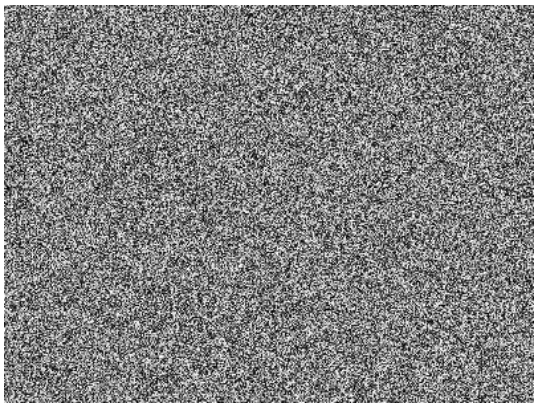
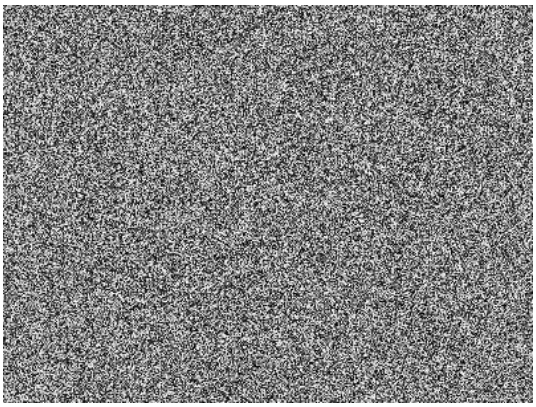
PART A

Img_key1

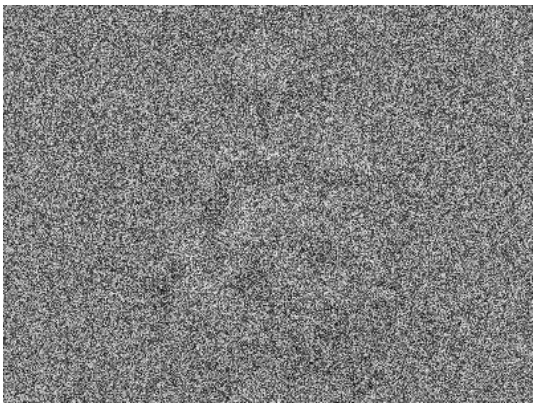


Img_key2

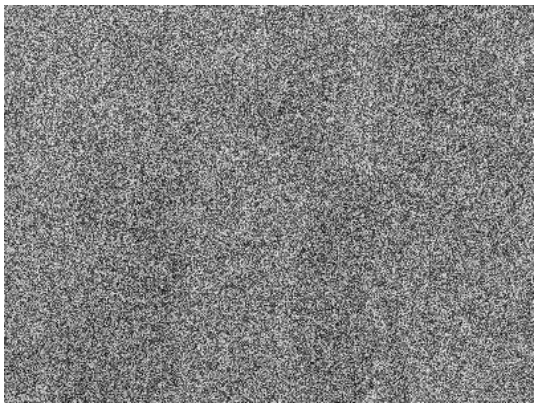


Img_target

Img_E



Img_Eprime



Final

Epoch =69

W 用 random 設 3 個數字

Rate = 1e-8

PART B

MaxlterLimt = > 400*300 (原圖片大小)

α=> Learging Rate

£ => 原為限制 Gradient Desent 訓練的距離 但程式未使用到

PART C

我先定義 W 為三隨機數字 (根據 pdf 的 step2)

後根據演算法計算出最後的 key 值

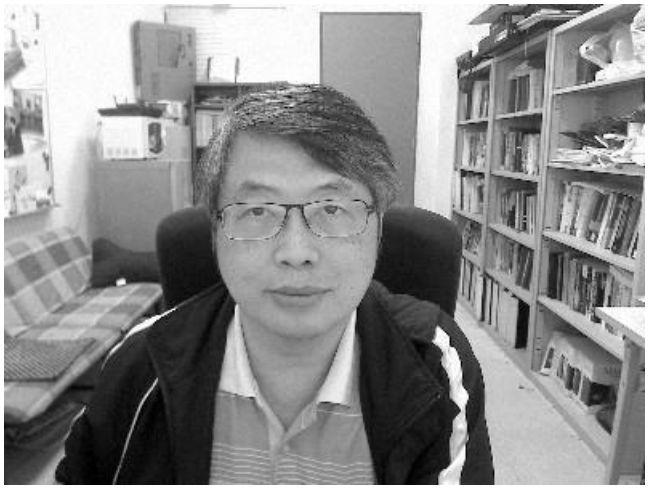
```
epoch-> 68 error=> 0.06644934488731735  W=> [0.24988744608542962, 0.6599076314921408, 0.09023009726282688]
```

PART D

$$I = \frac{E - w_1 K_1 - w_2 K_2}{w_3}.$$

根據公式

最後得出下圖



PART E

我原本沒有去看 READ.MD 去使用了 Cv2 去開啟圖片並轉呈灰階執行
去印出圖片的 type 之後
把 長寬 定義成 400 和 300
但是做到後面時發現存出去的檔案一直出現問題
後來上去看之後才發現要用 PIL 去做
把前面開啟圖片的方式修改
中間改成 getpixel 去執行後
最後出來的圖片終於是正確的
(原本上半會整片模糊 但不知道原因　可能是有參數弄錯也說不定　但是更改得太快　所以沒有去做測試)

另外仍不熟悉 Github Commit 的操作
可能是一開始設定的東西不對
每次要上傳的時候總是找不到我的檔案
把檔案單方面上傳上去好像會到 GitList
但發現的時候幾乎是已經尾聲了……
點下去也沒有 History 可以顯示
這點十分尷尬且抱歉
除此之外也不太熟悉直接打出 READ.MD 的方式　於是改作了 PDF
下次作業應該會嘗試使用.MD 的方式去做撰寫　以學習新的技巧

PART F

在實作之前並不太了解Single Neural Network
在詢問室友以及觀看了兩部YouTube上的講解後
大概理解的他的方式
在不斷地用data去餵食（像這次的作業共有400*300筆 ）
透過一個learning rate去計算
並去和原本的E 去做比對計算誤差
之後修正W0 W1 W2
得出最相近的KEY之後
回去利用公式解碼出最後的圖片
過程還蠻有趣的
因為很難想像訓練的過程是怎麼計算的
不過看到最後結果有嚇到一下　還算有趣的一次作業