatrix , 照此步驟即可以得到令名方式為 Dic_(algorithm)_(pixels)_(plants or wood)

Dic_MOD_8_plants	Dic_MOD_8_wood	Dic_MOD_12_plants	Dic_MOD_12_wood
Dic_KSVD_8_plants	Dic_KSVD_8_wood	Dic_KSVD_12_plants	Dic_KSVD_12_wood

以上的 dictionary 都有附在檔案裡

因為 pixels 有 8 與 12 兩種,

所以 pixels 為 8 的 dictionary 大小及為 64 * 128

pixels 為 12 的 dictionary 大小及為 144 * 288

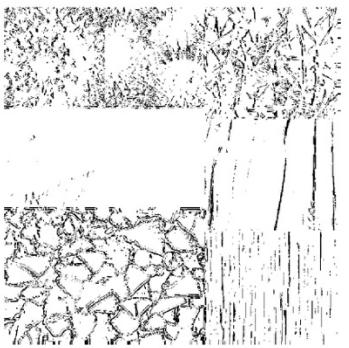
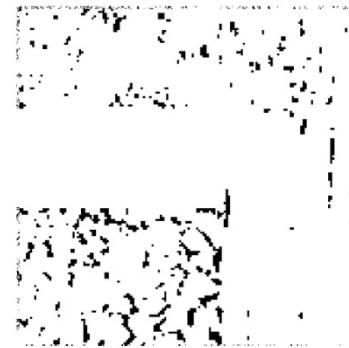
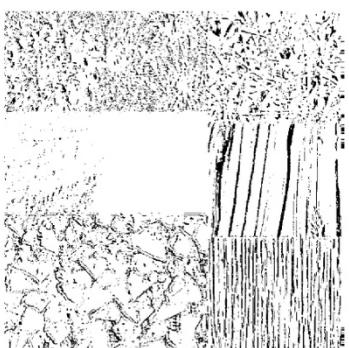
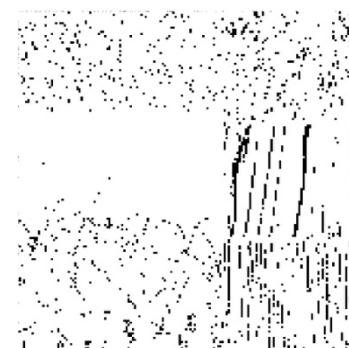
● [Testing]

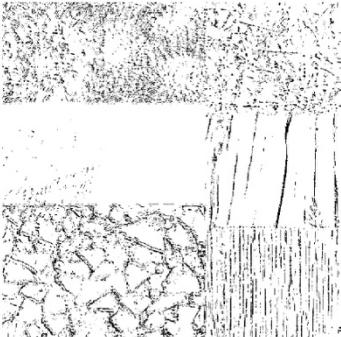
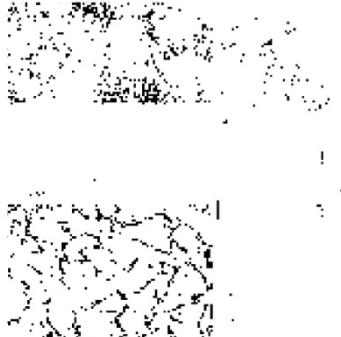
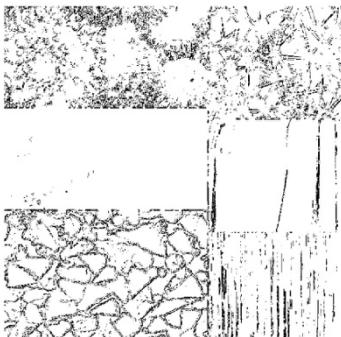
將圖片切成 pixels * pixels (8 or 12) , 之後再把 dictionary(wood,plants) 與切好的 test picture 做 OMP 即可求出係數 Coefw 和 Coefp , 把 係數與原先的 dictionary 做相乘 , 比較大小 , 將此分辨成 wood 或 plants

● [After Testing]

1. Majority vote: 將圖片依 “固定區塊” 統計 wood 與 plants 之點數 , 若 wood 點數大於一 “設定值” 則將此點判別為 wood , plants 也是此種方式
 - a. “固定區塊” 和 “設定值” 在後面會有調整的方式
2. MedianFilter: 將圖片利用 MedianFilter 消除雜訊雜點 , 以利判讀

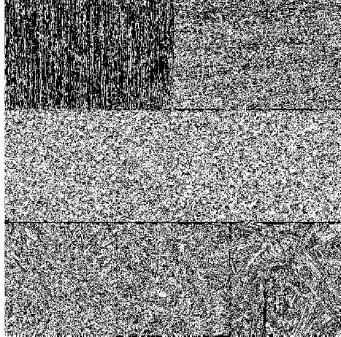
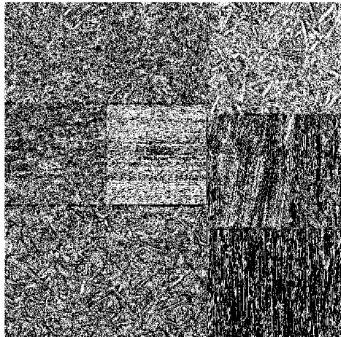
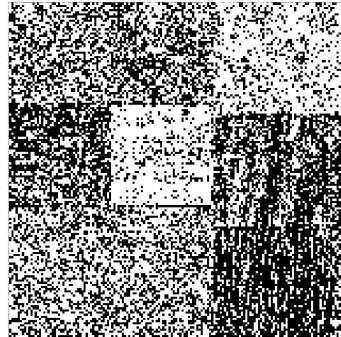
- [test0.jpg] MOD
- (a,b) --> a 代表 (pixel*pixel) 的 pixel
b 代表 OMP 中的 errt
- 次數為 Majority vote 做的次數,
區塊邊長 為 vote 的區塊邊長
設定值 為大於多少白點及為白色
- testing 完“很好”或“很差”者就不做 Majority vote
- 由以下圖片推論 MOD(12,8), MOD(8,2)可能為 MOD 中的最佳解

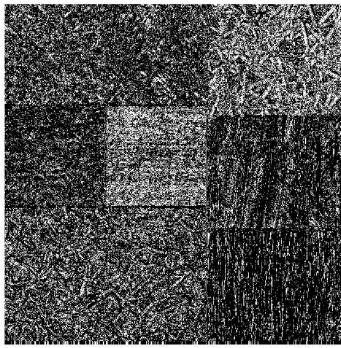
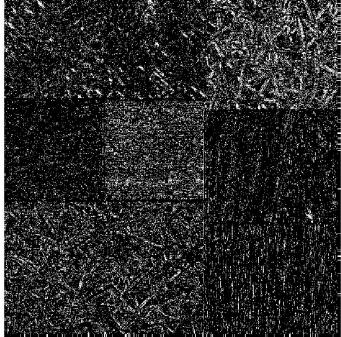
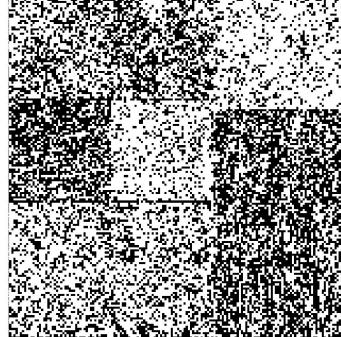
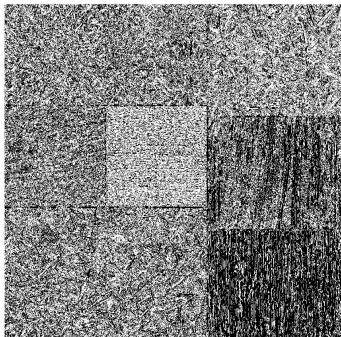
MOD(12,1)	次數: 4 區塊邊長: 2 設定值: 10
	
MOD(12,2)	次數: 4 區塊邊長: 2 設定值的: 10
	
MOD(12,4)	次數: 4 區塊邊長: 8 設定值的: 230

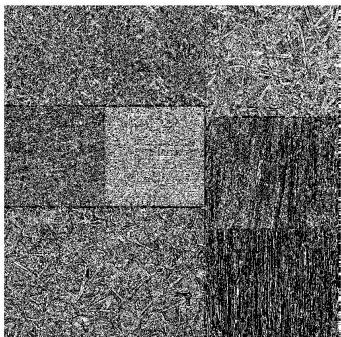
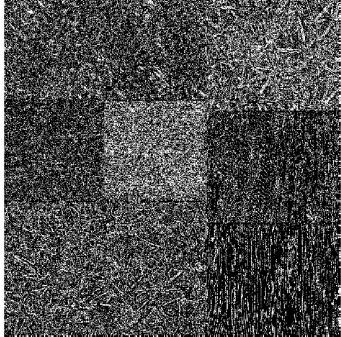
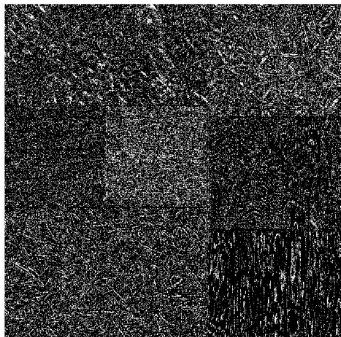
	
MOD(12,8)	次數: 1 區塊邊長: 2 設定值的: 22
	
MOD(8,1)	次數: 區塊邊長: 設定值的:
	
MOD(8,2)	次數: 區塊邊長: 設定值的:

	
MOD(8,4)	次數: 100 區塊邊長: 2 設定值的: 24
	
MOD(8,8)	次數: 區塊邊長: 設定值的:
	

- [test0.jpg] KSVD
- (a,b) --> a 代表 (pixel*pixel) 的 pixel
b 代表 OMP 中的 errt
- 次數為 Majority vote 做的次數,
區塊邊長 為 vote 的區塊邊長
設定值 為大於多少白點及為白色
- testing 完很差者就不做 Majority vote
- 由以下圖片推論 KSVD(12,1) 可能為 KSVD 中的最佳解

KSVD(12,1)	次數: 50 區塊邊長: 2 設定值: 12
	
KSVD(12,2)	次數: 50 區塊邊長: 2 設定值的: 5
	
KSVD(12,4)	次數: 區塊邊長: 設定值的:

	
KSVD (12,8)	次數: 50 區塊邊長: 2 設定值的: 1
	
KSVD (8,1)	次數: 區塊邊長: 設定值的:
	
KSVD (8,2)	次數: 區塊邊長: 設定值的:

	
KSVD (8,4)	次數: 100 區塊邊長: 2 設定值的: 24
	
KSVD(8,8)	次數: 區塊邊長: 設定值的:
	

- [利用以上論點實證 test1.jpg]

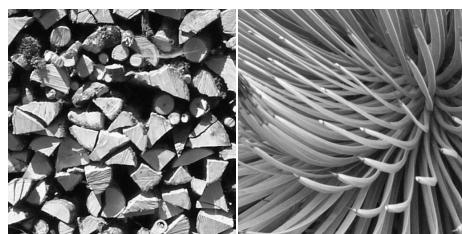


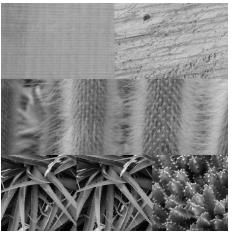
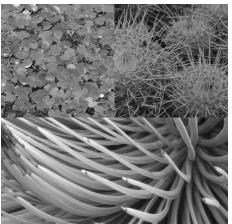
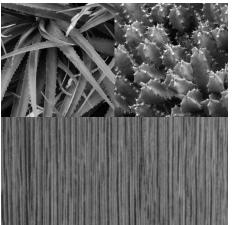
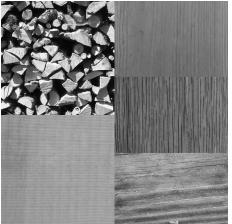
test1.jpg

picture	test_1_MOD_8_2	test_1_MOD_12_8	test_1_KSVD_12_1
test			
vote	不需要	不需要	
filter			

[結論]

對於 KSVD 與 MOD 兩種演算法，我實做出來 MOD 的完成度較高，判斷出來的也較正確，甚至也不約要做太多的後處理(例如我做的 vote 和 filter)，以下為我將所有 test.img 都實作出 MOD 的結果，有些圖可能需要在後處理，不過大致都可以判讀，然後誤判率最高的圖片為下兩圖



Test2		
Test3		
Test4		
Test5		
Test6		