

作業二其實不難
有通過的同學應該知道

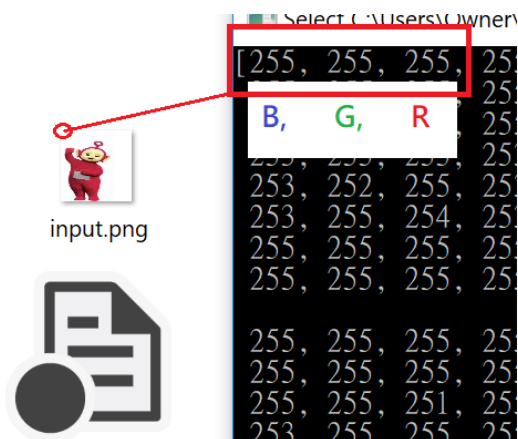
以範例case1為例

imread 讀進 argv[1] 後

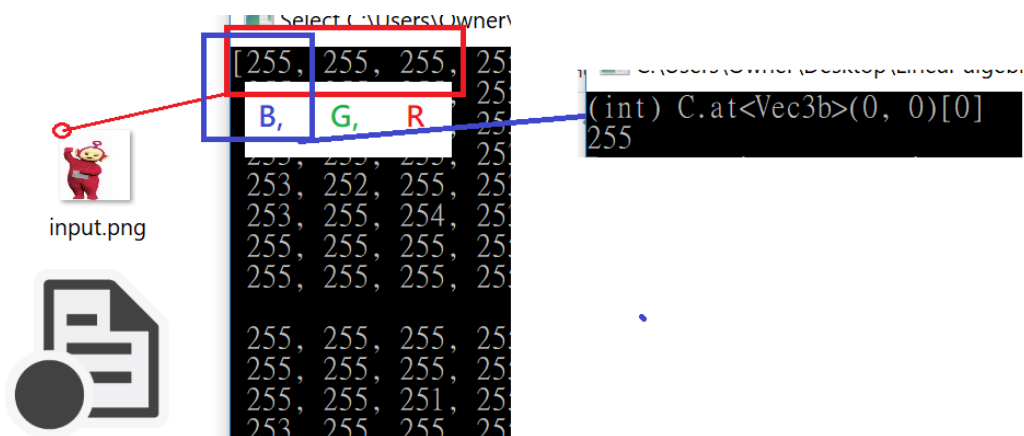
你可以直接將 mat cout 出來

他就是全圖的 BGR 分布 map

output 輸出時也可以用這個方法確認輸出是否正確



接著 需要把BGR分割出來分別做小波轉換



像是 `(int) C.at<Vec3b>(0, 0)[0]`

或是

`Vec3b& bgr = C.at<Vec3b>(0, 0)`

`cout<<(int) bgr[0]`

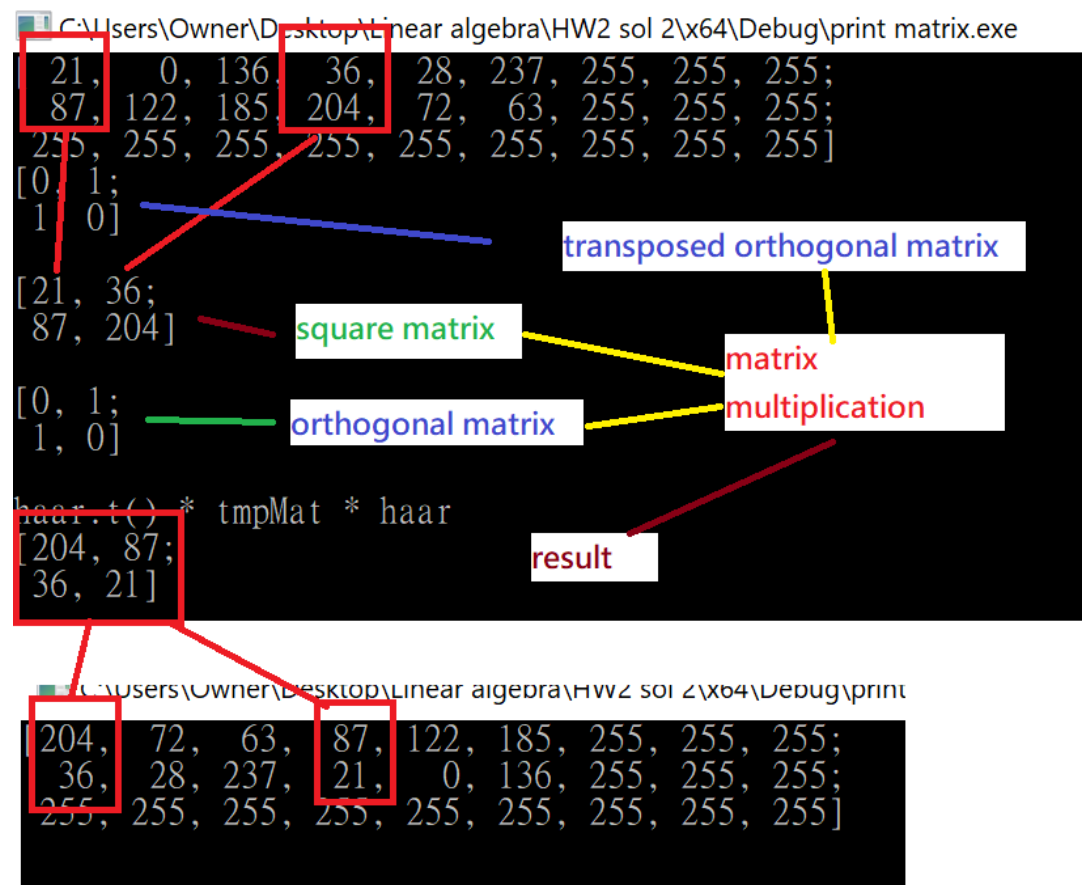
都可以輸出第一個顏色通道中第一個pixel的值

而 `image.rows` 與 `image.cols` 可以輸出影像行列多大

基本上做到這裡已經可以開始做小波轉換了

把他當成一般矩陣相乘

(或是你對opencv熟的話可以直接把他轉成ROI方式直接把圖片相乘，超作弊的☺)



The screenshot shows a terminal window with the following content:

```
C:\Users\Owner\Desktop\Linear algebra\HW2 sol 2\x64\Debug\print matrix.exe
[ 21,  0, 136,  36,  28, 237, 255, 255, 255;
 87, 122, 185, 204,  72,  63, 255, 255, 255;
255, 255, 255, 255, 255, 255, 255, 255, 255]
[0, 1;
 1, 0]
[ 21, 36;
 87, 204]
[0, 1;
 1, 0]
haar.t() * tmpMat * haar
[204, 87;
 36, 21]
```

Annotations in the image:

- transposed orthogonal matrix**: Points to the $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ matrix.
- square matrix**: Points to the $\begin{bmatrix} 21 & 36 \\ 87 & 204 \end{bmatrix}$ matrix.
- orthogonal matrix**: Points to the $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ matrix.
- matrix multiplication**: Points to the operation $\text{haar.t()} * \text{tmpMat} * \text{haar}$.
- result**: Points to the resulting matrix $\begin{bmatrix} 204 & 87 \\ 36 & 21 \end{bmatrix}$.

Below the first screenshot, the result matrix is shown again in a separate terminal window:

```
C:\Users\Owner\Desktop\Linear algebra\HW2 sol 2\x64\Debug\print
[204, 72, 63, 87, 122, 185, 255, 255, 255;
 36, 28, 237, 21,  0, 136, 255, 255, 255;
255, 255, 255, 255, 255, 255, 255, 255, 255]
```

最後跟上面的方法一樣

```
Vec3b& bgr = C.at<Vec3b>(j, i);
```

```
bgr[0] = xxx;
```

一個通道的數值就被改掉了

最後輸出 `imwrite(argv[3], C)` 即可

通常>0 分有問題

可能是你的方陣寫死(不一定是 2×2 ，如 8×8 就逐次需用 3 個 8×8 的方陣相乘)

或是浮點數的問題 (需用 `double` 運算，注意題目的規則)

#作業二不到一半的人通過

#會不會被當阿 還要期中考

#助教好難當 ☹