

# 瑕疵檢測

409410197 劉柏辰 409416541 周騏軍

# 目錄

01

用不同filter和offset偵測瑕疵

03

把選定好的filter合併

02

選定各filter可用的數值

04

針對彼此影響的filter進行改善

# 前言

我們有嘗試過使用autoencoder，但因為網路上的資料不夠多，加上老師的上課時間有限，所以也沒講清楚（到底哪個dir 要放甚麼），並且autoencoder需要大量的計算，以我們電腦的能力跑不太動，故改採用statistics的方法。



The background is a deep blue with a complex, abstract pattern. A prominent feature is a glowing, circular ring composed of many small, bright blue particles or dots, which appears to be the focal point of the image. Surrounding this ring are various wispy, smoke-like or particle-like structures that trail off into the background, creating a sense of motion and depth. The overall aesthetic is futuristic and technological.

用不同filter和  
offset大小偵測瑕疵

# 如何實現

## 關於offset

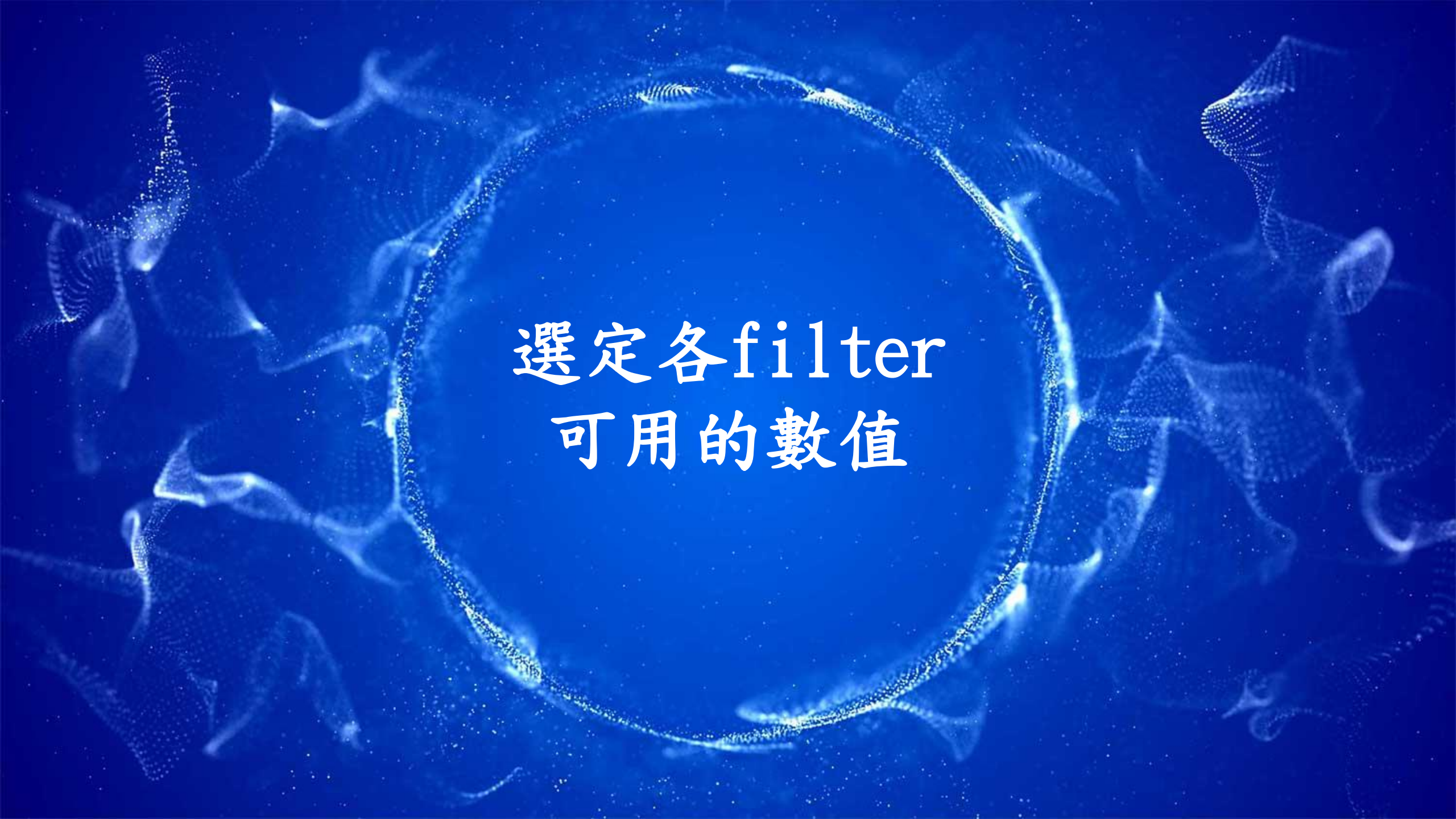
用offset來切割圖片大小，使圖片切割成適合的等分，以便我們能用我們選的算法（Correlation, Contrast, Uniformity）。

## 關於Filter

不一樣的filter會偵測出不同的瑕疵影像，所以在最一開始我們先試著看看，哪個filter大小較符合我們的需求。

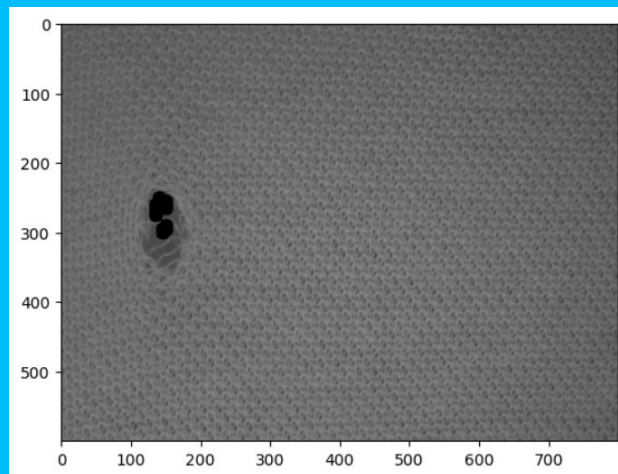




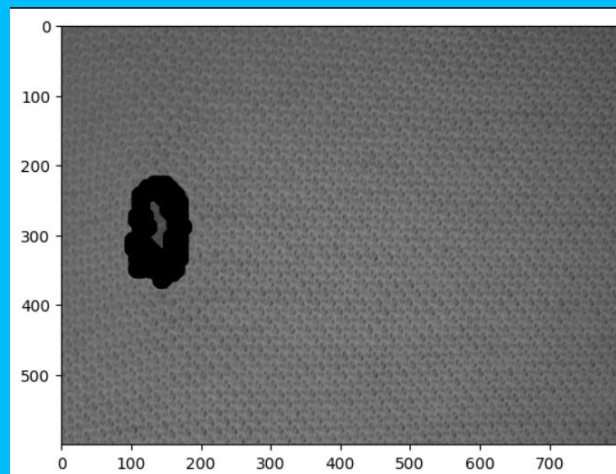
The background is a deep blue with a complex, abstract pattern. A prominent feature is a glowing, circular ring composed of many small, bright blue particles or dots, which appears to be in motion or vibrating. Surrounding this ring are various wispy, smoke-like or particle-like structures that also seem to be made of fine blue dots, creating a sense of dynamic energy and depth.

選定各filter  
可用的數值

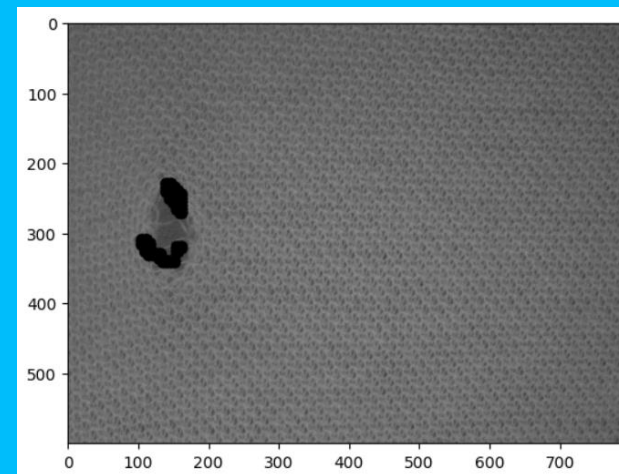
# 圓點Filter



Filter = 15  
Offset = 5



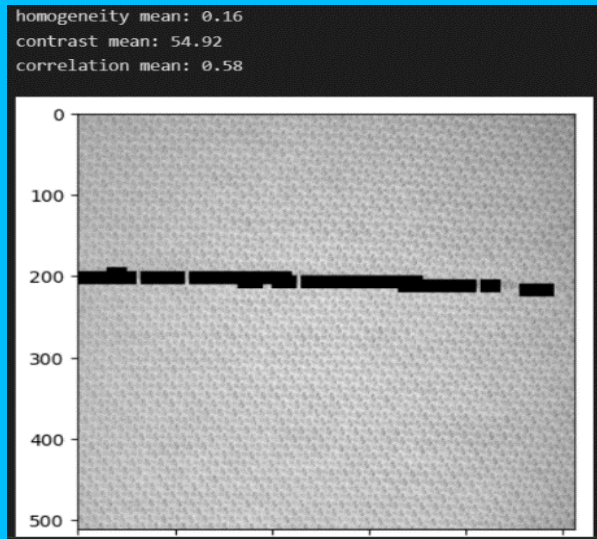
Filter = 23  
Offset = 5



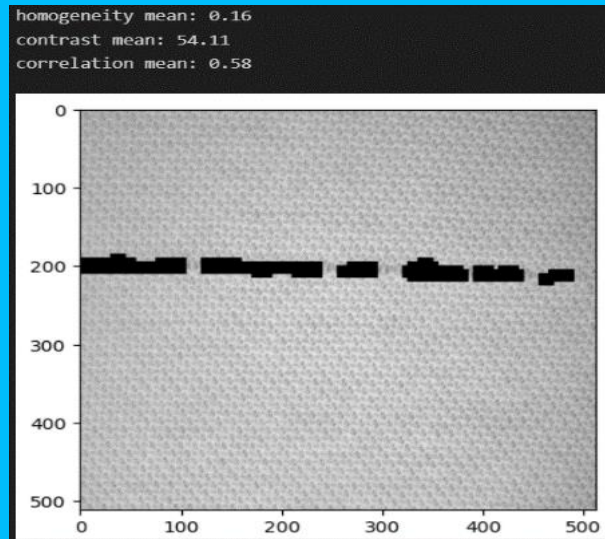
Filter = 33  
Offset = 5

在嘗試過哪個filter較適合我們後，緊接著我們就要調整我們選的算法（Correlation, Contrast, Uniformity）的值，以利偵測瑕疵更為精準。  
由於每個算法套用在不同的filter裡時，都會有不同結果，  
所以我們需要一個一個試，讓偵測出來的瑕疵盡可能的符合我們的需求。

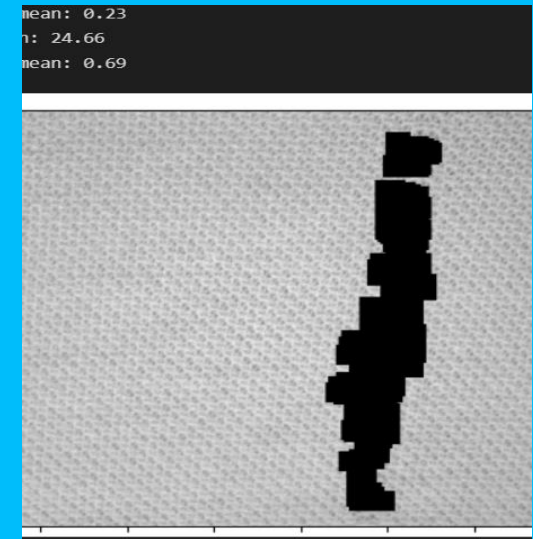
# 直線Filter



Filter = 17  
Offset = 5



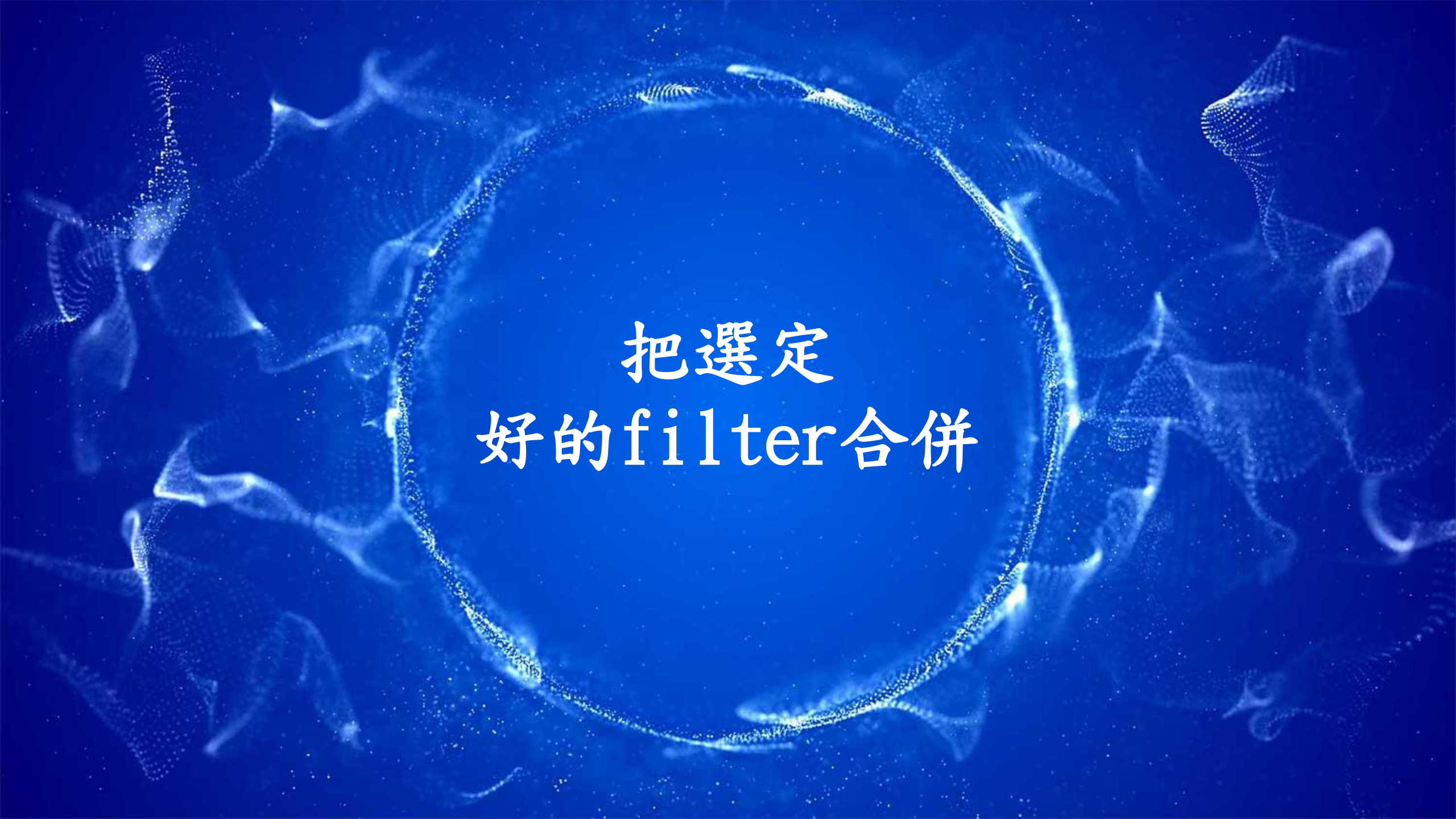
Filter = 22  
Offset = 5



Filter = 40  
Offset = 3

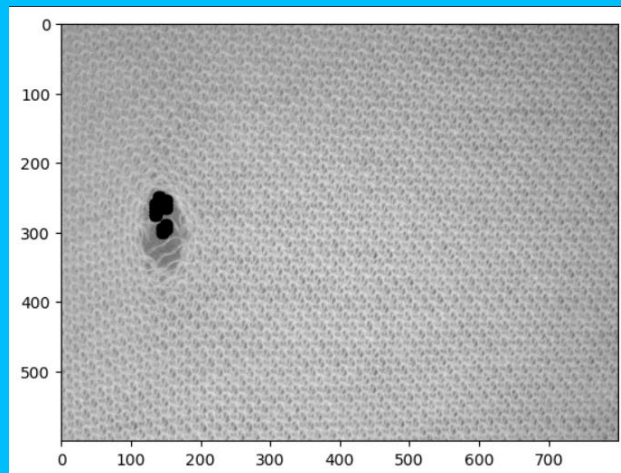
就如同圓點filter一樣，我們將選定好的filter與offset之算法的參數進行調整，以利我們能將測試結果逼近老師給的預期解答，但在多次的修正與改良後，我們發現直線圖的右方一直無法被偵測為缺陷，且經由人工確認之後，也無法有效判別是否為缺陷，故在人工及程式都無法辨識的情況下，我們就將它視為非缺陷的部分。



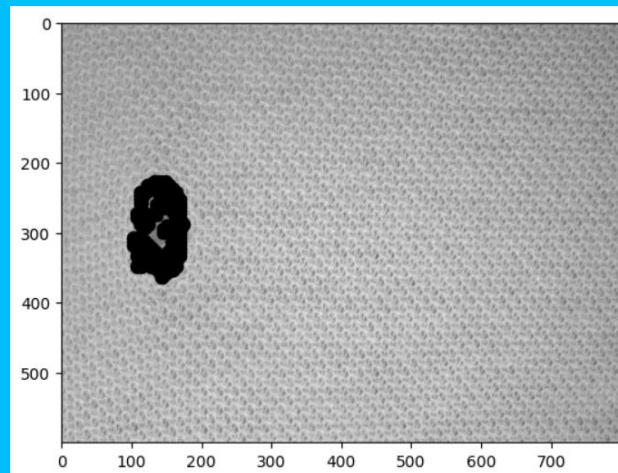
The background is a deep blue with a complex, abstract pattern. A prominent feature is a glowing, circular ring composed of many small, bright blue particles or dots, which appears to be the focal point of the composition. Surrounding this ring are various wispy, smoke-like or nebula-like structures in shades of blue and white, creating a sense of depth and movement. The overall aesthetic is futuristic and scientific.

把選定  
好的filter合併

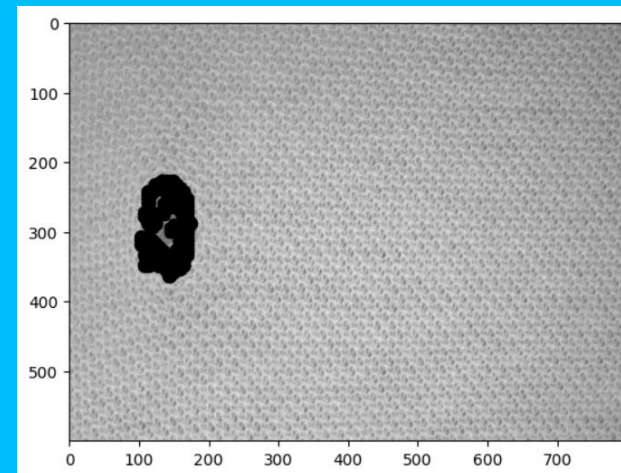
# 合併圓點Filter



Filter = 15  
Offset = 5



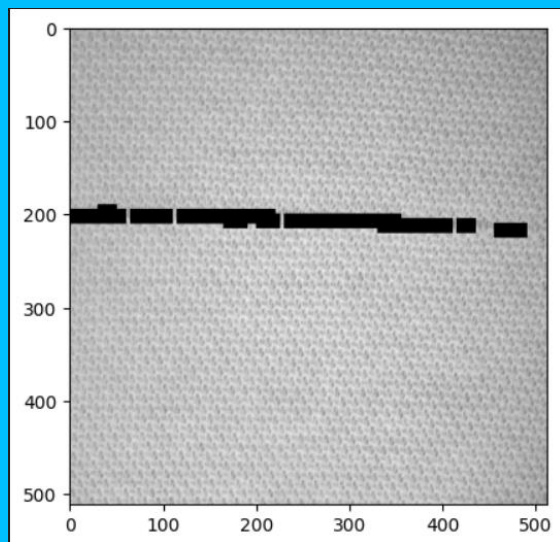
Filter = 15+23  
Offset = 5



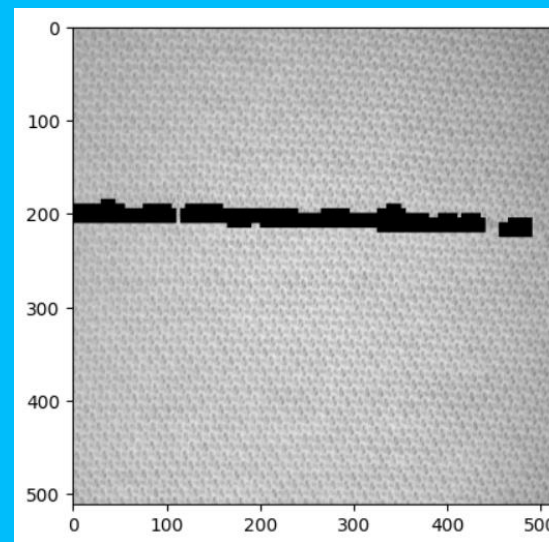
Filter = 15+23+33  
Offset = 5

再選定好filter與offset後，我們將其算法的參數做了調整，  
讓每一個filter都能接近老師要的結果，緊接著，我們就把它們合併，  
因為每個filter都有些偵測不到的地方，但偵測不到的地方在其他filter下又可以偵測，  
所以我們就利用一點小技巧把它們合併起來，使它們能更完整。

# 合併直線Filter



Filter = 17  
Offset = 5



Filter = 17+22  
Offset = 5

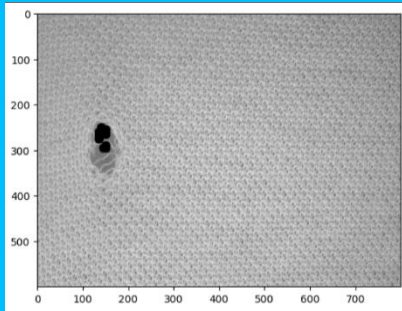
直線的合併與原點一樣，故不在贅述，上頁中有提到我們使用了小技巧將其合併，  
因此，這頁就來講解一下是什麼技巧能將不同filter合併，  
首先將一個filter為底，接著再套用其他filter時就判斷一下是否被偵測過，  
如果有的話就不偵測同一點，如果沒有就偵測看看，  
這樣就能將不同的filter所產生的結果合併。



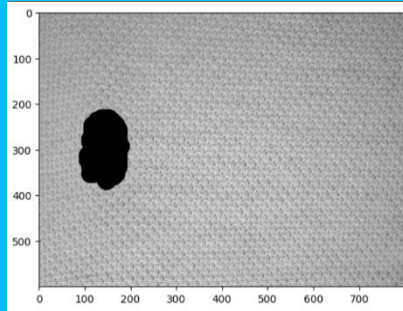
The background is a deep blue with a complex, abstract pattern. A prominent feature is a glowing, circular ring composed of many small, bright blue particles or dots, which appears to be in motion or vibrating. Surrounding this ring are various wispy, smoke-like or particle-like structures that also seem to be made of fine blue dots, creating a sense of dynamic energy and depth.

針對彼此影響  
的filter進行改善

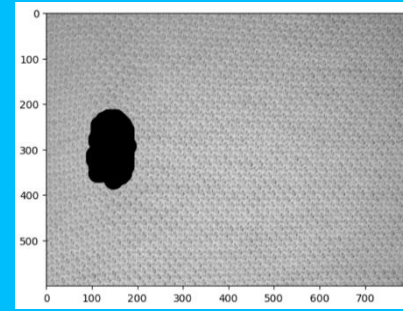
# 細節修復



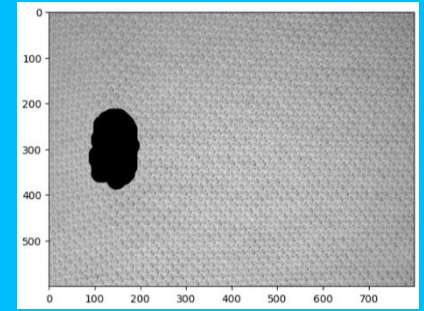
Filter = 15  
Offset = 5



Filter = 15+23  
Offset = 5



Filter = 15+23+33  
Offset = 5



Filter = 再+40  
Offset = 5

因為不同的filter 會互相影響，  
例如filter size 17 的會將img154 非瑕疵部分給偵測錯誤，  
因而新增了限制以防止此事件之發生，  
利用filter size 15 能判讀出img154 的特性對filter size 23 的限制進行調正，  
但其代價為img154 cover到的範圍會過大。

# 限制概述

```
# Detect the hole edge
# Test by image2 and img154 from texture_video
case 23:
    if co_correlation[i,j] > 0.809:
        if co_contrast[i,j] > 35:
            if co_homogeneity[i,j] < 0.31:
                if(count[15] > 15):
                    for k in range(0, 10):
                        if(has_print):
                            break

                        for l in range(0, 10):
                            if(new_img[i+k, j+l] == 0):
                                has_print = 1
                                break

                    if(not has_print):
                        if co_correlation[i,j] > 0.8079:
                            if co_contrast[i,j] > 35:
                                if co_homogeneity[i,j] < 0.45:
                                    cv2.circle(new_img,( j*offset+8 , i*offset+8 ),21 ,0,-1)
```

而我們發現有部分地方雖然周圍有被cover到了，但是卻還是有一小部分沒有被cover到，所以設定說若是周圍被cover的pixel小於一定數量的話視作沒有被cover到。

我們針對被我們過濾的瑕疵部分進行檢測，若是他的周圍已經被cover過了就不再對該位置進行cover的動作。

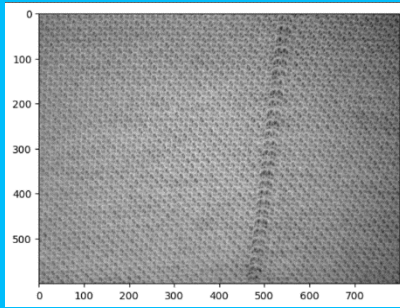
```
else:
    for k in range(0, 7):
        if(has_print):
            break

    for l in range(0, 7):
        # skip the pixel nearby center of circle
        if(not k==3 and not l==3):
            if(i*offset+5+k < x and j*offset+5+l < y):
                if(g[i*offset+5+k, j*offset+5+l] == 0):
                    has_print = 1
                    break

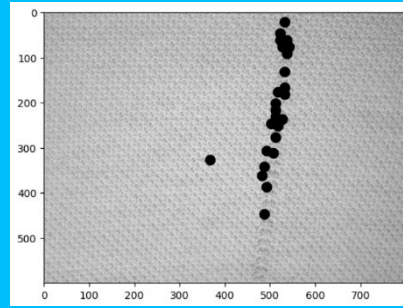
    if(not has_print):
        cv2.circle(g,( j*offset+8 , i*offset+8 ), 12 , 0, -1)
        count[23]+=1
```



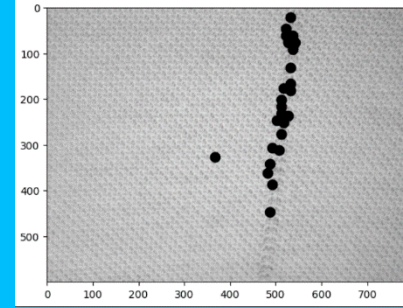
# 細節修復



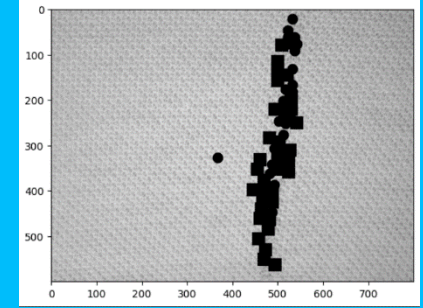
Filter = 15  
Offset = 5



Filter = 15+23  
Offset = 5



Filter = 15+23+33  
Offset = 5



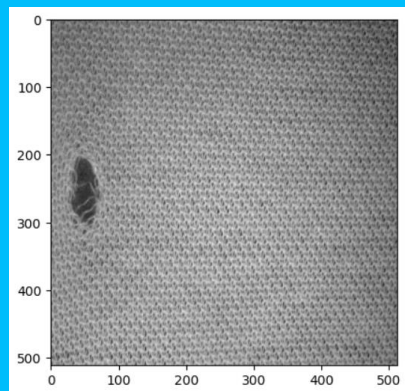
Filter = 再+40  
Offset = 5

前述偵測線條瑕疵的無法將從影片中擷取下來的影像內之線條瑕疵做辨識，  
故再做一個特別偵測該影像用之部分，  
因為有些難以判讀此部分線條的瑕疵，經過數10小時的測試後最後得出，  
將filter放大並將offset縮小來做判讀，  
想法為若有瑕疵的話那其有瑕疵的部分應會很明顯地和周圍無瑕疵的部分有所不同。

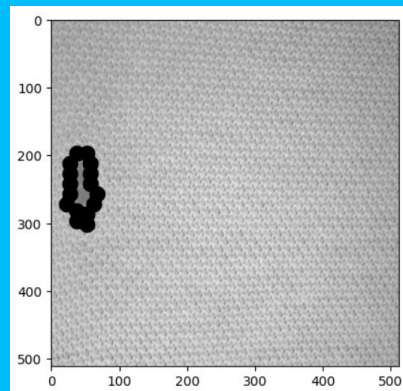
The background is a deep blue gradient. A prominent, glowing circular ring of light particles is centered in the frame. Surrounding this ring are several faint, ethereal, and somewhat abstract shapes that resemble smoke or particle trails, giving a sense of movement and depth. The overall aesthetic is futuristic and scientific.

成果

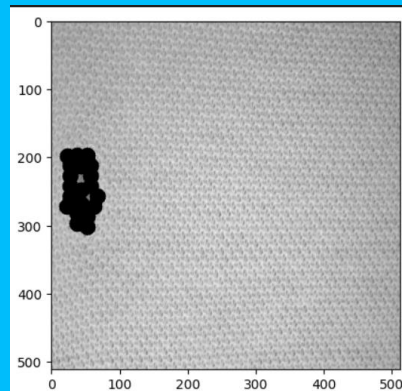
# 偵測Image2



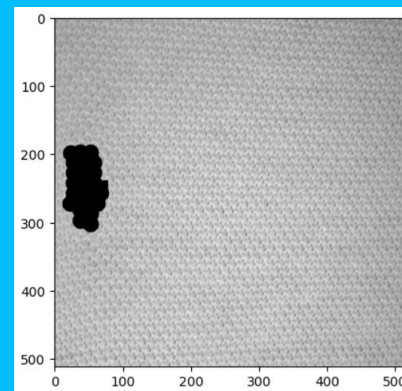
Filter = 15  
Offset = 5



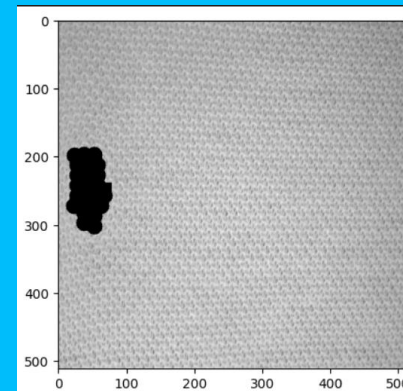
Filter = 15+23  
Offset = 5



Filter = 15+23+33  
Offset = 5



Filter = 再+17  
Offset = 5

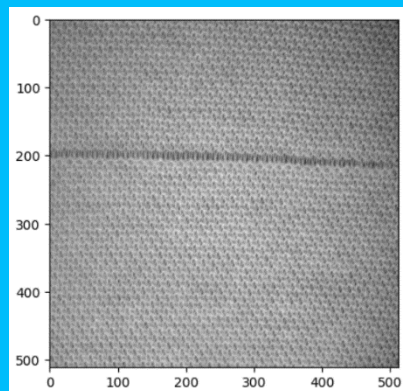


Filter = 再+40  
Offset = 5

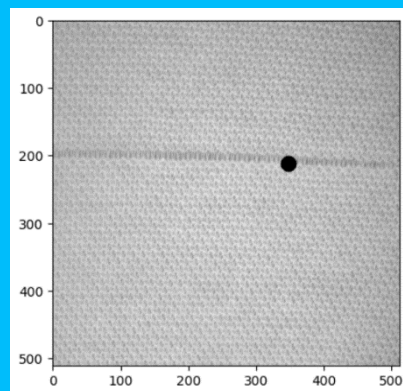
最後的結果如最右圖所呈現的，雖然有點超出瑕疵本身範圍，  
但就新手而言且是用statistics計算而非autoencoder，  
整體的表現，我覺得應該算是蠻不錯的，有達到老師的目標。



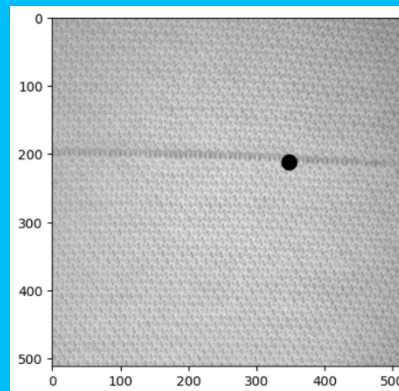
# 偵測Image1



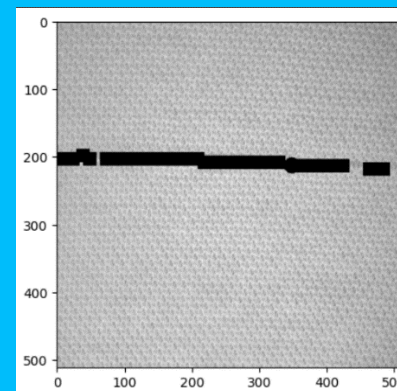
Filter = 15  
Offset = 5



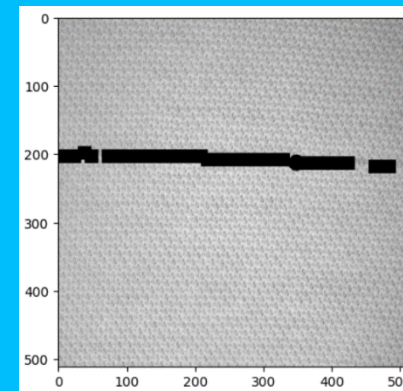
Filter = 15+23  
Offset = 5



Filter = 15+23+33  
Offset = 5



Filter = 再+17  
Offset = 5



Filter = 再+40  
Offset = 5

最後的結果如最右圖所呈現的，除了最右邊沒有偵測到以外，大致OK，  
而最右邊會偵測不到就如同先前的猜測，它應該非瑕疵點，  
所以實際上那邊本來就應該長那樣，  
顧在這張圖的表現來看，也是相當的不錯，有逼近老師的要求。

# THANK YOU

409410197 劉柏辰 409416541 周騏軍