

# 多媒體概論-Final Project

目錄：(可直接點擊，藍色為實驗，綠色為分析，橘色為總結)

<u>One-feature-RGB</u> .....	2
<u>One-feature-HIST</u> .....	4
<u>One-feature-LBP</u> .....	5
<u>One-feature-Correlogram</u> .....	8
<u>One-feature-Analysis</u> .....	11
<u>More-features-RGB<sub>(0.9)</sub>、HIST<sub>(0.2)</sub>、Correlogram<sub>(0.5)</sub></u> .....	14
<u>More-features-RGB<sub>(0.9)</sub>、LBP<sub>(0.5)</sub>、Correlogram<sub>(0.5)</sub></u> .....	15
<u>More-feature-Analysis</u> .....	17
<u>All-features-RGB<sub>(0.9)</sub>、HIST<sub>(0.15)</sub>、LBP<sub>(0.5)</sub>、Correlogram<sub>(0.2)</sub></u> .....	19
<u>All-features-RGB<sub>(0.9)</sub>、HIST<sub>(0.15)</sub>、LBP<sub>(0.2)</sub>、Correlogram<sub>(0.25)</sub></u> .....	19
<u>All-feature-Analysis</u> .....	21
<u>Collection for 1082-VGG16</u> .....	22
<u>Collection for 1082-ResNet18</u> .....	22
<u>VGG16 &amp; ResNet18-Analysis</u> .....	24
<u>Summary</u> .....	25

## One-feature-RGB :

The screenshot shows the One-feature-RGB interface with the following parameters:

- Display:** Target = 45, << page 0 of 43, >>
- Choose img:** Collection of 1082
- Cell size:** LBP 32
- Enter Weight:** Feature 1 (RGB)
- Feature:** HIST
- Query:** Vgg16, ResNet18
- Other settings:** Color Bar, Global Statistics

Below the interface, a grid of 10 images is displayed, labeled 45, 70, 59, 194, 73, 112, 239, 283, 56, 107.

▲ target=45 , Img\_origin p0

The screenshot shows the One-feature-RGB interface with the following parameters:

- Display:** Target = 45, << page 0 of 43, >>
- Choose img:** Collection of 1082
- Cell size:** LBP 32
- Enter Weight:** Feature 1 (Color Bar)
- Feature:** HIST
- Query:** Vgg16, ResNet18
- Other settings:** Global Statistics

Below the interface, a grid of 10 images is displayed, labeled 45, 70, 59, 194, 73, 112, 239, 283, 56, 107.

▲ target=45 , Colorbar p0

The screenshot shows the One-feature-RGB interface with the following parameters:

- Display:** Target = 45, << page 0 of 43, >>
- Choose img:** Collection of 1082
- Cell size:** LBP 32
- Enter Weight:** Feature 1 (RGB)
- Feature:** HIST
- Query:** Vgg16, ResNet18
- Other settings:** Global Statistics

Below the interface, a grid of 10 images is displayed, labeled 47, 256, 201, 276, 124, 100, 258, 195, 53, 54.

▲ target=45 , Img\_origin p1

The screenshot shows the One-feature-RGB interface with the following parameters:

- Display:** Target = 45, << page 0 of 43, >>
- Choose img:** Collection of 1082
- Cell size:** LBP 32
- Enter Weight:** Feature 1 (Color Bar)
- Feature:** HIST
- Query:** Vgg16, ResNet18
- Other settings:** Global Statistics

Below the interface, a grid of 10 images is displayed, labeled 47, 256, 201, 276, 124, 100, 258, 195, 53, 54.

▲ target=45 , Colorbar p1

The screenshot shows the One-feature-RGB interface with the following parameters:

- Display:** Target = 185, << page 0 of 43, >>
- Choose img:** Collection of 1082
- Cell size:** LBP 32
- Enter Weight:** Feature 1 (RGB)
- Feature:** HIST
- Query:** Vgg16, ResNet18
- Other settings:** Global Statistics

Below the interface, a grid of 10 images is displayed, labeled 185, 183, 173, 174, 180, 178, 175, 182, 189, 181.

▲ target=185 , Img\_origin p0

The screenshot shows the One-feature-RGB interface with the following parameters:

- Display:** Target = 185, << page 0 of 43, >>
- Choose img:** Collection of 1082
- Cell size:** LBP 32
- Enter Weight:** Feature 1 (Color Bar)
- Feature:** HIST
- Query:** Vgg16, ResNet18
- Other settings:** Global Statistics

Below the interface, a grid of 10 images is displayed, labeled 185, 183, 173, 174, 180, 178, 175, 182, 189, 181.

▲ target=185 , Colorbar p0

The screenshot shows the One-feature-RGB interface with the following parameters:

- Display:** Target = 185, << page 0 of 43, >>
- Choose img:** Collection of 1082
- Cell size:** LBP 32
- Enter Weight:** Feature 1 (RGB)
- Feature:** HIST
- Query:** Vgg16, ResNet18
- Other settings:** Global Statistics

Below the interface, a grid of 10 images is displayed, labeled 185, 186, 179, 176, 177, 188, 184, 193, 187, 192.

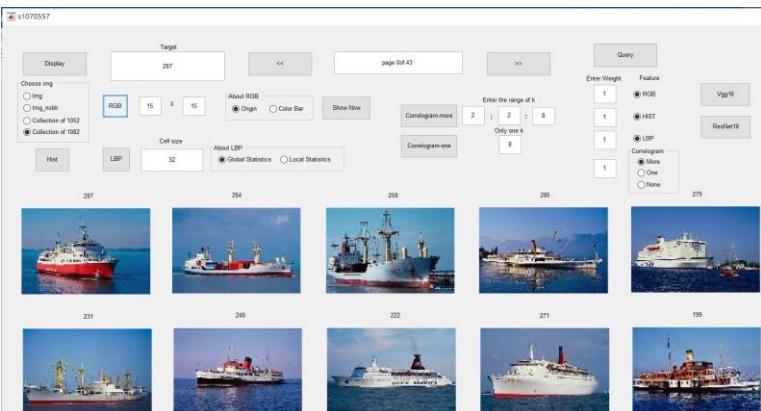
▲ target=185 , Img\_origin p1

The screenshot shows the One-feature-RGB interface with the following parameters:

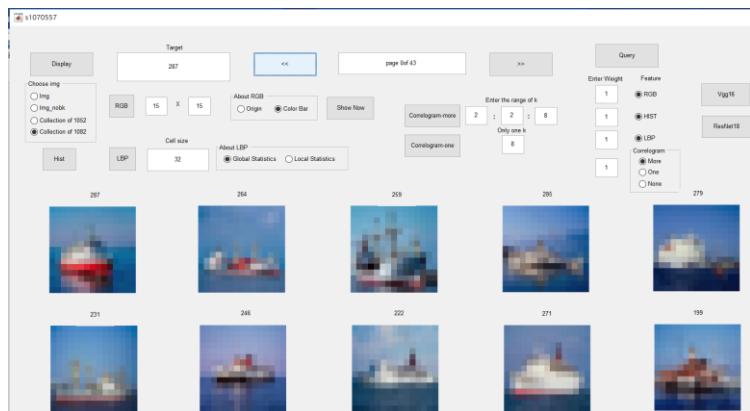
- Display:** Target = 185, << page 0 of 43, >>
- Choose img:** Collection of 1082
- Cell size:** LBP 32
- Enter Weight:** Feature 1 (Color Bar)
- Feature:** HIST
- Query:** Vgg16, ResNet18
- Other settings:** Global Statistics

Below the interface, a grid of 10 images is displayed, labeled 185, 186, 179, 176, 177, 188, 184, 193, 187, 192.

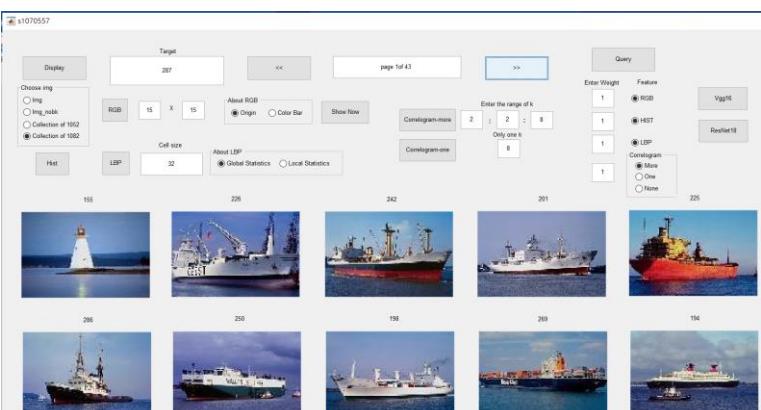
▲ target=185 , Colorbar p1



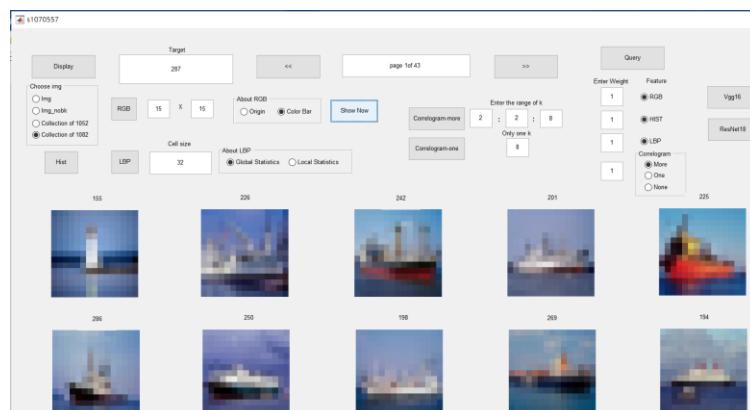
▲ target=287 , Img\_Origin p0



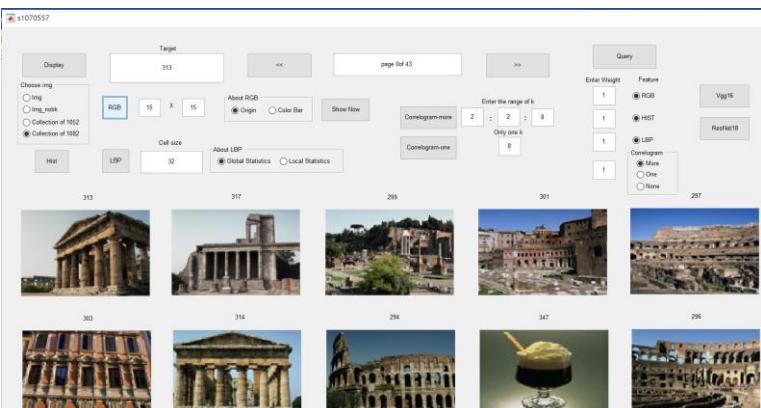
▲ target=287 , Colorbar p0



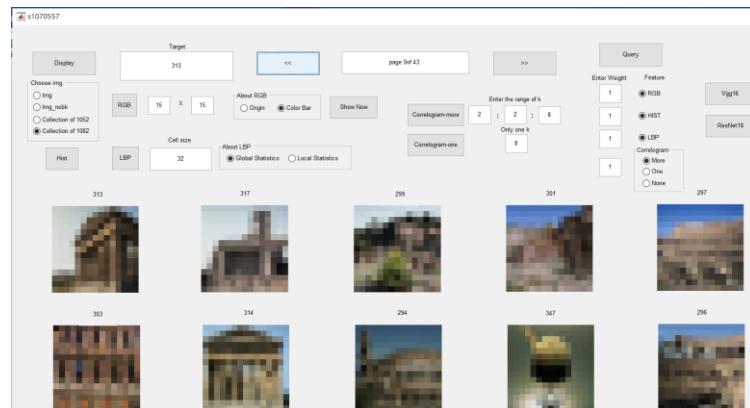
▲ target=287 , Img\_Origin p1



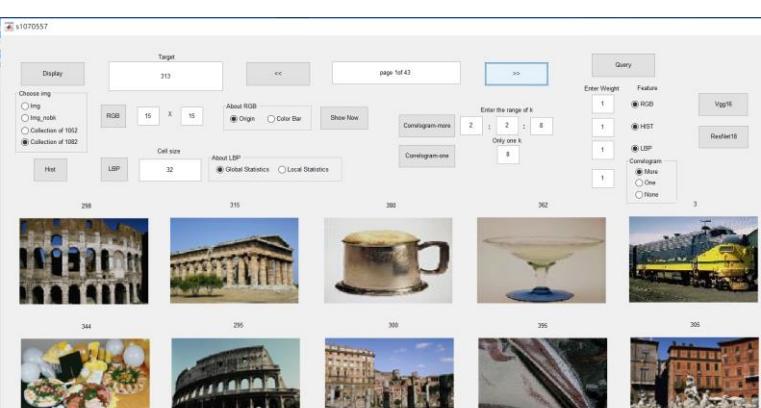
▲ target=287 , Colorbar p1



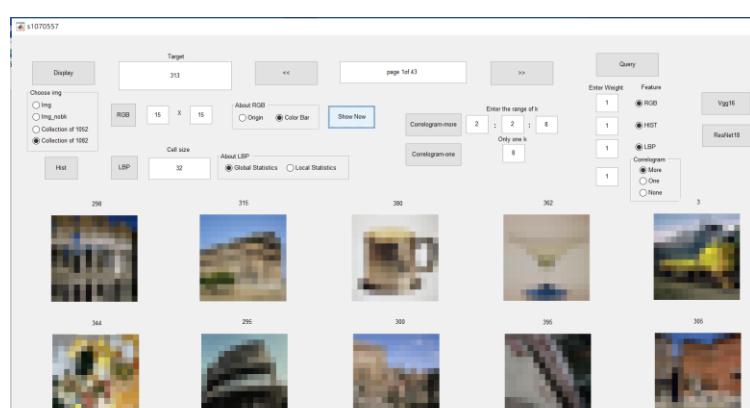
▲ target=313 , Img\_Origin p0



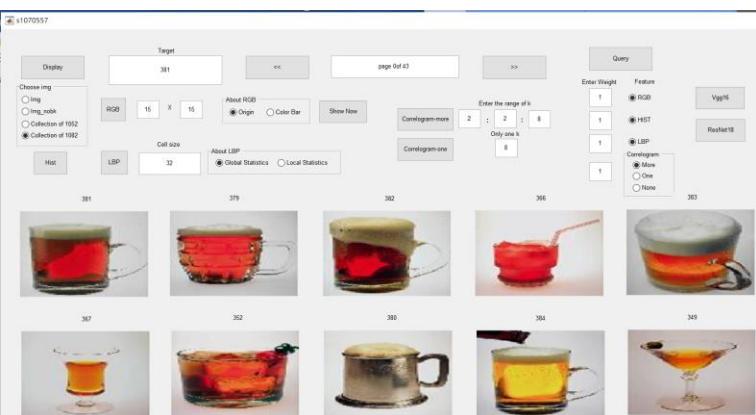
▲ target=313 , Colorbar p0



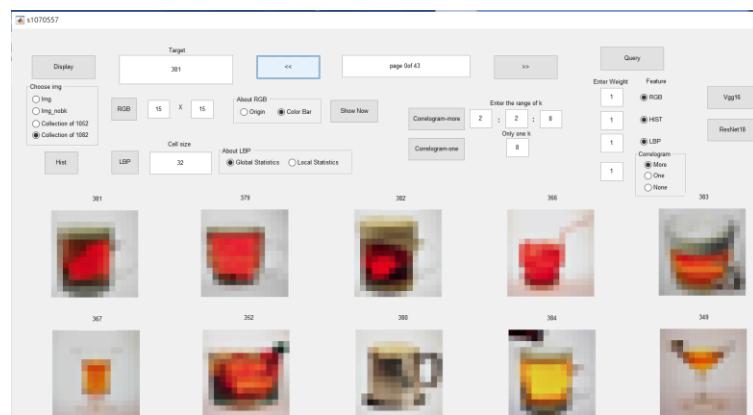
▲ target=313 , Img\_Origin p1



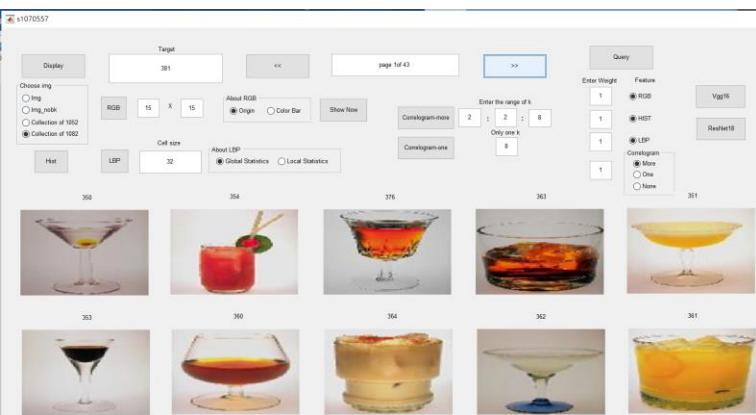
▲ target=313 , Colorbar p1



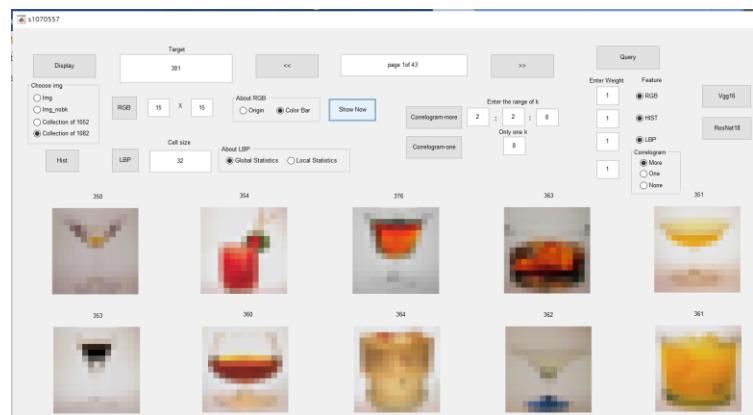
▲ target=381 , img\_orgin p0



▲ target=381 , Colorbar p0

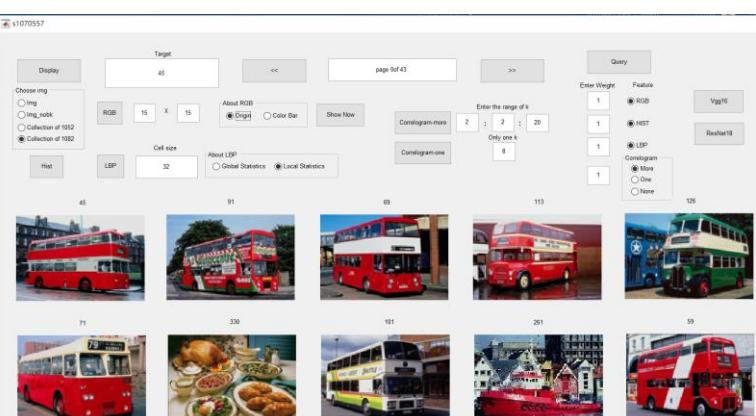


▲ target=381 , img\_orgin p1

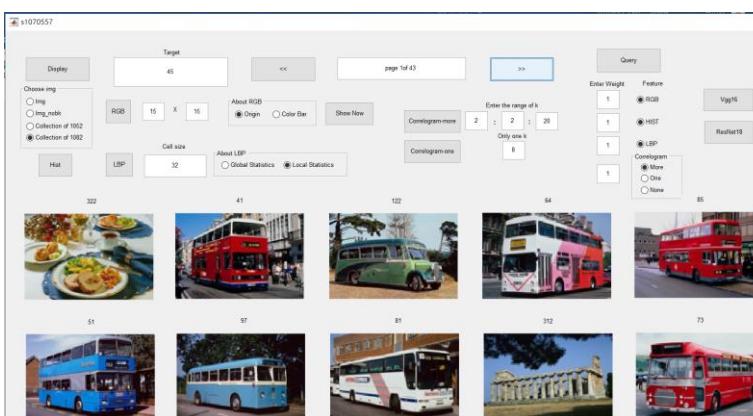


▲ target=381 , Colorbar p1

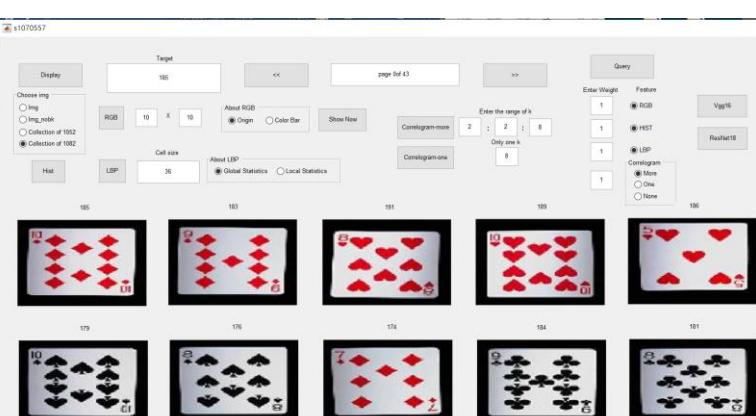
## One-feature-HIST :



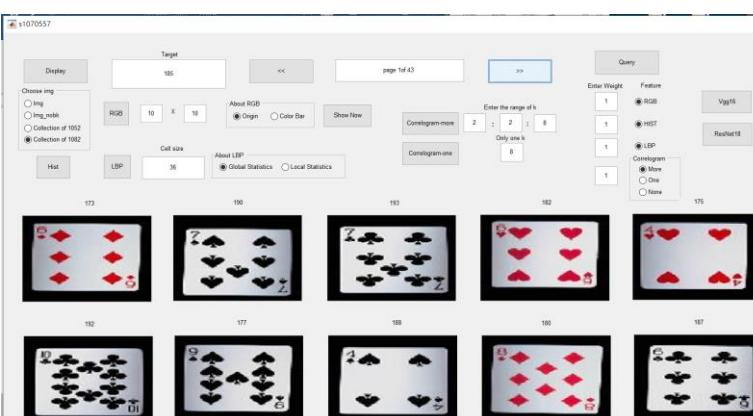
▲ target=45 , img\_orgin p0



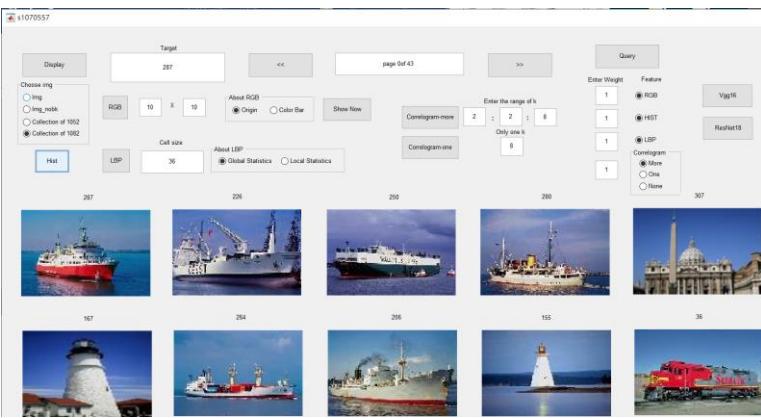
▲ target=45 , img\_orgin p1



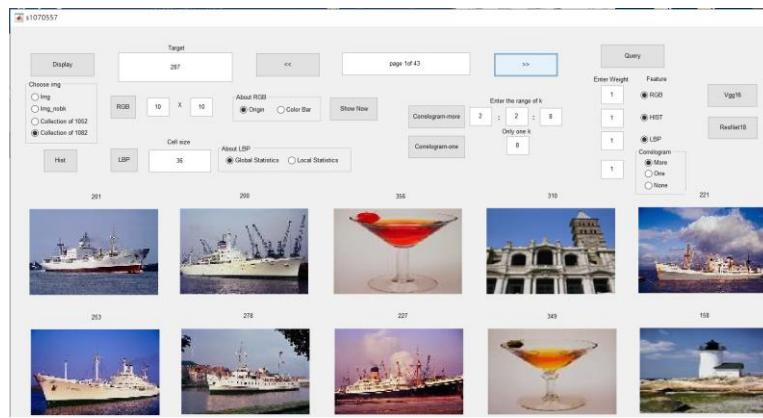
▲ target=185 , img\_orgin p0



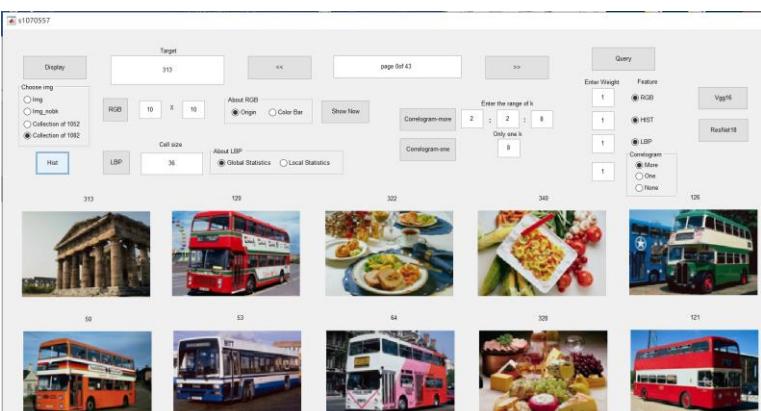
▲ target=185 , img\_orgin p1



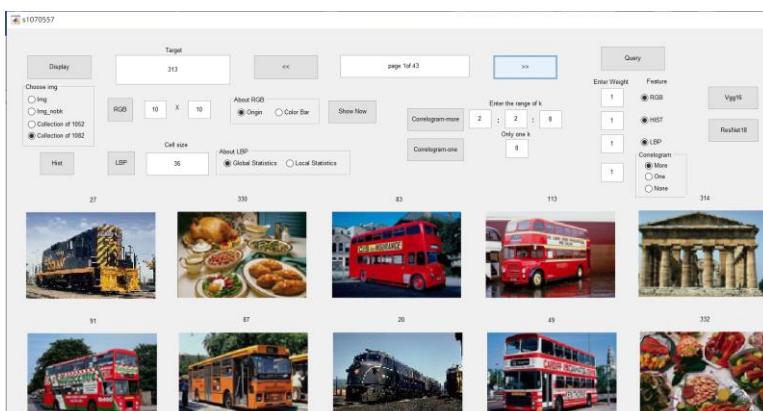
▲ target=287 , img\_origin p0



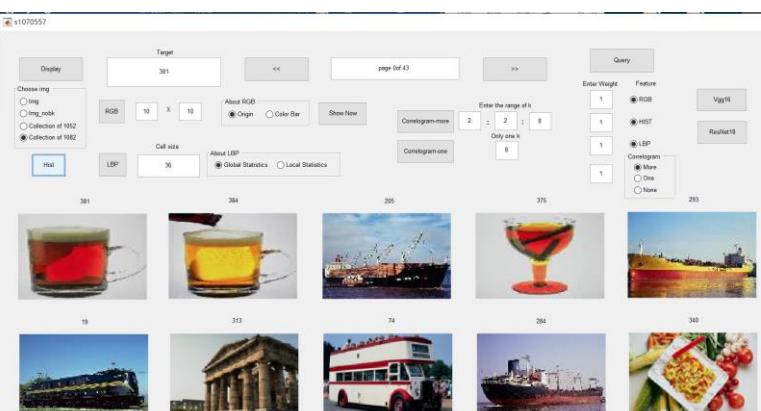
▲ target=287 , img\_origin p1



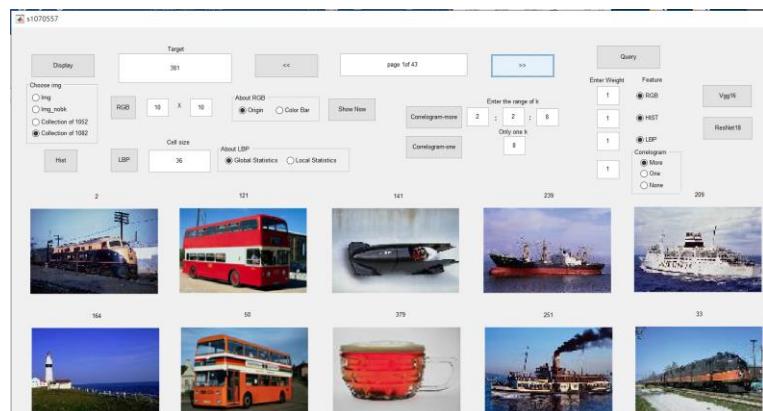
▲ target=313 , img\_origin p0



▲ target=313 , img\_origin p1

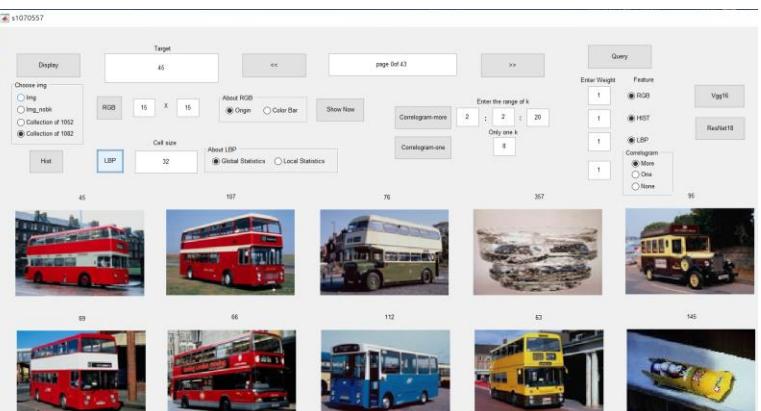


▲ target=381 , img\_origin p0

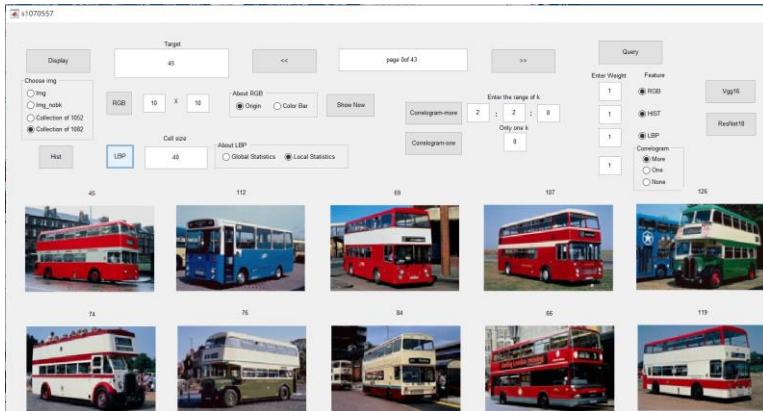


▲ target=381 , img\_origin p1

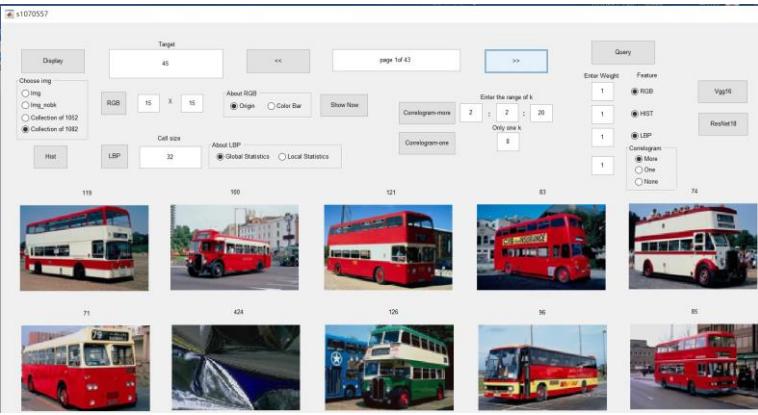
## One-feature-LBP :



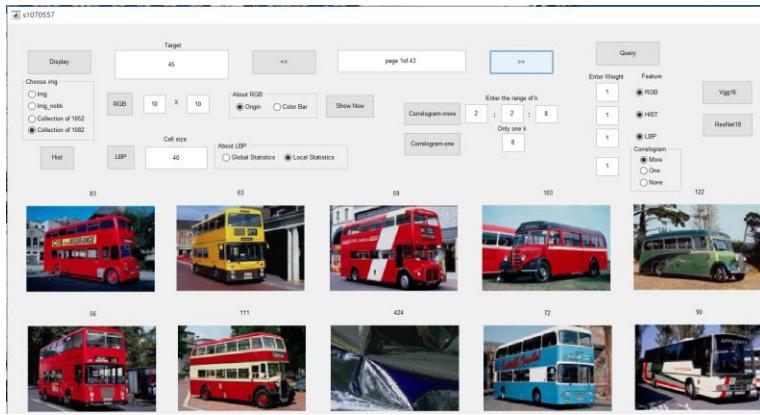
▲ target=45 , Global, p0



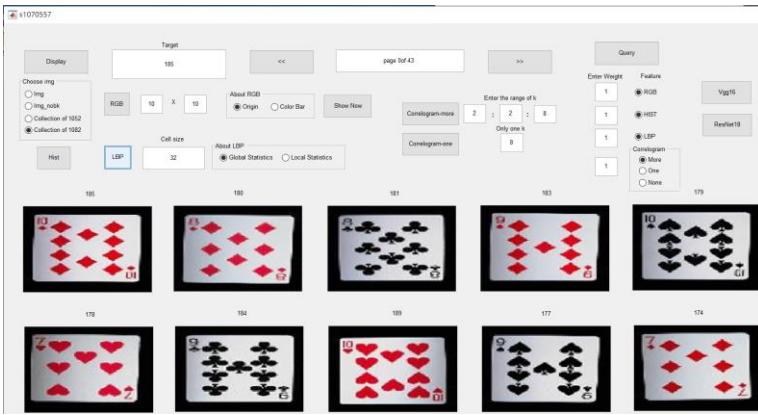
▲ target=45 , Local, p0



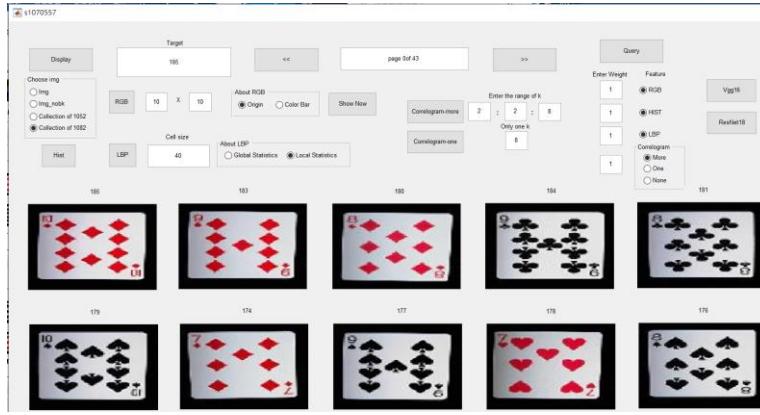
▲ target=45 , Global, p1



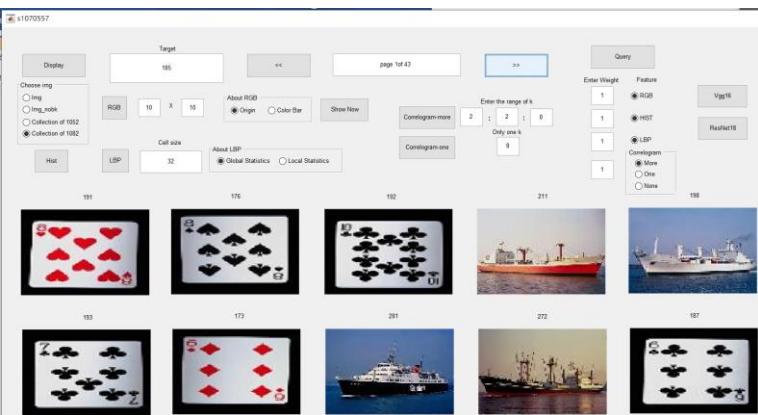
▲ target=45 , Local, p1



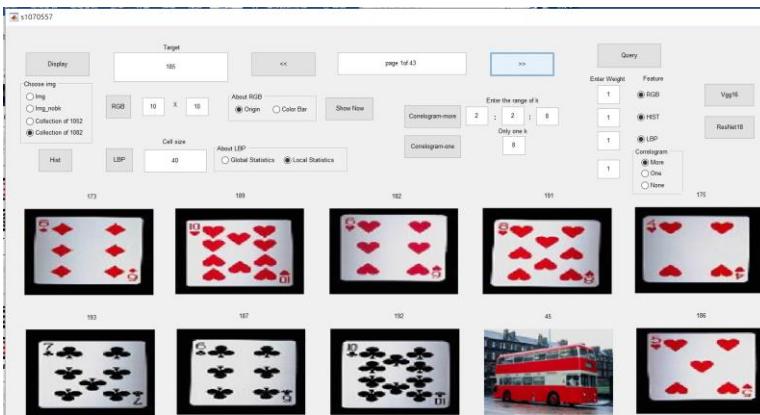
▲ target=185 , Global, p0



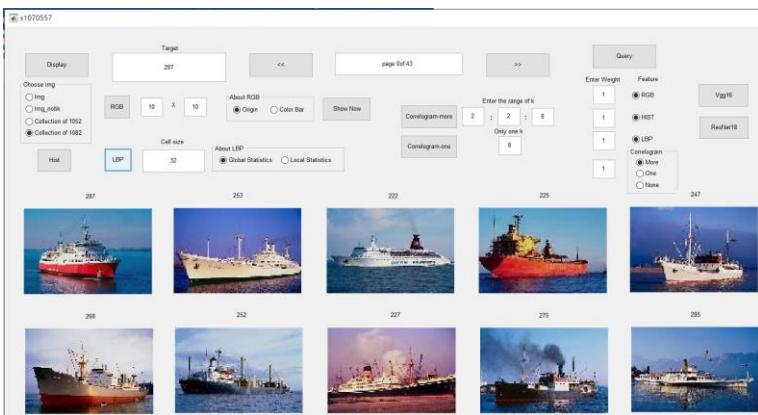
▲ target=185 , Local, p0



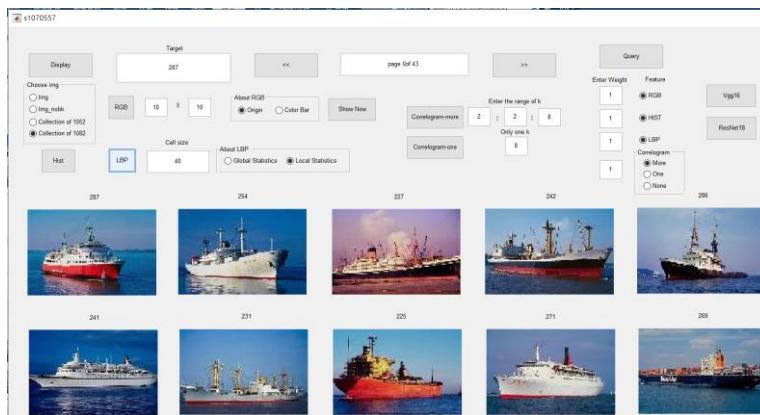
▲ target=185 , Global, p1



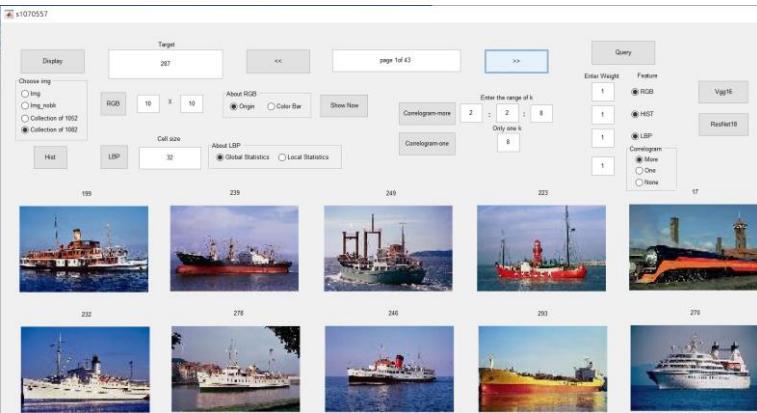
▲ target=185 , Local, p1



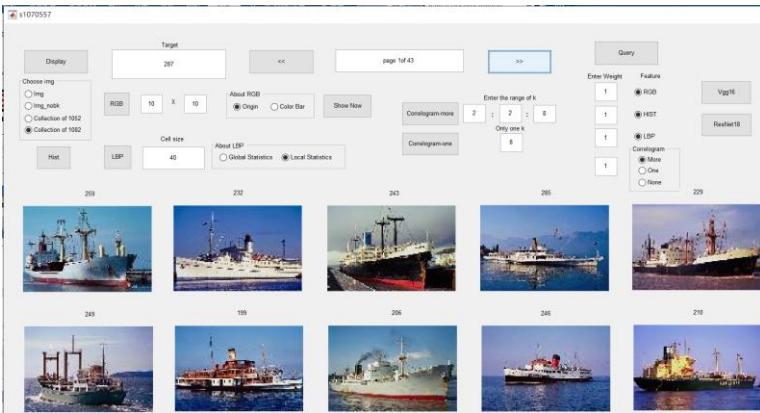
▲ target=287 , Global, p0



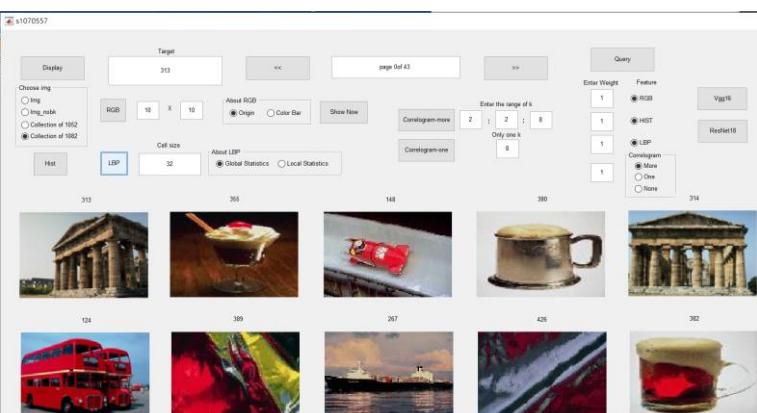
▲ target=287 , Local, p0



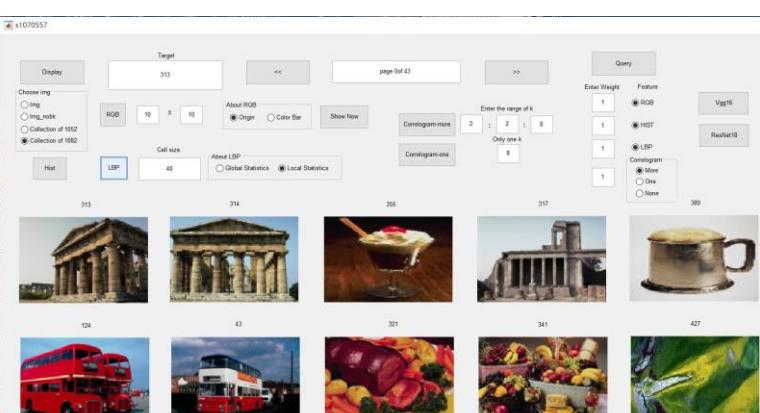
▲ target=287 , Global, p1



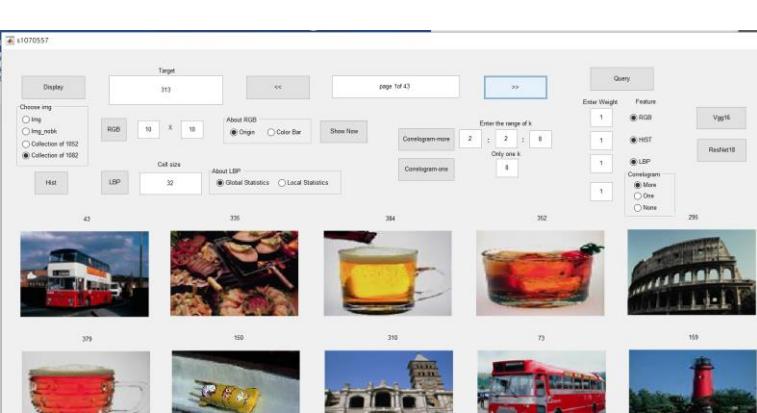
▲ target=287 , Local, p1



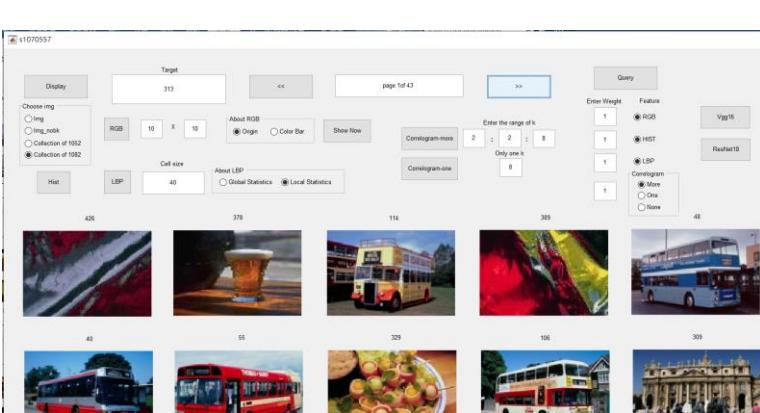
▲ target=313 , Global, p0



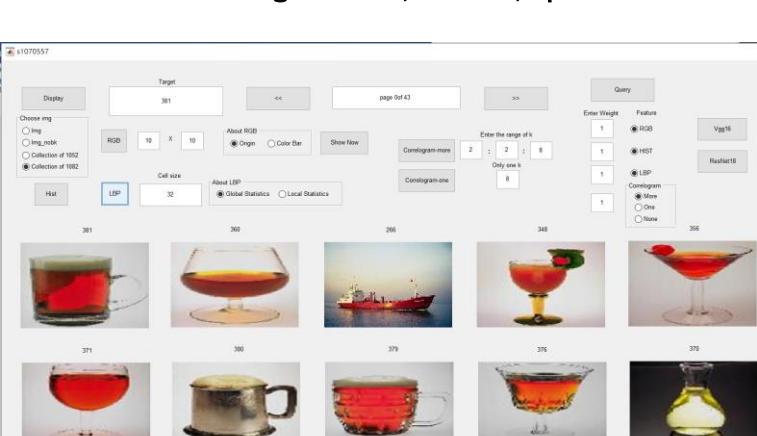
▲ target=313 , Local, p0



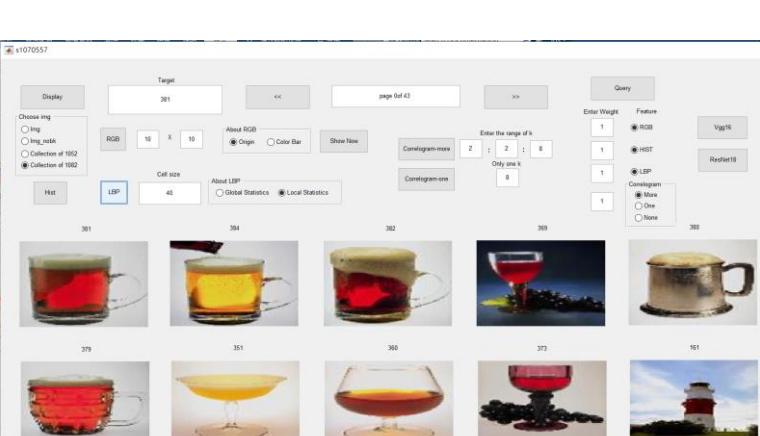
▲ target=313 , Global, p1



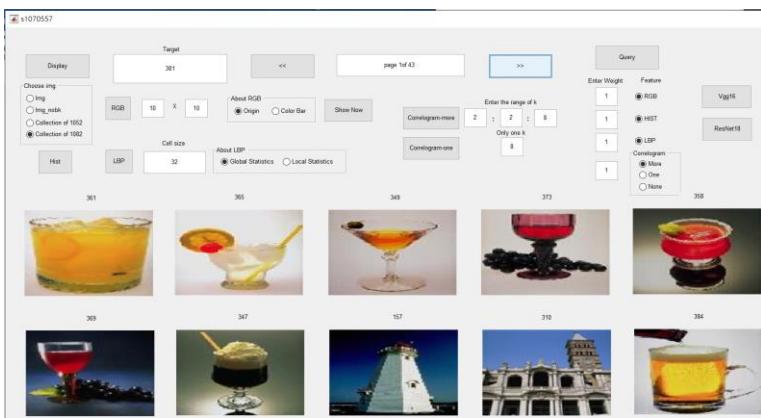
▲ target=313 , Local, p1



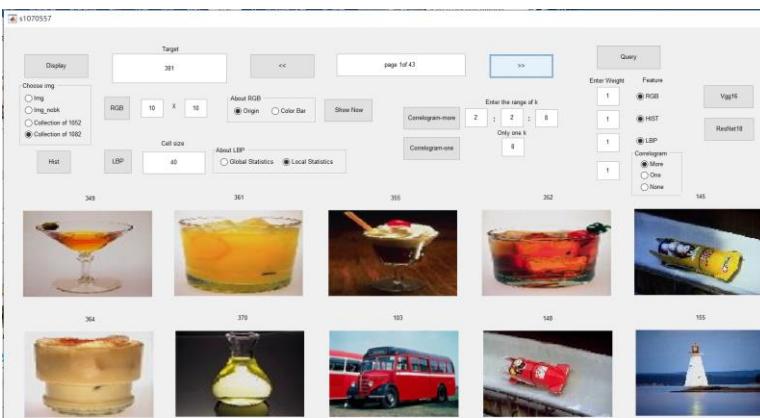
▲ target=381 , Global, p0



▲ target=381 , Local, p0

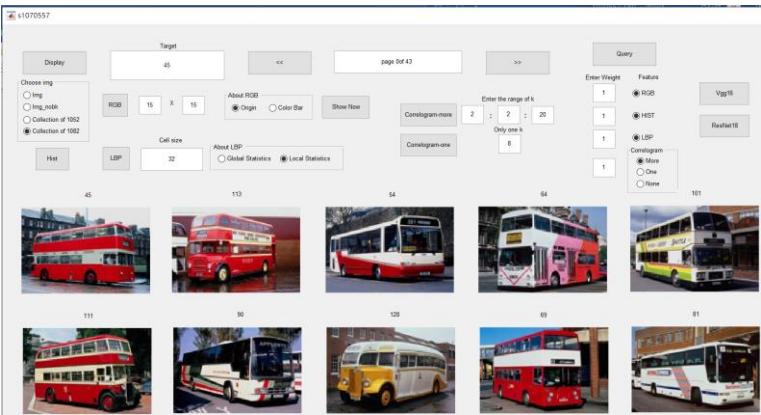


▲ target=313 , Global, p1

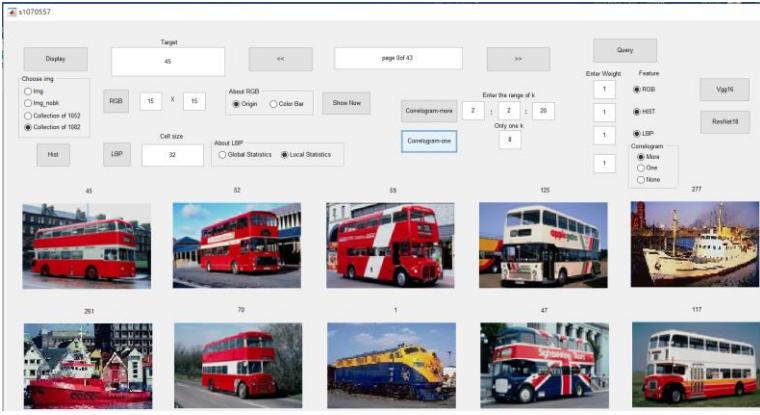


▲ target=313 , Local, p1

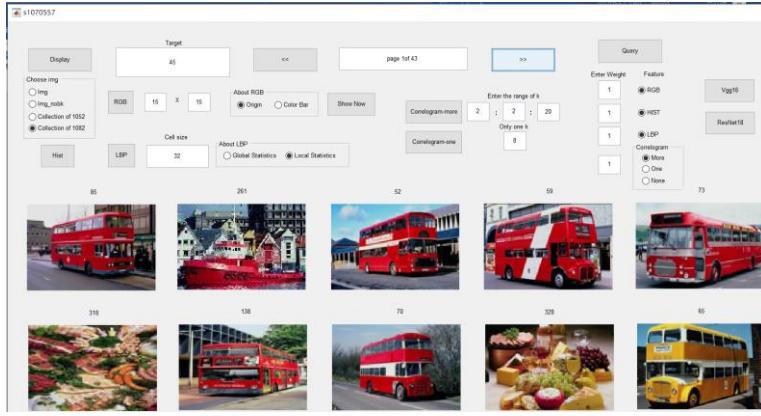
## One-feature-Correlogram



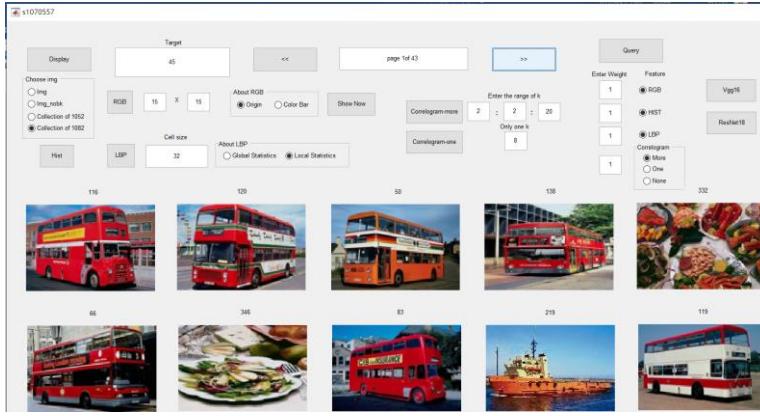
▲ target=45 , More-K, p0



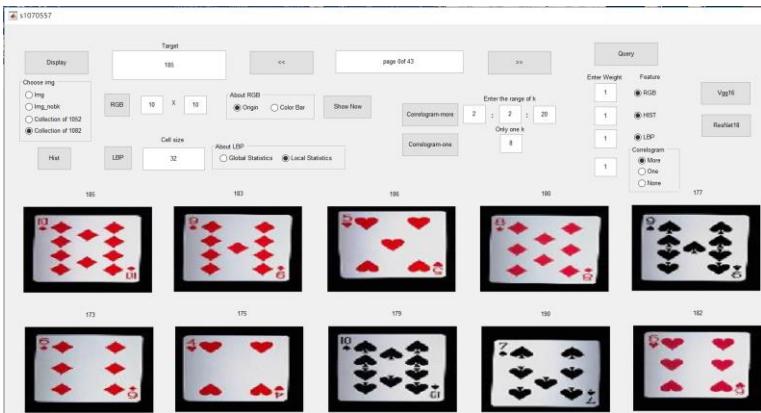
▲ target=45 , One-K, p0



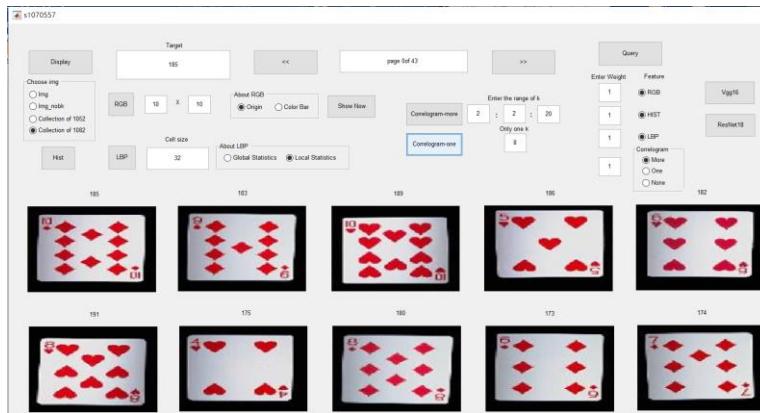
▲ target=45 , More-K, p1



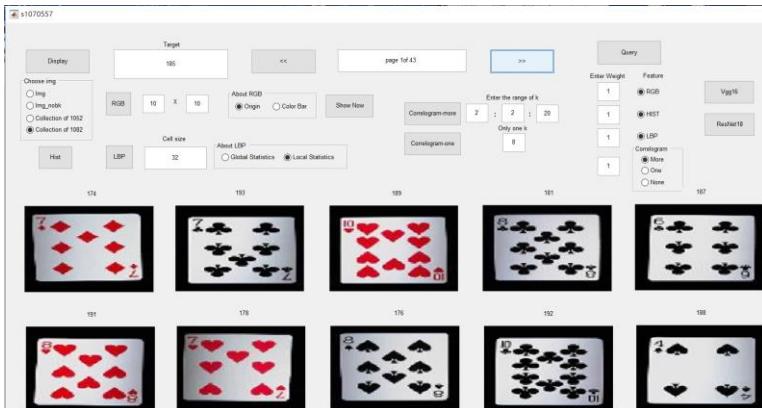
▲ target=45 , One-K, p1



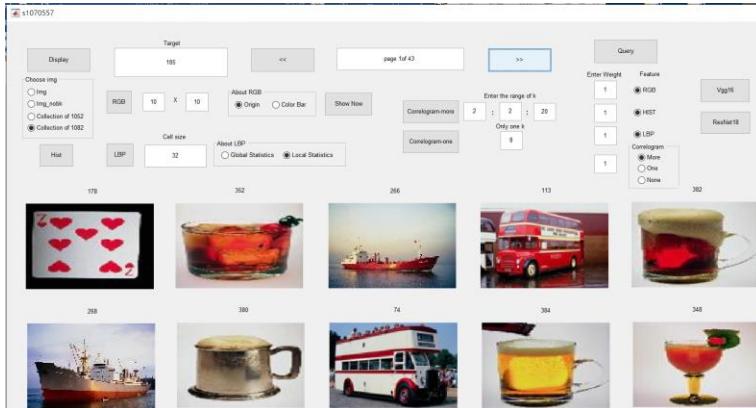
▲ target=185 , More-K, p0



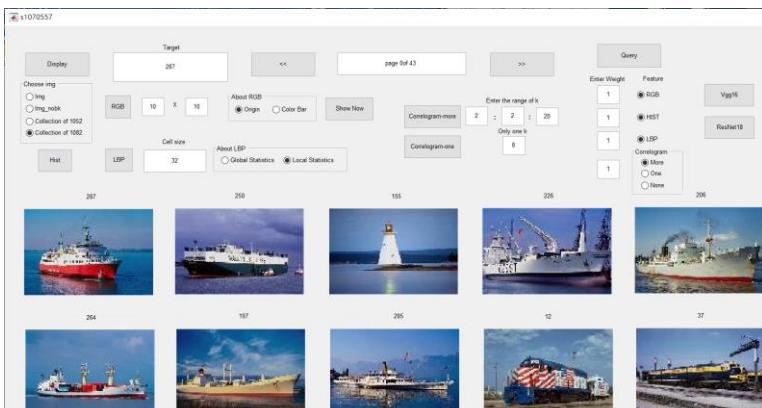
▲ target=185 , One-K, p0



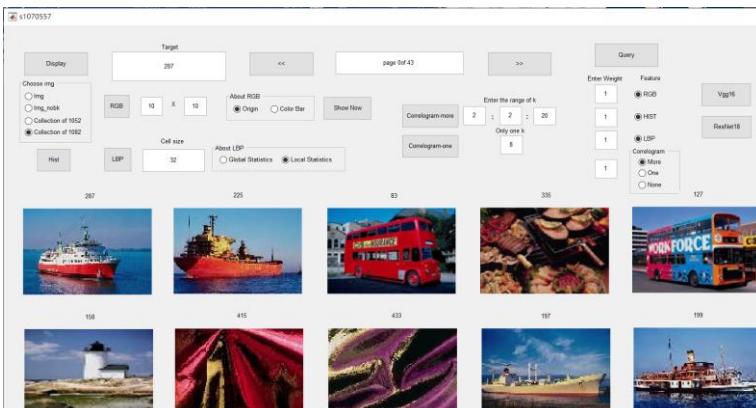
▲ target=185 , More-K, p1



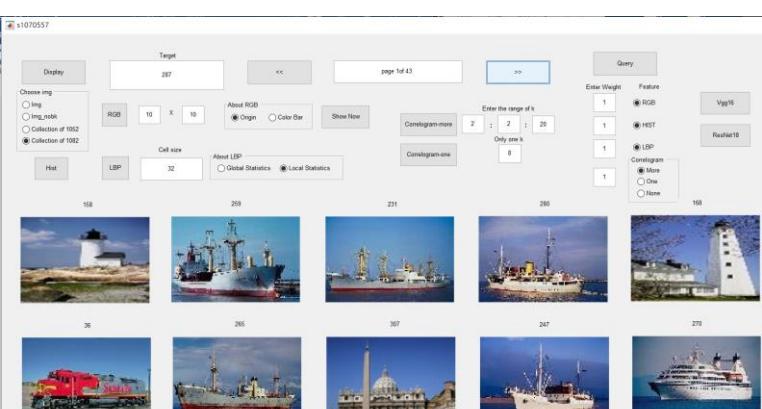
▲ target=185 , One-K, p1



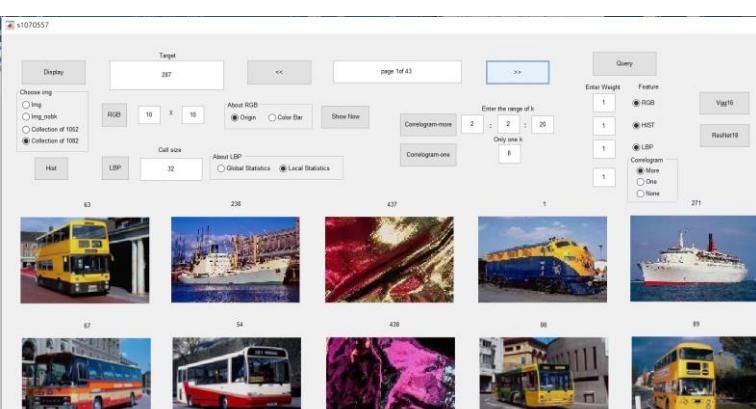
▲ target=287 , More-K, p0



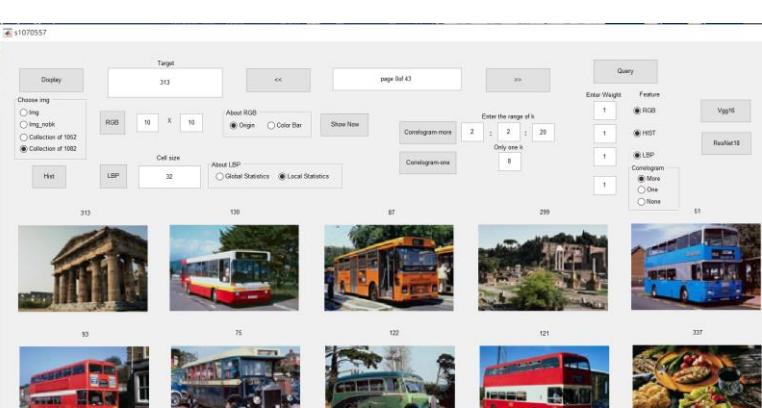
▲ target=287 , One-K, p0



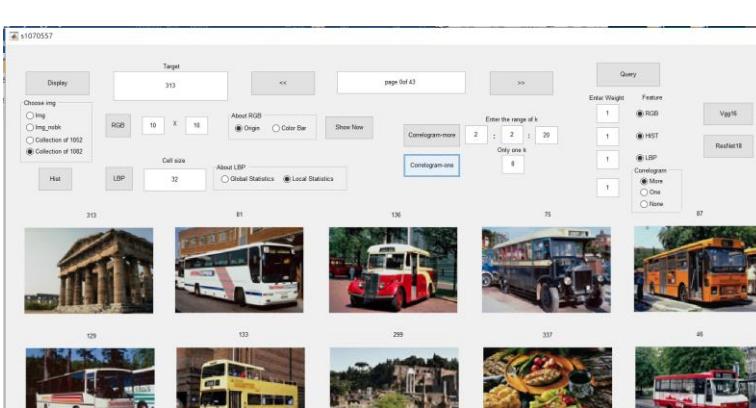
▲ target=287 , More-K, p1



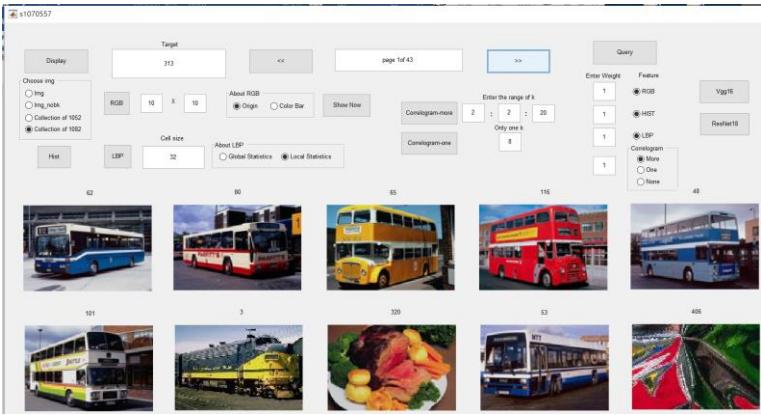
▲ target=287 , One-K, p1



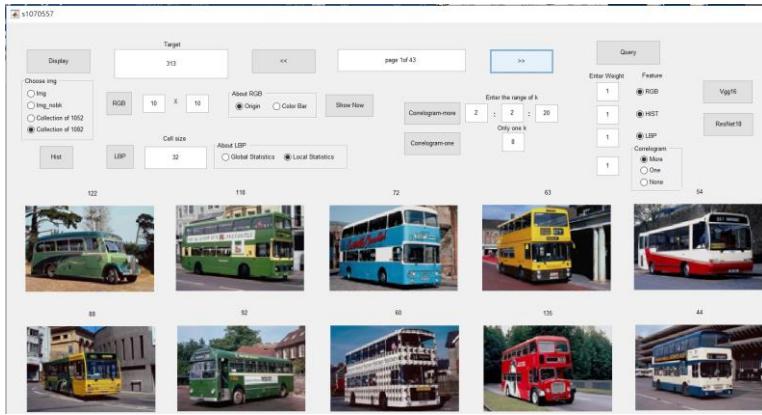
▲ target=313 , More-K, p0



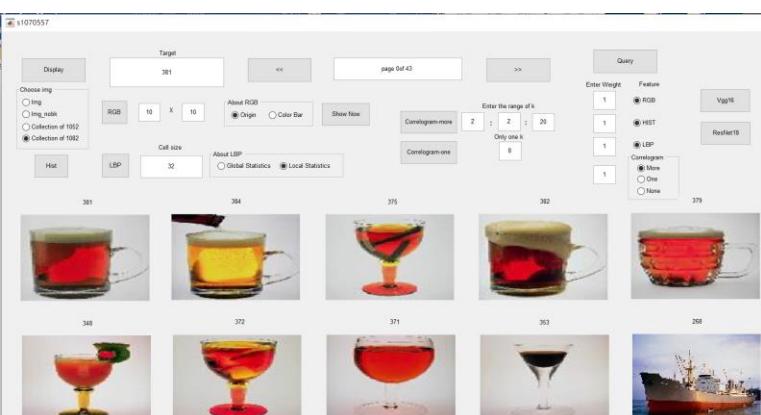
▲ target=313 , One-K, p0



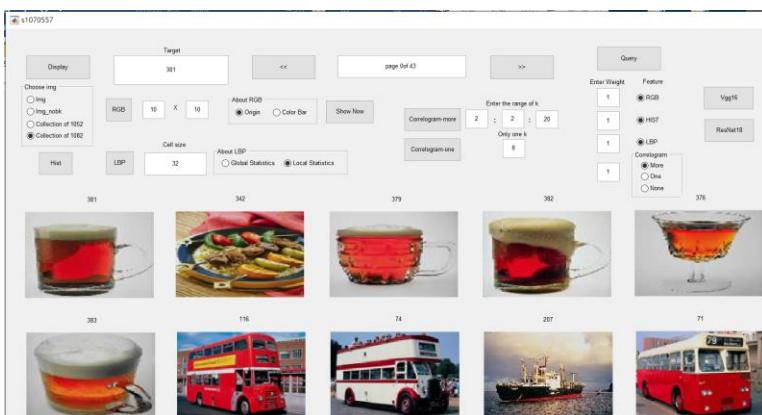
▲ target=313 , More-K, p1



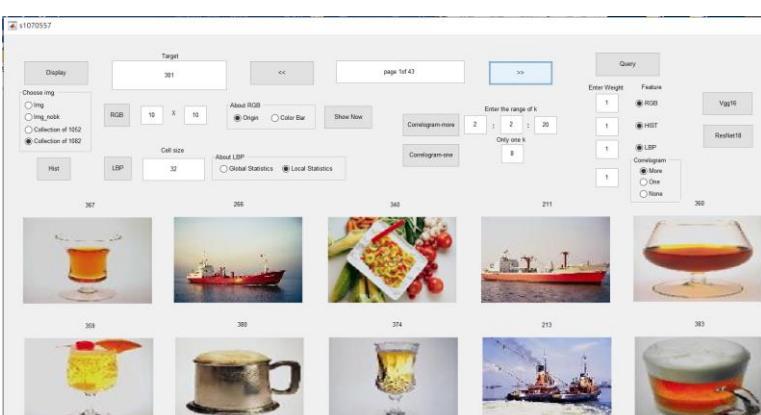
▲ target=313 , One-K, p1



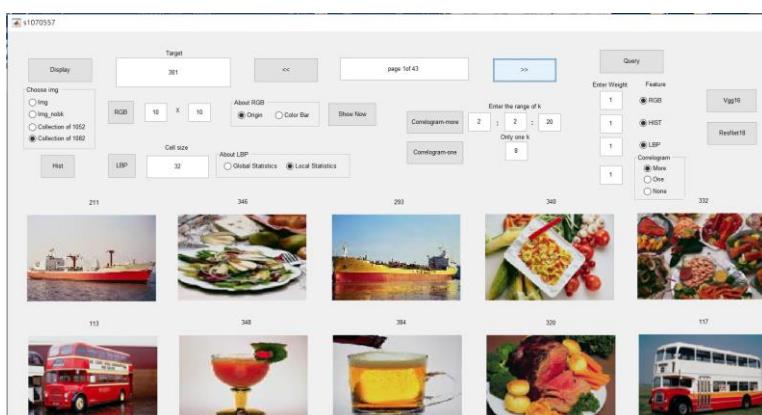
▲ target=381 , More-K, p0



▲ target=381 , One-K, p0



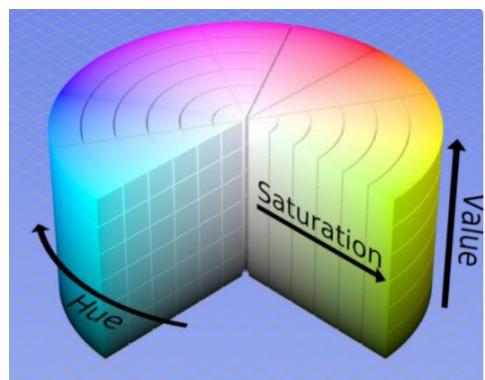
▲ target=381 , More-K, p0



▲ target=381 , One-K, p0

## One-feature-Analysis :

從 RGB 和 HIST 來看，我們可以發現到其實兩者是很相像的邏輯，但是也因為這一點點的邏輯差異，造成在不同種類的圖片上做篩選時，篩選能力直接表露無遺。在 RGB 中，是利用將圖片縮到特定幾成幾的大小以後去計算每個方格的顏色分布，而將 rgb2HSV 以後顏色可以更加精準，HSV 考量了色相、飽和度、明度，較類似於人類感覺的顏色，原本 RGB 則是代表紅綠藍三個通道的顏色，明度則是加總，故 255、255、255 是白色。HIST 則是直接去拿整張圖片去劃分成 16 個顏色含量去做比較，方法較前者粗略。

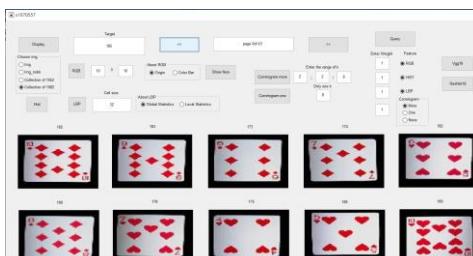


▲ HSV 色彩空間

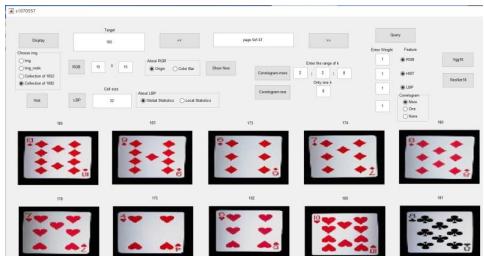
而從圖片比較結果來看 RGB 和 HIST，在 45-公車上，雖然看起來似乎是 HIST 表現較優秀，但我認為是格子大小劃分的影響，因為在 RGB 中， $15 \times 15$  在五個類別的平均表現上較好，但是公車似乎用  $10 \times 10$  可以有更好的效果。但是卻在 185-撲克牌上  $10 \times 10$  不夠精細，改成  $15 \times 15$  以後它能準確地將 21 張撲克牌放在前 21 名，HIST 也能達到這個結果。



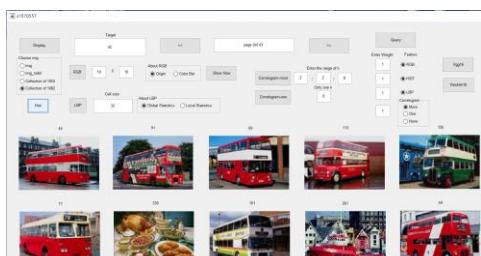
▲ 公車 45, RGB, 10x10, p0



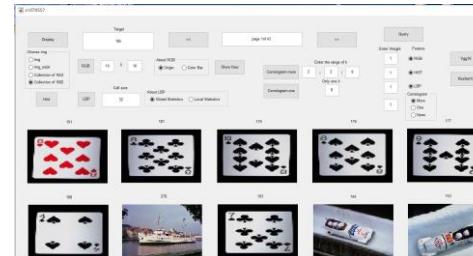
▲ 撲克牌 185, RGB, 10x10, p0



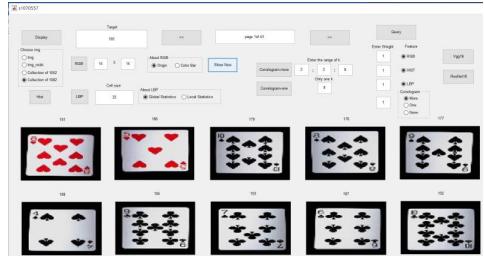
▲ 撲克牌 185, RGB, 15x15, p0



▲ 公車 45, HIST, p0

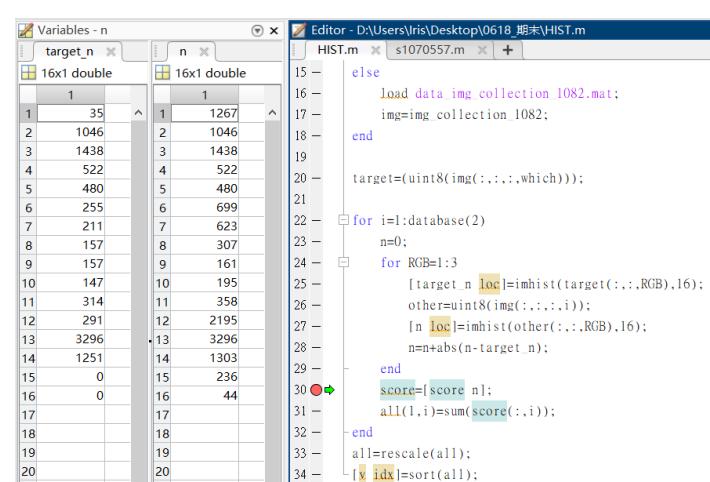


▲ 撲克牌 185, RGB, 10x10, p1



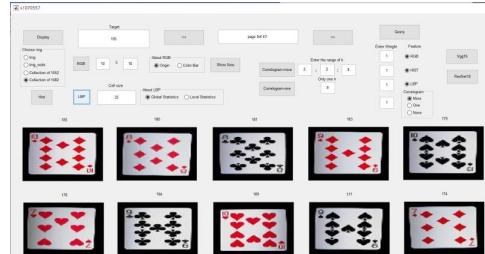
▲ 撲克牌 185, RGB, 15x15, p1

在 287-船、313-建築、381 飲料上，HIST 的篩選能力明顯弱了許多，在排序第 11~20 名中更加明顯，在 p1 會出現許多不相干的圖片，像是船會出現燈塔，建築會出現佳餚，飲料會出現船.....等。我認為是因為在 HIST 我們一次用整張圖片去分析 16 種顏色的分布情形，在顏色越複雜的圖片可能分析越不準確，而飲料可能因為數值過於偏頗導致不少同樣背景偏向單色、中間有一物的圖片也排序在前面。

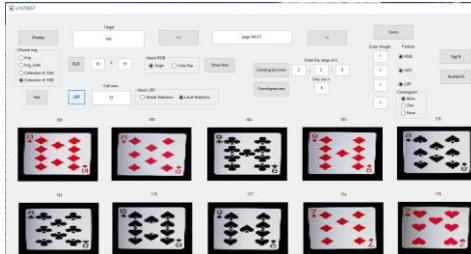


▲ 飲料 381 與圖片 205 比較時的數值  
(Target\_n 為 381, n 為相減後)

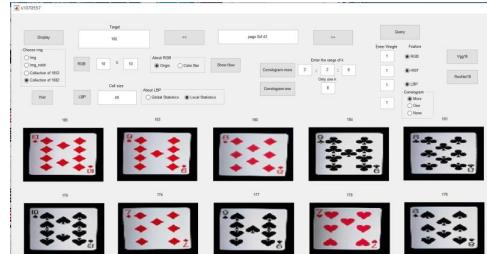
在 LBP 中，它有分 Global 和 Local 取特徵方式，普遍而言會覺得第 0 頁的差異度不大，LBP 有點類似篩以圖片的輪廓去做篩選比對，較不會考量到顏色，在 Local 中有一個 CellSize 的數值可以去調整特徵的提取，窗口 CellSize 的大小會影響到輪廓的清晰與否，在沒有調整到合適的 CellSize 時，Global 的篩選能力會較 Local 突出，而我原先是用 CellSize=32 去篩選，後來逐漸調整到 40 是我認為較好的特徵提取。



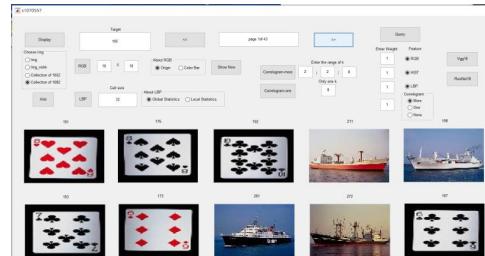
▲ 撲克牌 185, Global, p0



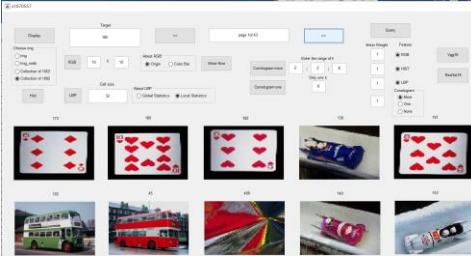
▲ 撲克牌 185, Local, p0  
(CellSize=32)



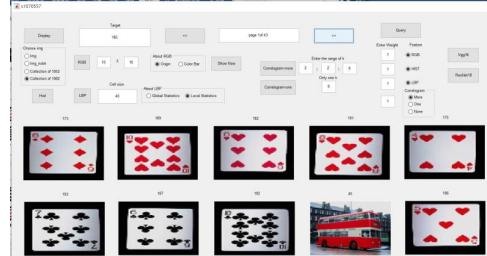
▲ 撲克牌 185, Local, p0  
(CellSize=40)



▲ 撲克牌 185, Global, p1

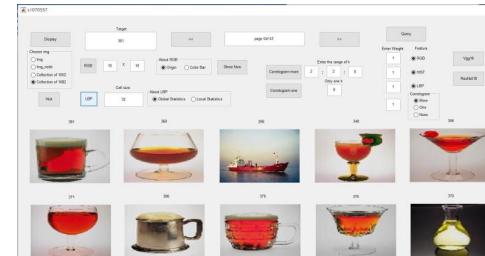


▲ 撲克牌 185, Local, p1  
(CellSize=32)

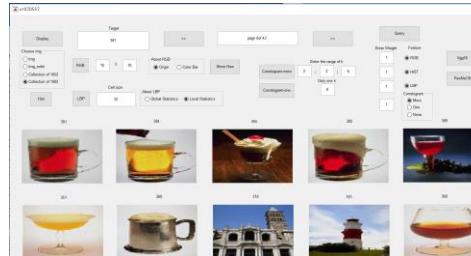


▲ 撲克牌 185, Local, p1  
(CellSize=40)

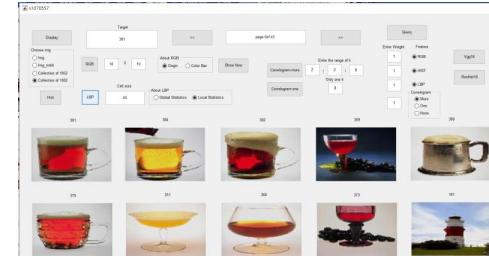
在 45-公車、185-撲克牌、287-船中，LBP 都有不錯的篩選結果，可能是因為輪廓較容易抓取，且無太過雜亂的背景。在 313-建築上，兩者其實表現都沒有很好，而 Local 相較 Global 可能略勝一籌，我認為是因為圖庫中的建築大部分差異較大，加上可能有更適合篩選建築的 CellSize 大小，畢竟不一定每種圖片都對同一個特徵提取方式有良好的結果。在 381-飲料中我們可以觀察到，Local 很明顯地將杯子相同而飲料也是裝滿的圖片排序為前幾名。



▲ 飲料 381, Global, p0



▲ 飲料 381, Local, p0  
(CellSize=32)



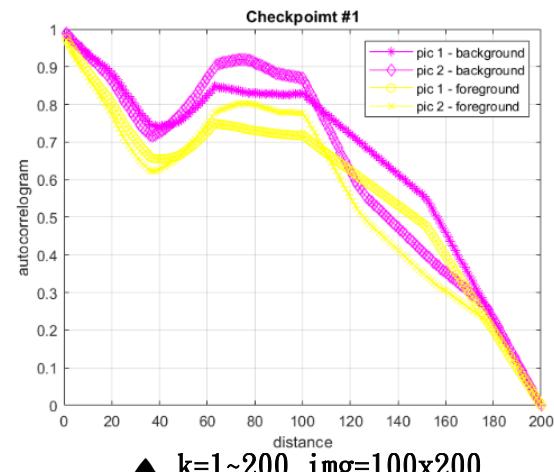
▲ 飲料 381, Local, p0  
(CellSize=40)

在 LBP 中，整體而言我認為 Local 相較 Global 來的好，而 CellSize=40 的結果比 CellSize=32 的結果優秀，但是感覺較不適用於輪廓雜亂的圖片(像是建築)。

在 Correlogram 中可以發現到 · Correlogram 是利用  $k$  矩形上出現和中心相同值之機率去做計算 · 若是越相似的圖片在機率-距離曲線上的走勢會越接近(以之前的報告為例) · 之後再帶入公式做運算得到差異值去做排序。

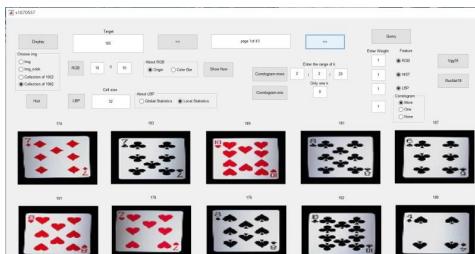


▲ 原始圖片

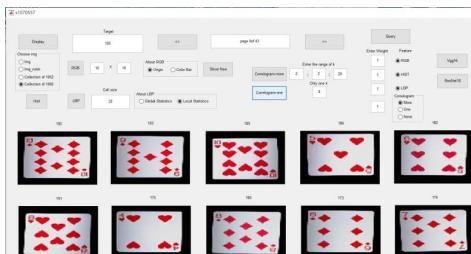
▲  $k=1 \sim 200$ , img=100x200

而採取單一個  $K$  和採取多個  $K$  為特徵時，在不同種類的圖片上的表現差異甚大，甚至會覺得似乎很難找到一個在篩選五種類型圖片中都能表現還不錯的  $k$  值，在採取多種  $k$  時，我仿造之前報告所定的數值去做微調 ·  $k$  值取的越多不一定就代表越精準 · 而  $k$  值的連續不連續亦會對結果有所影響 · 在想取得越遠又不計算得太久的前提下，我採用  $k=2:2:20$  做比對(在運算之前圖片皆已縮放到 16x24 的大小) · 在採取單一  $k$  值的部分我認為  $k=8$  是平均效果尚可的數值。

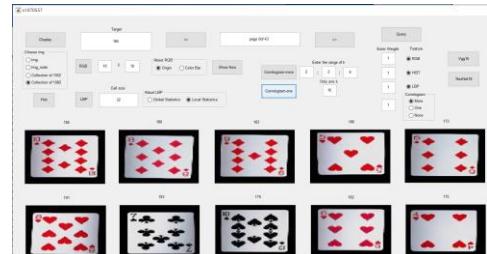
普遍而言我們可以發現到在篩五種不同種類的圖片時，無論是採取多  $k$  還是單一種  $k$ ，他們在前十名所呈現的結果都十分相像，在 185-撲克牌中若單看前十名會覺得是 One-k 勝出，因為在篩選方塊 10 上它可以完美的挑出所有圖庫裡紅色花色的牌，並將他們至於前 11 名 · 但是在 11 名以後會出現非撲克牌的圖片，對於這種特徵提取而言，因為會看重顏色和相對應之距離，因此黑桃、梅花的牌反而不會接續在 12 名之後。但若將  $k$  值調整成 10，雖然黑桃梅花能出現在前幾名，但是對於其他種類的圖片篩選反而都是扣分居多。



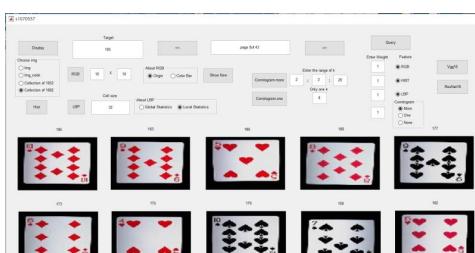
▲ 撲克牌 185 , More-K, p0



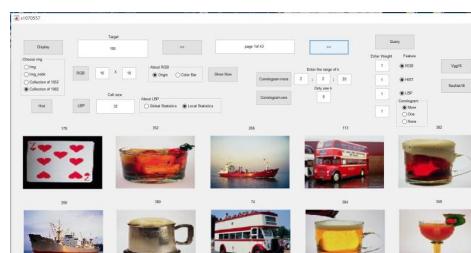
▲ 撲克牌 185 , One-K=8, p0



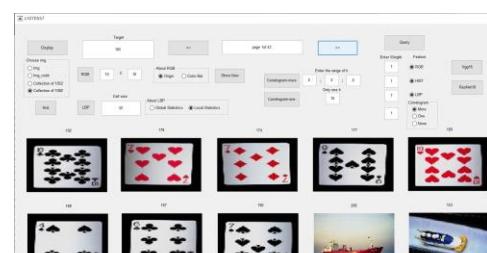
▲ 撲克牌 185 , One-K=10, p0



▲ 撲克牌 185 , More-K, p0



▲ 撲克牌 185 , One-K=8, p0

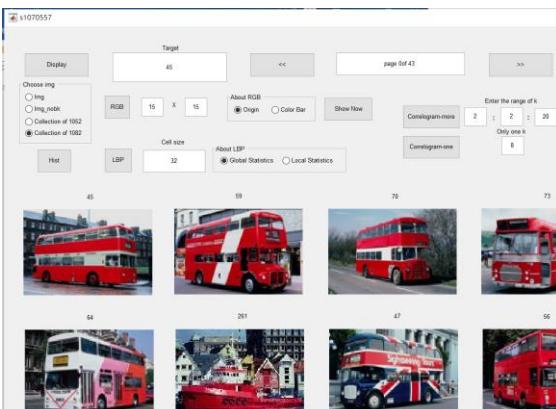


▲ 撲克牌 185 , One-K=10, p0

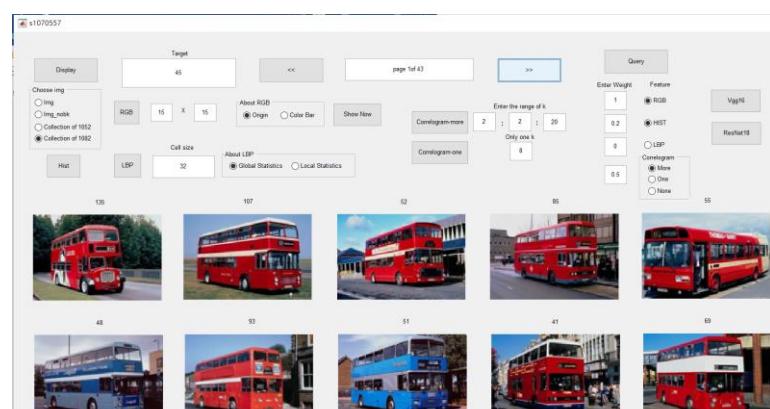
在 45-公車和 381-飲料的篩選上，More-k 所呈現的效果最好，我認為有一部分原因是因為若單一只參考一種 k，不同種類的圖片可能不一定都能在同一固定距離下呈現最好的結果，而 More-k 的前 20 名幾乎都不會有太多出乎意料的圖片種類出現。在 313-建築和 287-船中，相較前面三種圖片效果較沒那麼好，尤其是建築的部分，會篩出許多公車出來，而船我認為可能因為許多圖片背景都同為藍天，造成在 More-k 中，篩出部分不是船，但是構圖、顏色類似的圖片。

在 Correlogram 中，我認為在篩選圖片可能有許多種類時，較適合採用 More-k，除非是針對特定種類圖片就有可能用 One-k 較合適，但是在使用 More-k 時並非採取數值越多越好，需要再三調整數值才行。

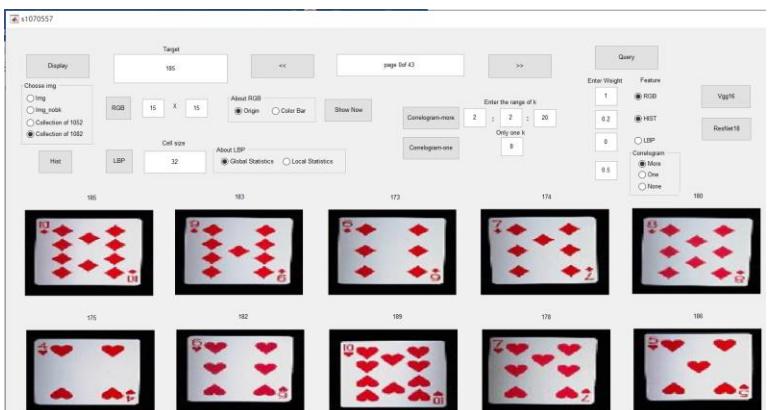
### More-features-RGB<sub>(0.9)</sub>、HIST<sub>(0.2)</sub>、Correlogram<sub>(0.5)</sub>：



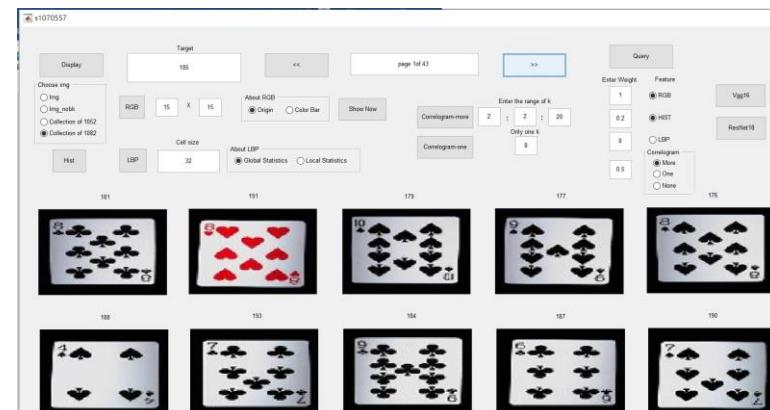
▲ target=45, p0



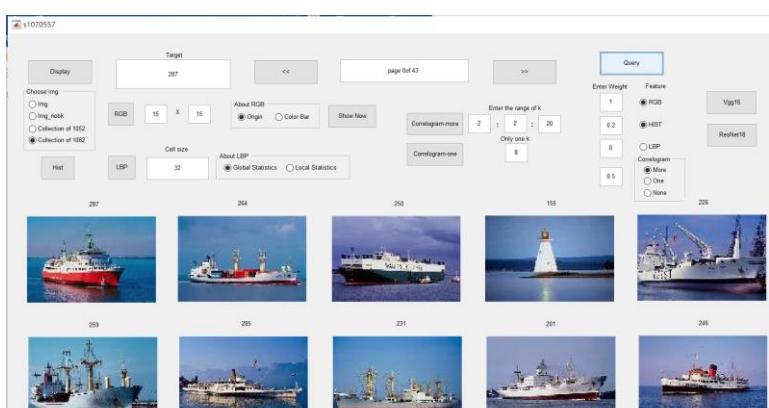
▲ target=45, p1



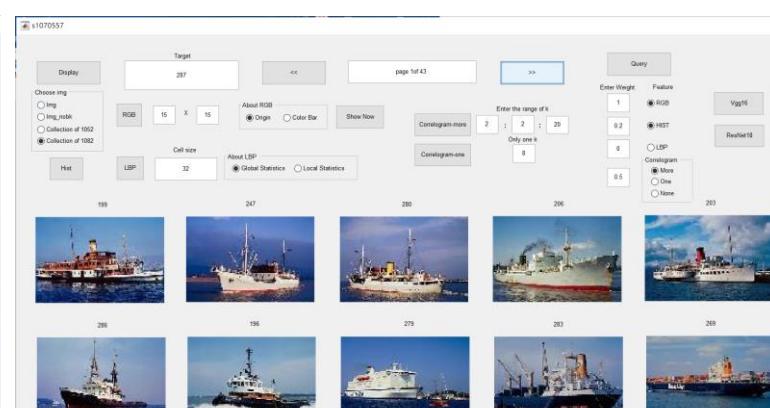
▲ target=185, p0



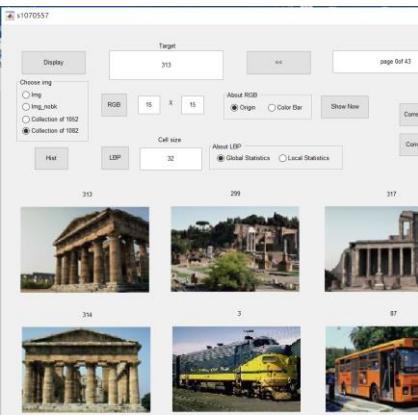
▲ target=185, p1



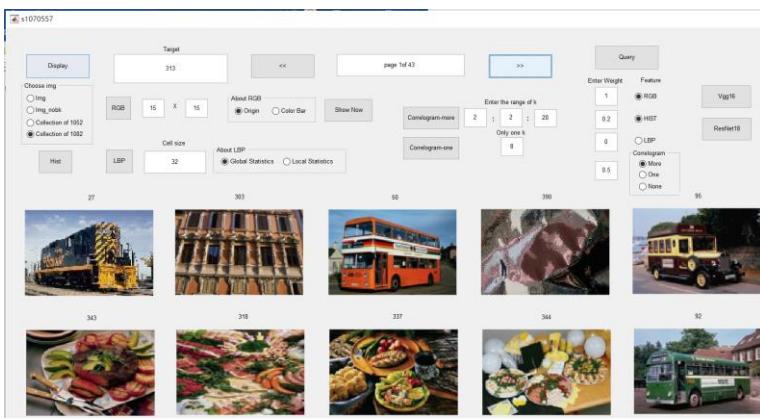
▲ target=287, p0



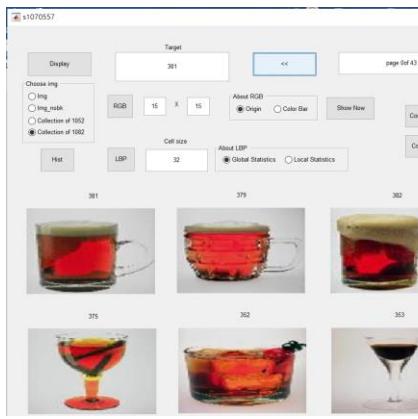
▲ target=287, p1



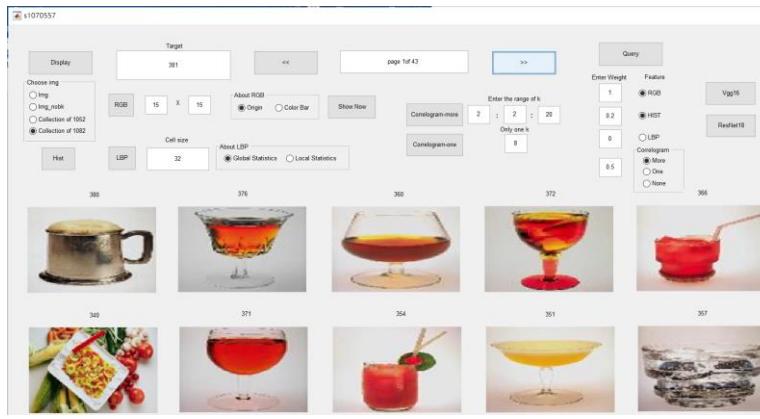
▲ target=313, p0



▲ target=313, p1

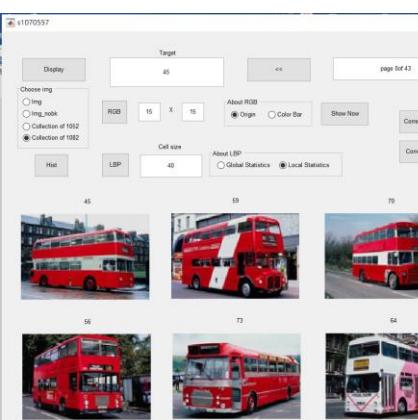


▲ target=381, p0

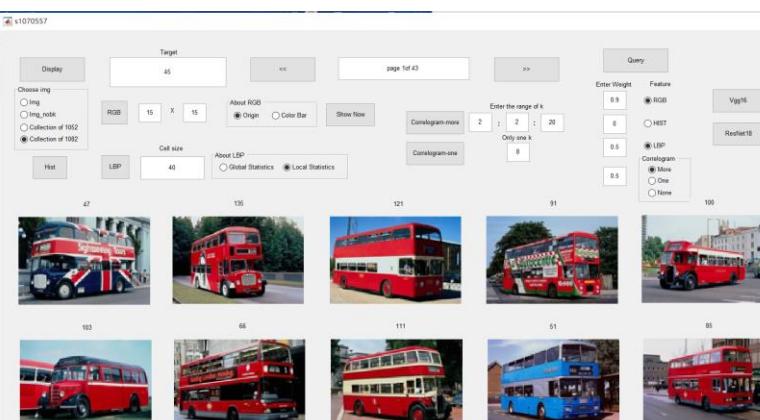


▲ target=381, p1

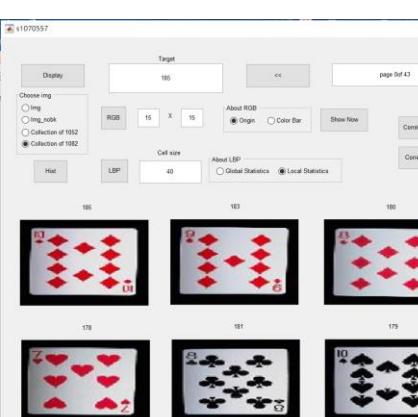
## More-features-RGB<sub>(0.9)</sub>、LBP<sub>(0.5)</sub>、Correlogram<sub>(0.5)</sub> :



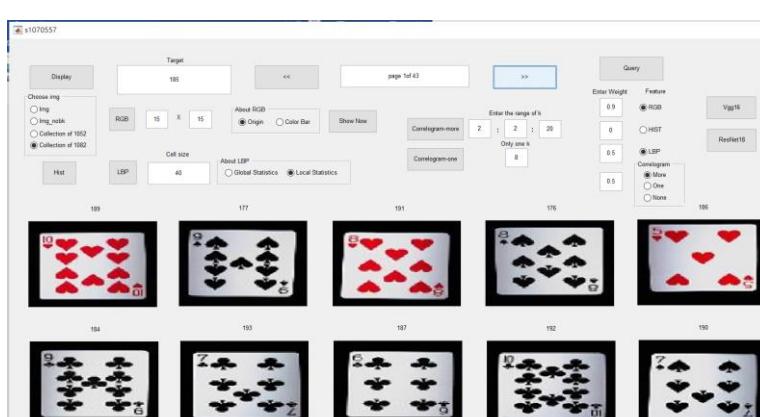
▲ target=45, p0



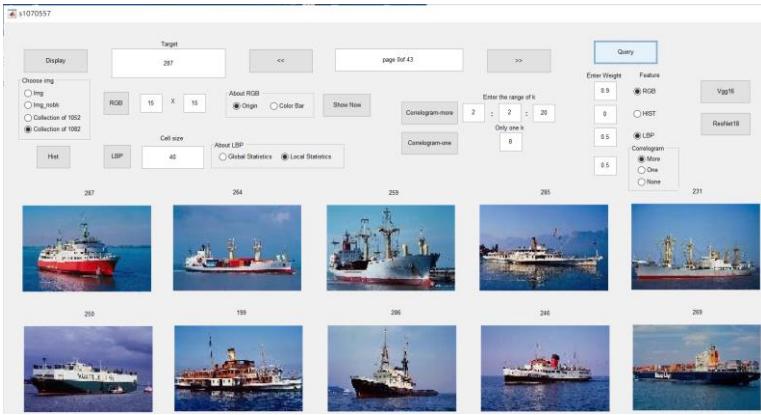
▲ target=45, p1



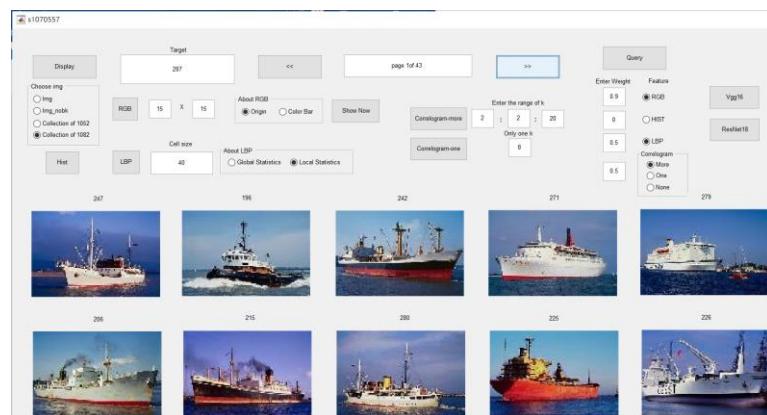
▲ target=185, p0



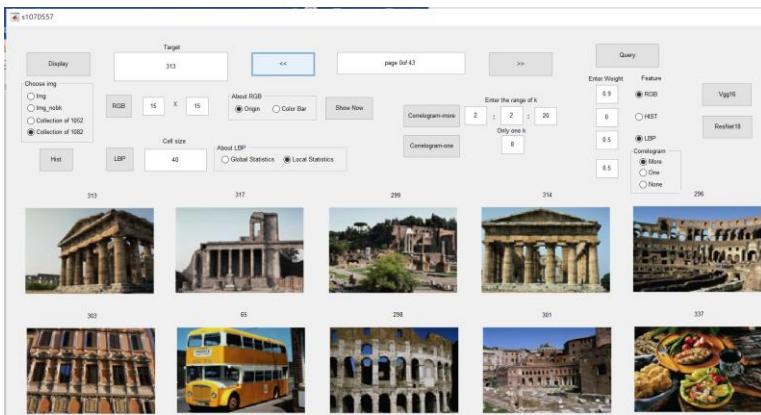
▲ target=185, p1



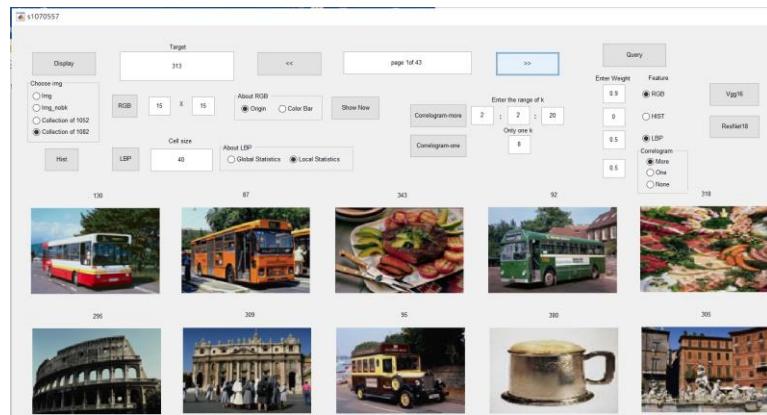
▲ target=287, p0



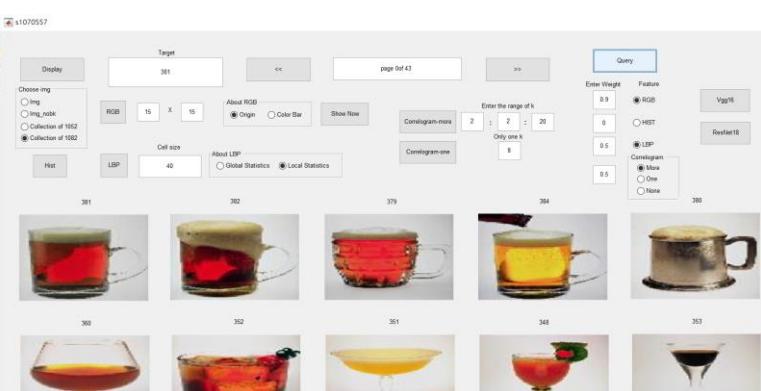
▲ target=287, p1



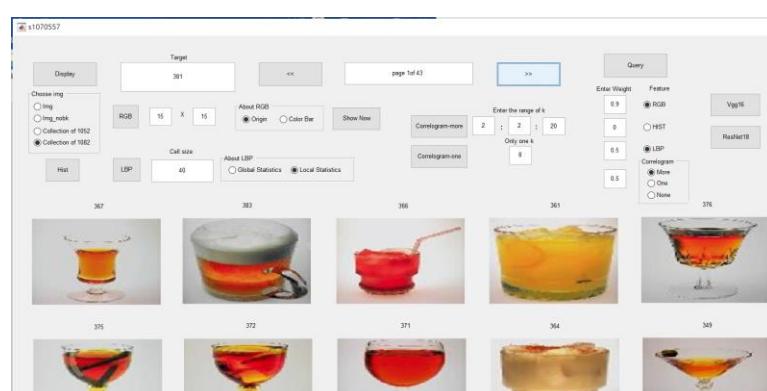
▲ target=313, p0



▲ target=313, p1



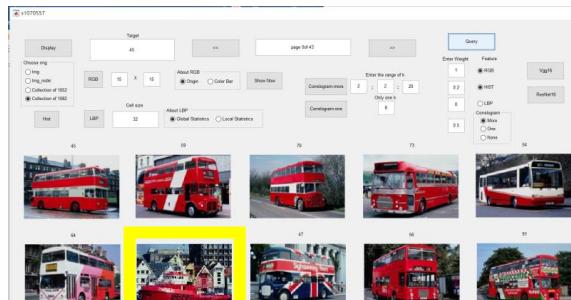
▲ target=381, p0



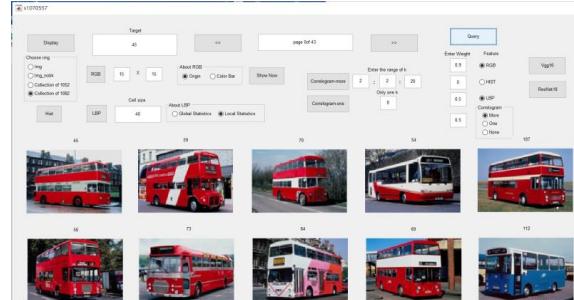
▲ target=381, p1

## More-features-Analysis :

在 More-feature 中，我採用四種任選三種的方式去做比對，一開始我利用幾乎只考慮顏色的特徵去做篩選，第一種 RGB=0.9、HIST=0.2、Correlogram=0.5，會這樣選是因為我認為從顏色下手應該比較好抓取到相似的圖片，而且在單一特徵的比對上，我認為 RGB 在五種類型的圖片上普遍都有不錯的效果，因此 RGB 的權重也會特別大，而呈現出的效果似乎也比之前只取單一種特徵來的好，像是 45-公車其實在 RGB 單一特徵是別中會篩出許多的船隻，而改用第一種 RGB=0.9、HIST=0.2、Correlogram=0.5 以後，公車幾乎都能成功篩選出，唯獨一艘背景同樣繽紛的船隻還留在前 20 名。第二種是 RGB=0.9、LBP=0.5、Correlogram=0.5，不同的是第二種考量了圖片輪廓進去，而他的效果呈現也顛覆了我認為顏色比較重要的邏輯，在第二種中，可以看到我依然將 RGB 的數值調成最高，但我讓 LBP 和 Correlogram 的比重相等而又不會占比太少，乍看之下可能覺得效果差不多，但仔細觀察可以發現因為考量了輪廓，所以船隻就不在排名前面。

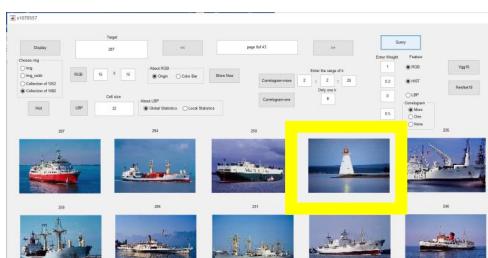


▲ 第一種，公車 45, p0

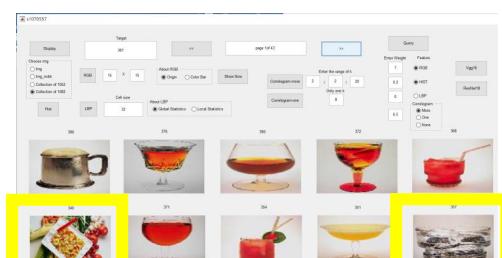


▲ 第二種，公車 45, p0

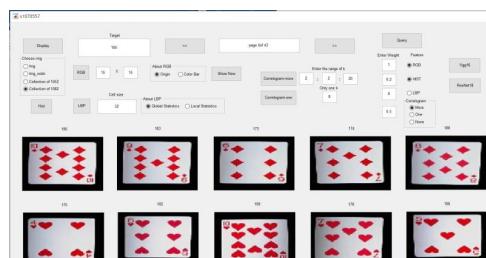
在 287-船、381-飲料中，同樣的在第二種有考量 LBP 的特徵之下，將船中誤選到的燈塔、飲料中混入的食物和白開水，都能成功往後移至 20 名之後，將更相像的圖片移到前面。但是在 185-撲克牌中，我個人是偏向第一種篩選的結果，因為我認為同樣是紅色的撲克牌排序在前面比較合理，不過兩種方法的共通點是都能將 21 張撲克牌放在前 21 名。



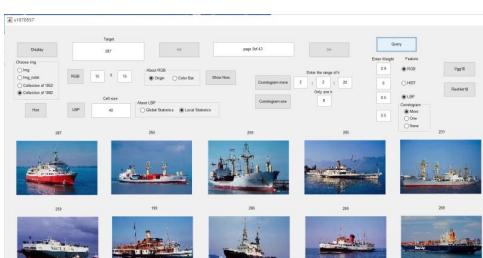
▲ 第一種，船 287, p0



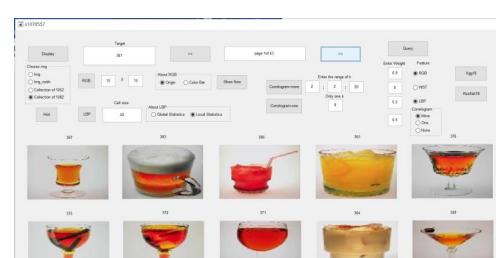
▲ 第一種，飲料 381, p1



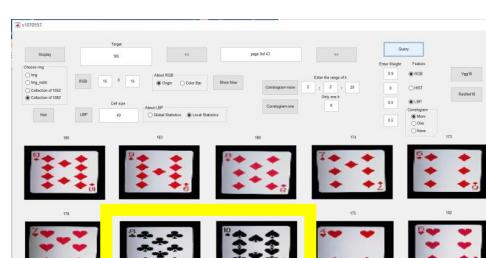
▲ 第一種，撲克牌 185, p0



▲ 第二種，船 287, p0

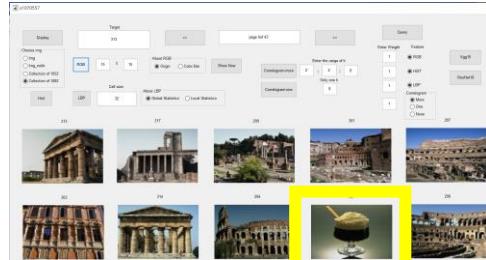


▲ 第二種，飲料 381, p1

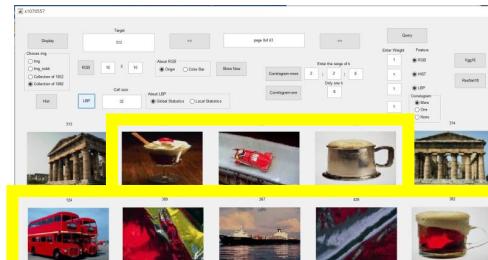


▲ 第二種，撲克牌 185, p0

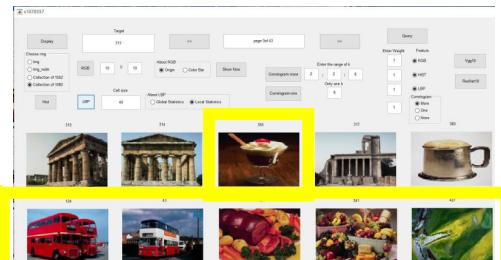
在建築的比對上，我認為在第二種是 RGB=0.9、LBP=0.5、Correlogram=0.5 中，是目前(篩選一個特徵和三個特徵中)最有明顯差異的種類，從一開始 RGB 只能抓到部分相近顏色的建築、HIST 則是普遍抓不到建築種類的圖片、LBP 能以輪廓去抓到些許極相似的圖片並排名在前、Correlogram 則亦是普遍抓不大到(也可能是 k 值的設定不適合建築種類的分析)。在第一種和第二種的權重微調後，許多建築物都能準確抓取出來，或許還沒有像分析其他 4 種類型圖片來的完美，但比起先前已大幅進步。



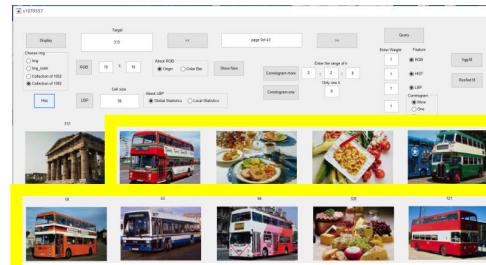
▲ RGB, 建築 313, p0



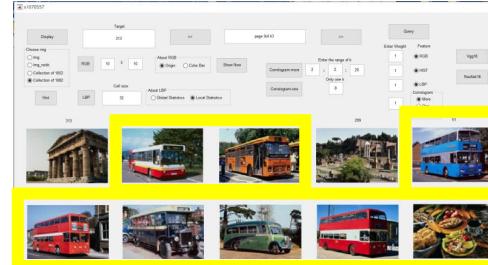
▲ LBP-Global, 建築 313, p0



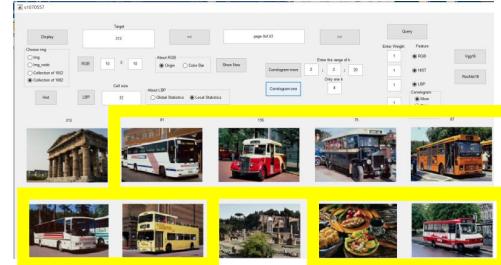
▲ Cor-More, 建築 313, p0



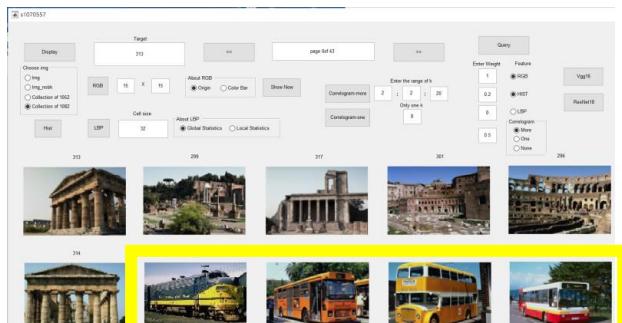
▲ HIST, 建築 313, p0



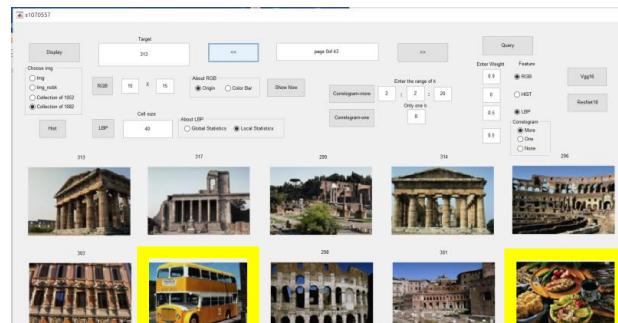
▲ LBP-Local, 建築 313, p0



▲ Cor-One, 建築 313, p0



▲ 第一種, 建築 313, p0



▲ 第二種, 建築 313, p0

整體而言，在考慮多種特徵之後，會發現到 LBP 是不可少的一種特徵，雖然在單一使用它時看不出來，但是在綜合結果上會發現，其實輪廓也是很重要的一個特徵，而第二種的結果比第一種來得更好。

## All-features-A

$RGB_{(0.9)}$ 、 $HIST_{(0.15)}$ 、 $LBP_{(0.5)}$ 、 $Correlogram_{(0.2)}$ ：

▲ 權重 A-公車 45, p0

## All-features-B

$RGB_{(1)}$ 、 $HIST_{(0.1)}$ 、 $LBP_{(0.7)}$ 、 $Correlogram_{(0.4)}$ ：

▲ 權重 B-公車 45, p0

▲ 權重 A-公車 45, p1

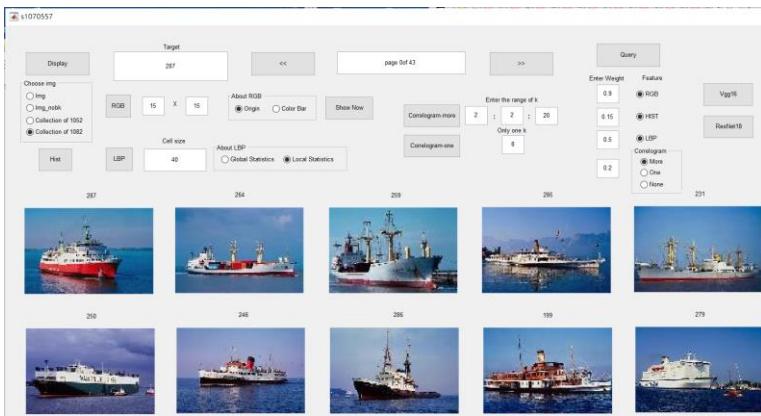
▲ 權重 B-公車 45, p1

▲ 權重 A-撲克牌 185, p0

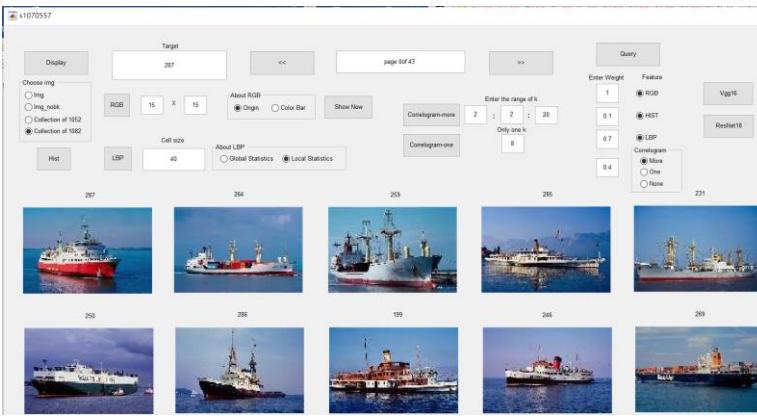
▲ 權重 B-撲克牌 185, p0

▲ 權重 A-撲克牌 185, p1

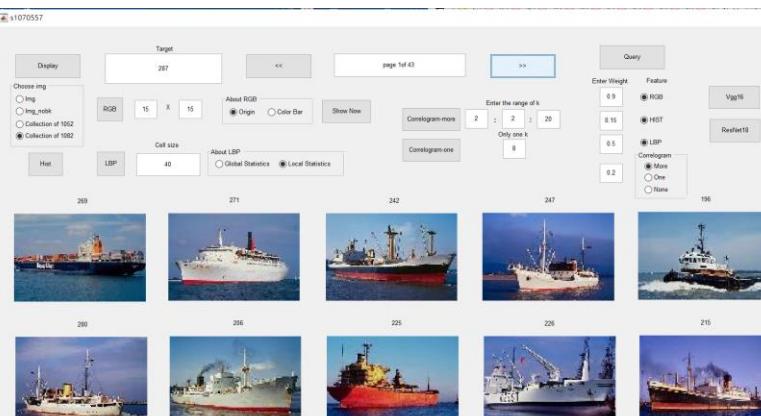
▲ 權重 B-撲克牌 185, p1



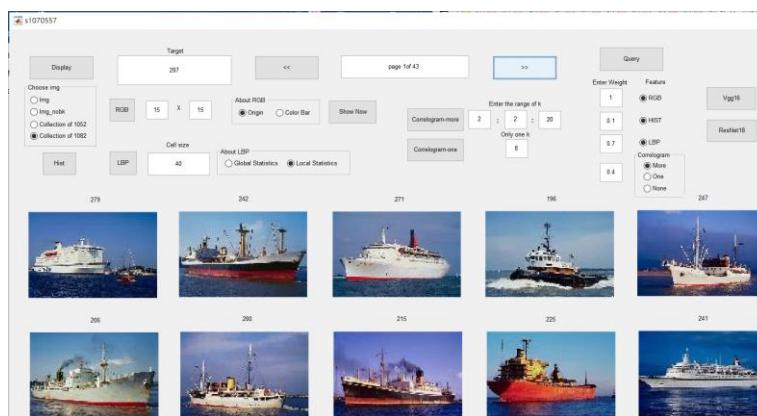
▲ 權重 A-船 287, p0



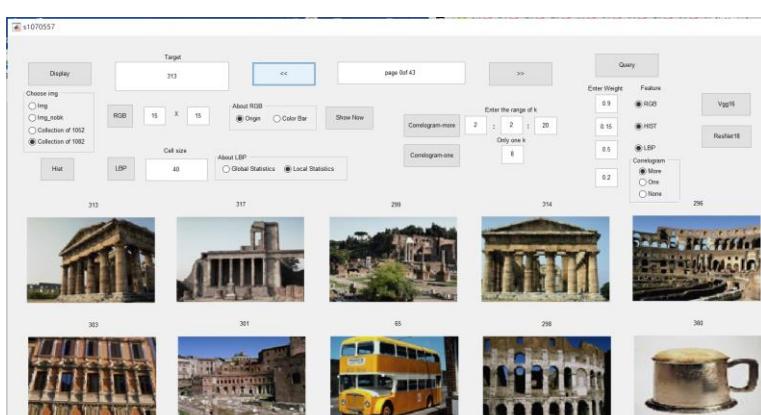
▲ 權重 B-船 287, p0



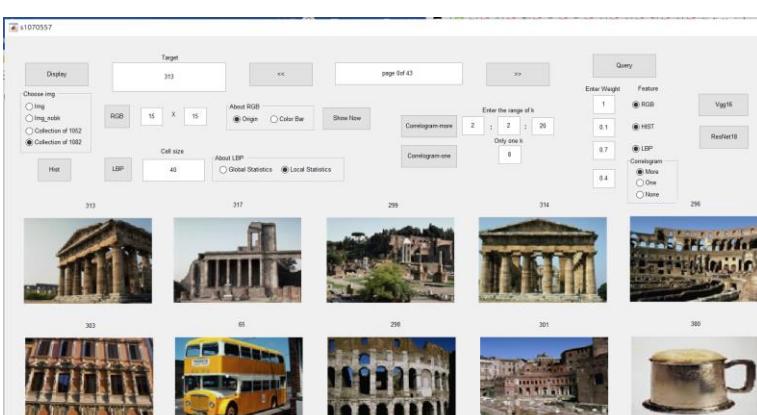
▲ 權重 A-船 287, p0



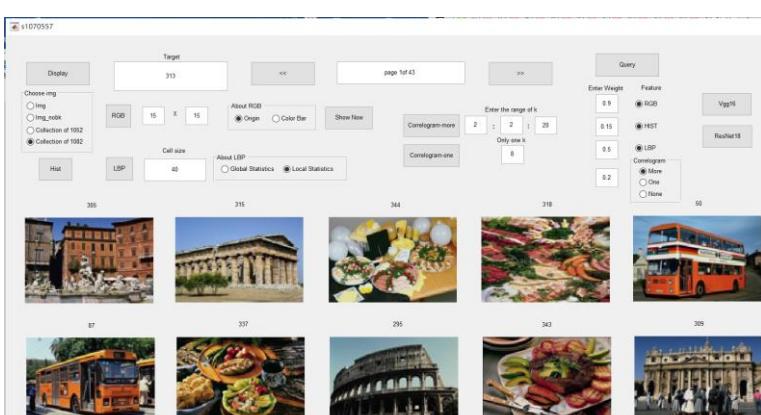
▲ 權重 B-船 287, p0



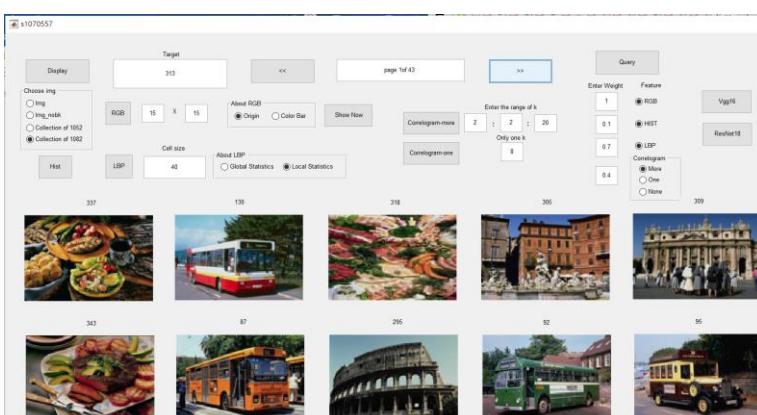
▲ 權重 A-建築 313, p0



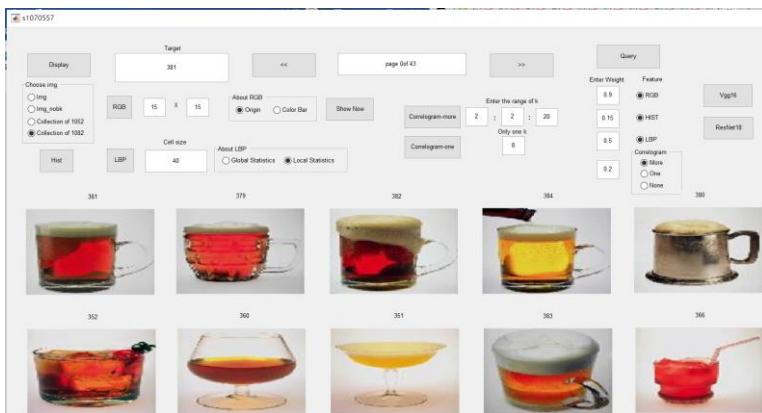
▲ 權重 B-建築 313, p0



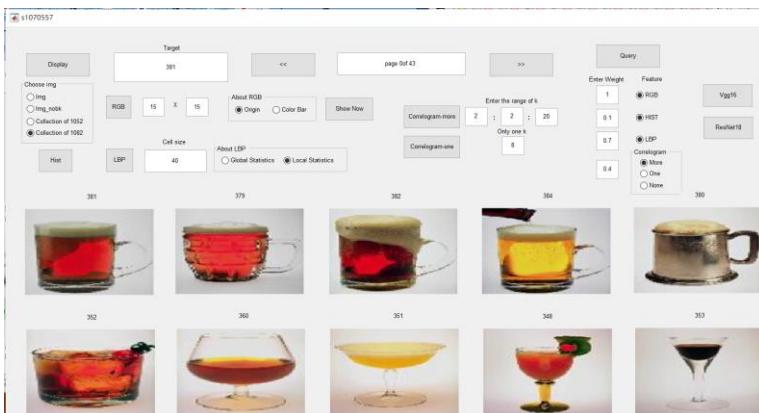
▲ 權重 A-建築 313, p1



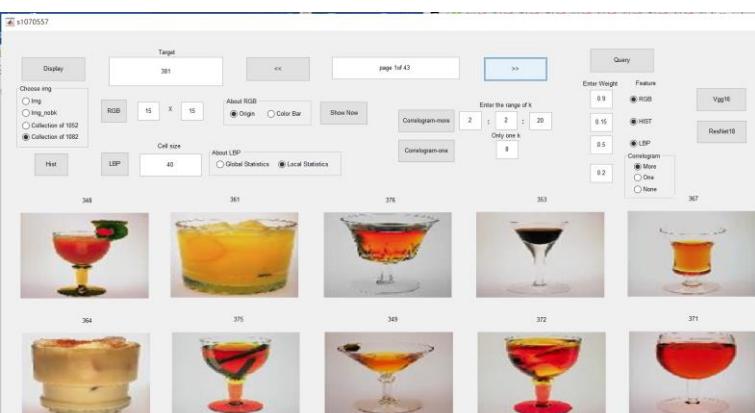
▲ 權重 B-建築 313, p1



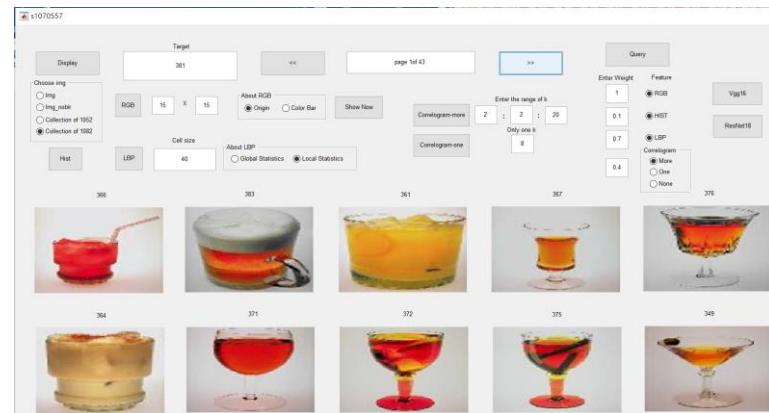
▲ 權重 A-飲料 381, p0



▲ 權重 B-飲料 381, p0



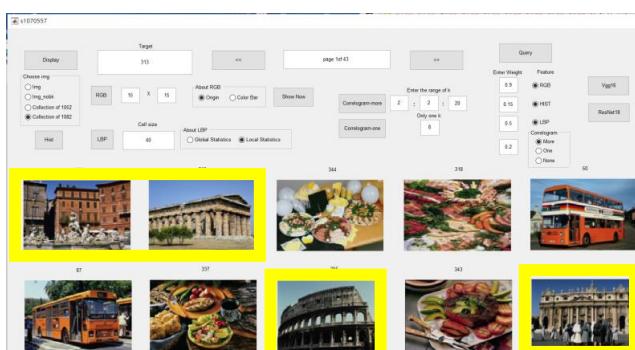
▲ 權重 A-飲料 381, p1



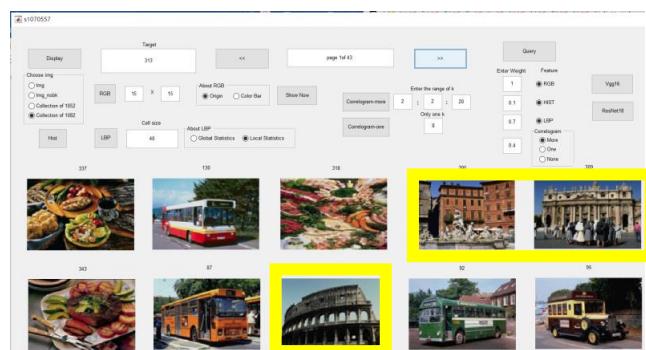
▲ 權重 B-飲料 381, p1

## All-features-Analysis :

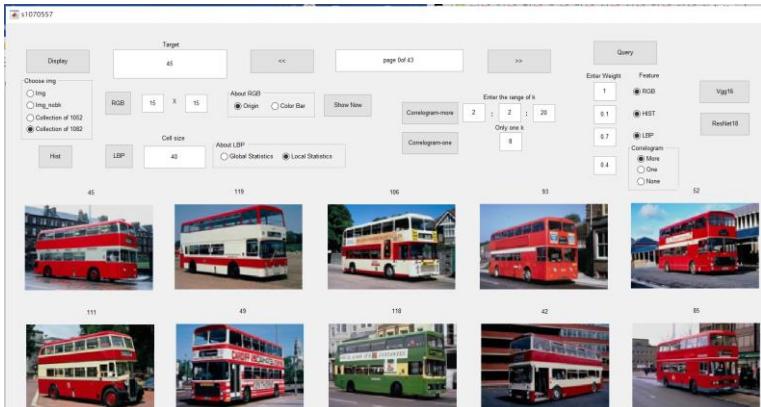
由權重 A 和 權重 B 可以發現，權重 A : RGB=0.9、HIST=0.15、LBP=0.5、Correlogram=0.2 和 權重 B RGB=1、HIST=0.1、LBP=0.7、Correlogram=0.4 的相同處是我一樣都將 RGB 的權重調最高、HIST 最低，之後再調整 LBP 和 Correlogram 對整體的影響力，兩者其實還是有些許的落差，兩者權重的前五名幾乎都相同，而在 45-公車、185 撲克牌、287-船、381-飲料上其實非常相像，比較不同的是在建築的篩選第 1 面，雖然無法抓出所有建築物，但是第 11~20 排名中若是有建築，在權重 B 中卻無法排序在前面，我認為是因為 LBP 的影響力過大，雖然能抓取到但是可能在 LBP 特徵中它的分數並不高，因此雖然 LBP 很有用，但也是一昧的去加大權重，故權重 A 的結果相較權重 B 來的好。



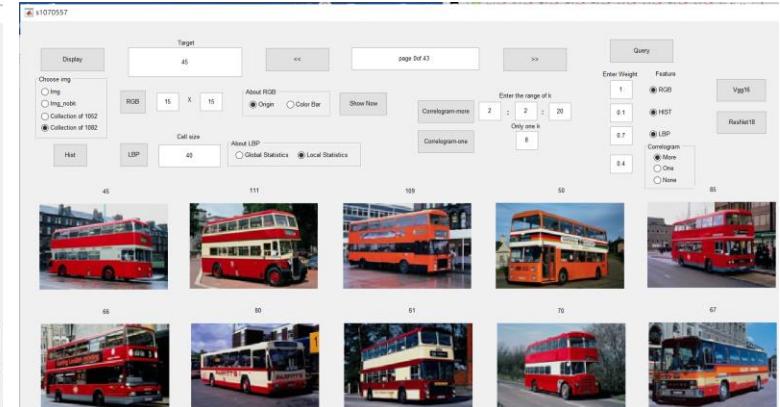
▲ 權重 A-建築 313, p1



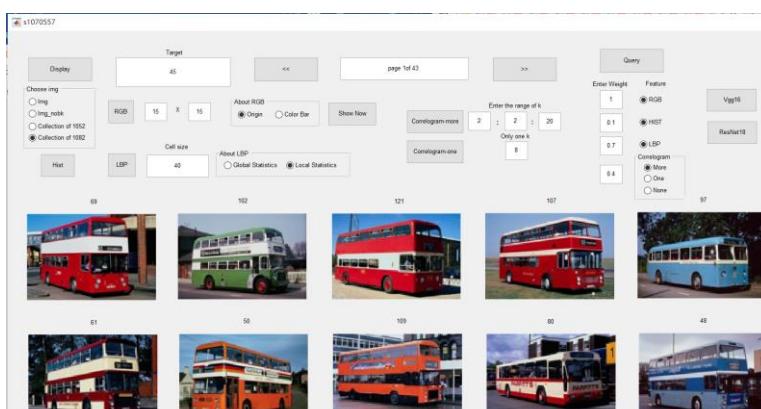
▲ 權重 B-建築 313, p1

*Collection\_for\_1082-VGG16*

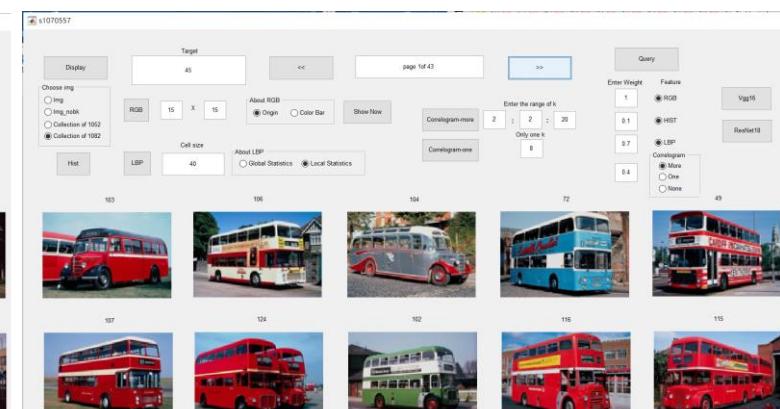
▲ VGG16-公車 45, p0

*Collection\_for\_1082-ResNet18*

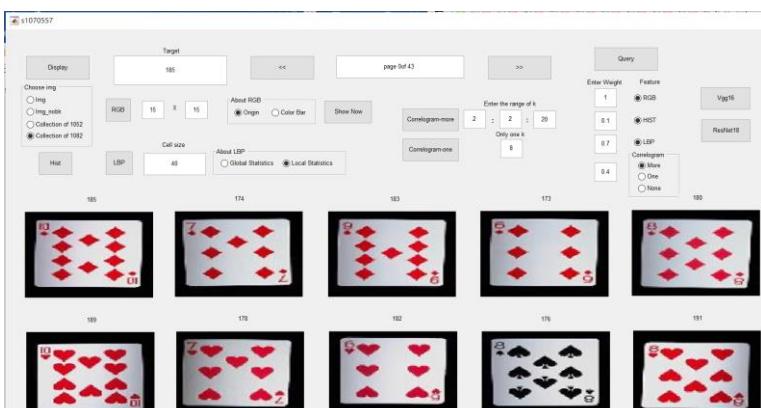
▲ ResNet18-公車 45, p0



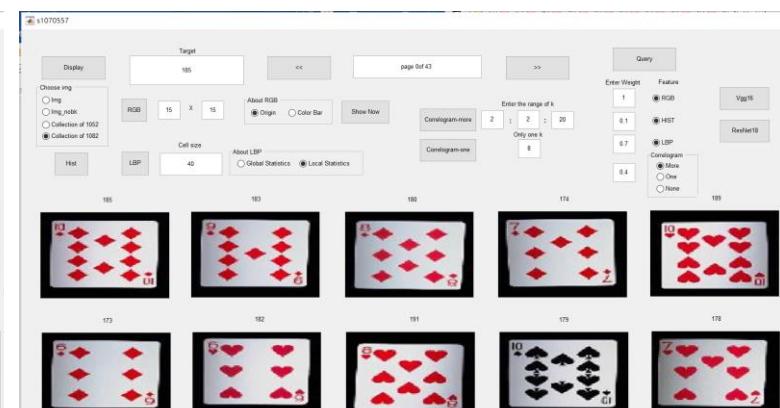
▲ VGG16-公車 45, p1



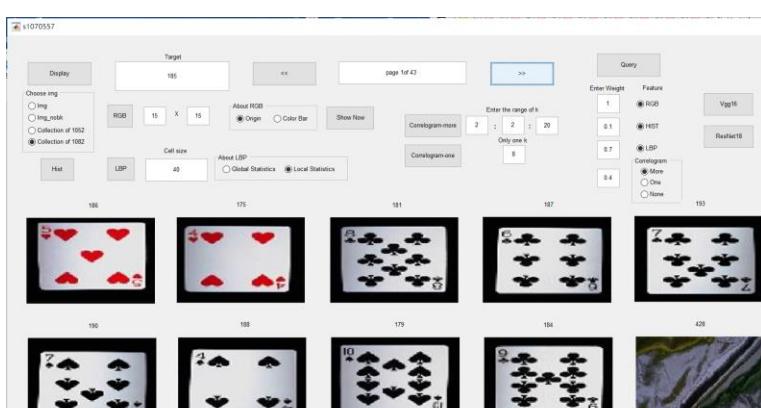
▲ ResNet18-公車 45, p1



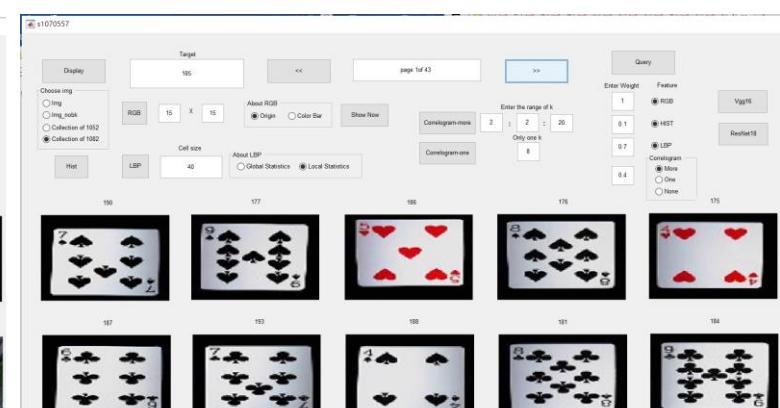
▲ VGG16-撲克牌 185, p0



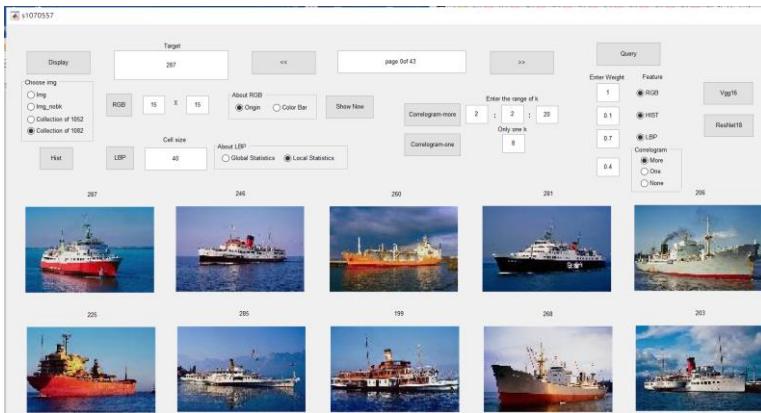
▲ ResNet18-撲克牌 185, p0



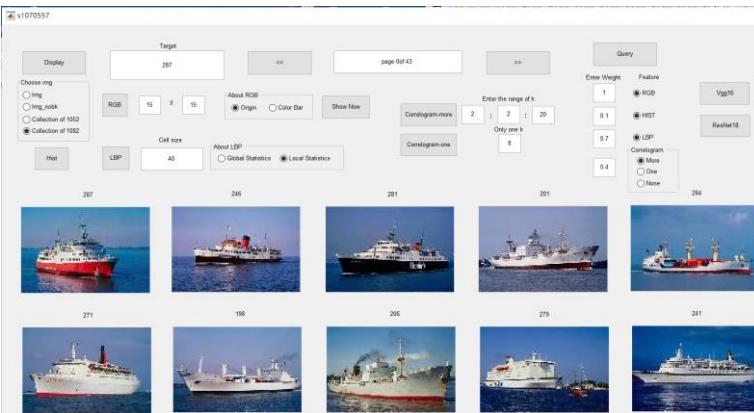
▲ VGG16-撲克牌 185, p1



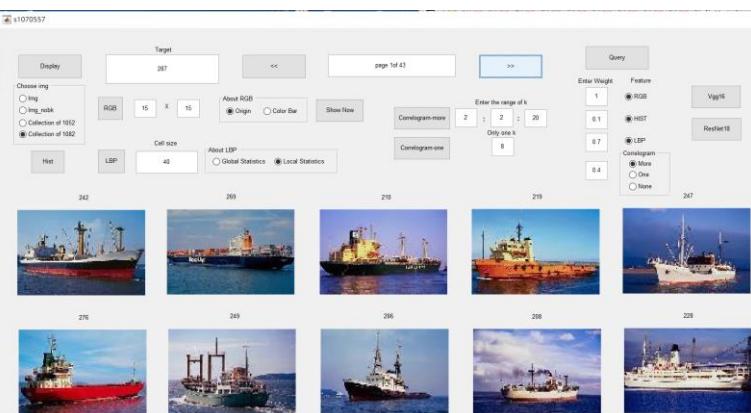
▲ ResNet18-撲克牌 185, p1



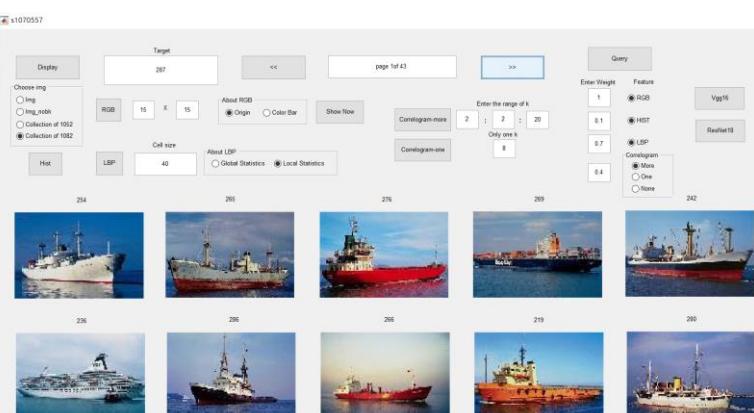
▲ VGG16-船 287, p0



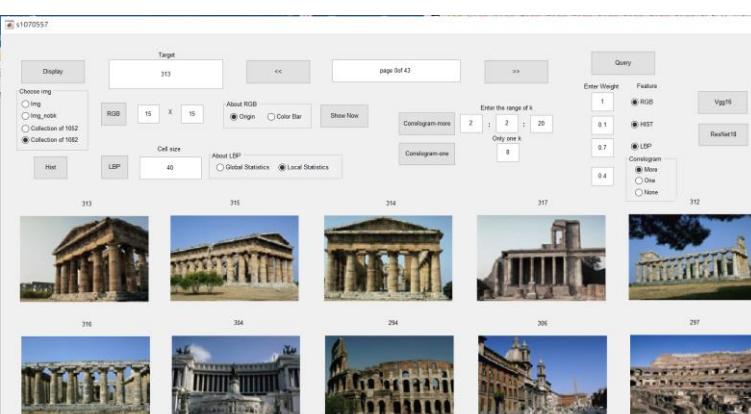
▲ ResNet18-船 287, p0



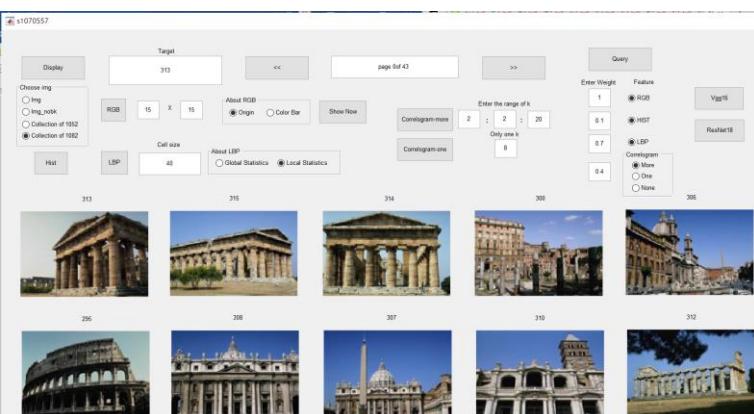
▲ VGG16-船 287, p1



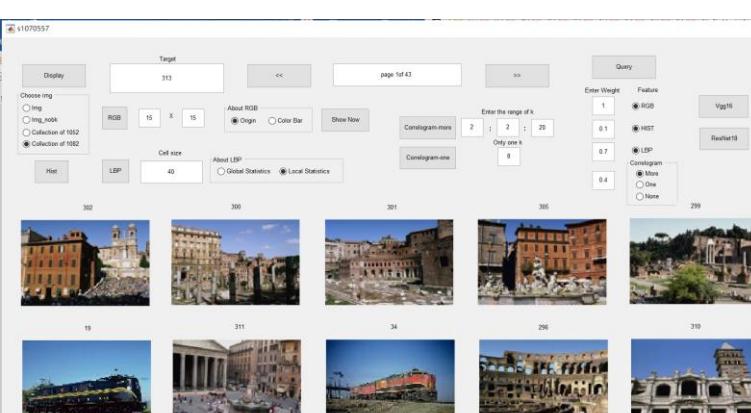
▲ ResNet18-船 287, p1



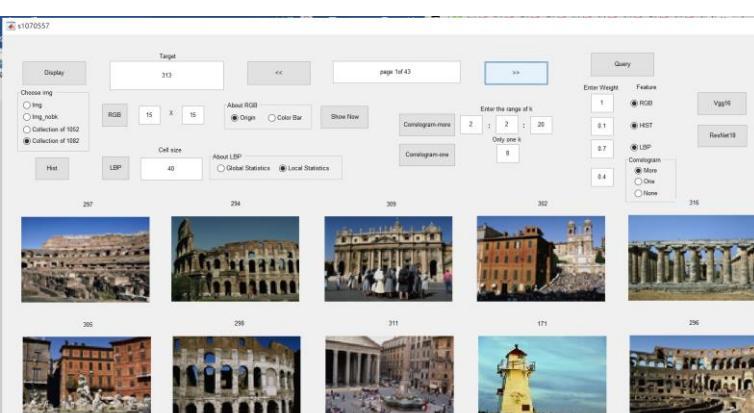
▲ VGG16-建築 313, p0



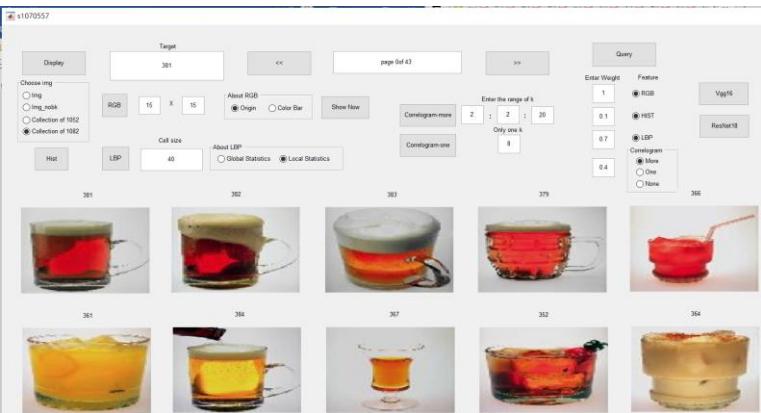
▲ ResNet18-建築 313, p0



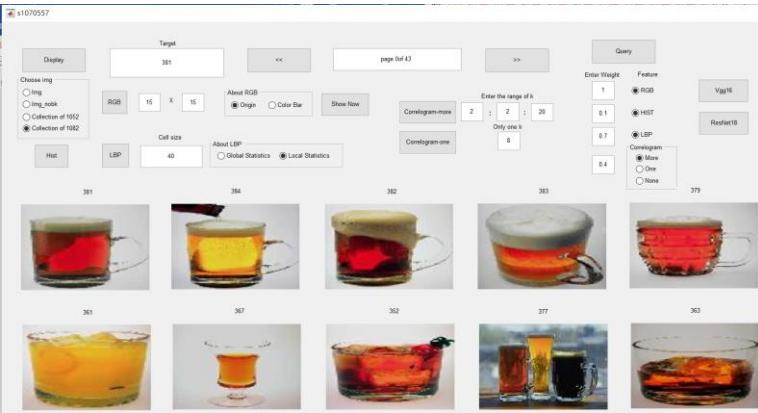
▲ VGG16-建築 313, p1



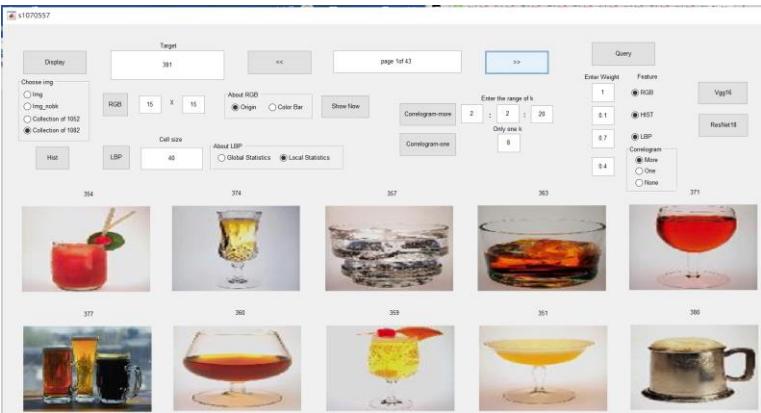
▲ ResNet18-建築 313, p1



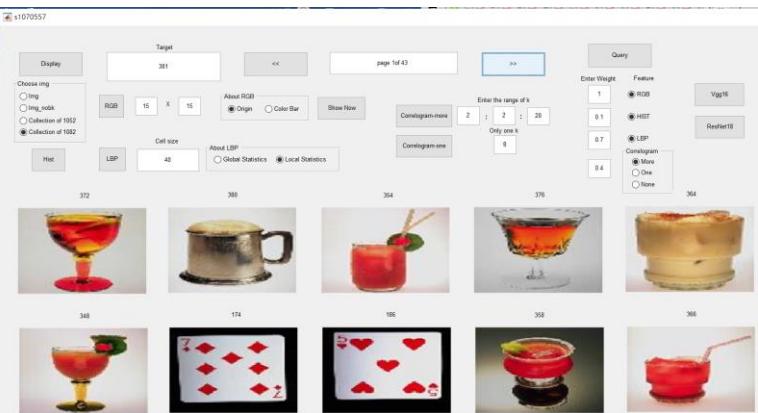
▲ VGG16-飲料 381, p0



▲ ResNet18-飲料 381, p0



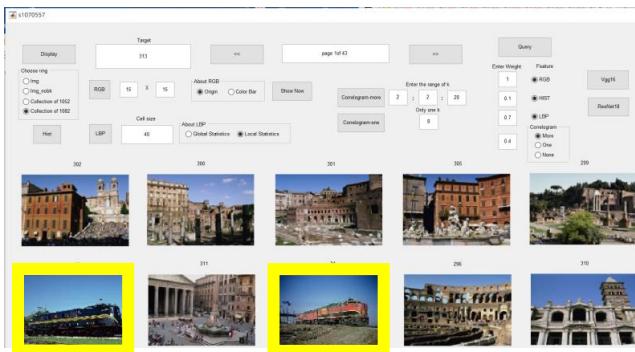
▲ VGG16-飲料 381, p1



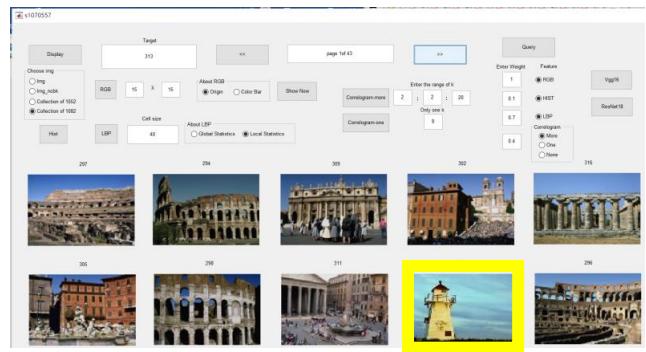
▲ ResNet18-飲料 381, p1

## VGG16 & ResNet18 -Analysis

比較 VGG16 和 ResNet18 兩者 pretrained model，其中在 VGG16 我使用的是' fc6' 而再 ResNet18 使用的是' fc1000' 做特徵值比較的部分，覺得普遍而言，似乎是 VGG16 的效果較好。在 45-公車中，我們可以發現到兩者的分析細到可以也將條紋的公車排在前面，若不仔細看會覺得其實差不多，而他們在 287-船隻的表現中也幾乎一樣，能抓出數艘相似的船隻。185-撲克牌上的表現也大致相同，只是 VGG16 在排名 20 的部分出現了一張毫不相關的圖片。然而在 313-建築和 381-飲料上的差異就比較明顯了，尤其是第 1 面(11~20 名)入選的照片都不盡相同，神奇的是透過 ResNet18 的多層特徵提取，我們最後結算對比 381-飲料的特徵值時，撲克牌會參雜於其中，再更後面一點的頁數也是如此(撲克牌參雜其中)，VGG16 則是到第 2 面才出現撲克牌，在 313-建築中反而是 ResNet18 稍好一點，整體而言我認為單只看 1~10 名的話，我是偏向 ResNet18，但若是考慮 1~20 名，我認為是 VGG16 較好。



▲ VGG16-建築 313, p1



▲ ResNet18-建築 313, p1

## Summary

就上述所有實驗而言，

總共用了 12 種方式去對圖片做篩選排序。其中若要從每一大類比較的話我認為是如此的：

One-Feature	More-Feature	Others
RGB 	RGB + HIST + Cor-more-k (W : 0.9, 0.2, 0.5)	VGG 16 
HIST	RGB + LBP-local + Cor-more-k  (W : 0.9, 0.5, 0.5)	ResNet 18
LBP-global	<b>All-Feature</b>	
LBP-local	RGB + HIST + LBP-local + Cor-more-k  (W : 0.9, 0.15, 0.5, 0.2)	
Cor-more-k	RGB + HIST + LBP-local + Cor-more-k (W : 1, 0.1, 0.7, 0.4)	
Cor-one-k		

▲ 所有的方法分類

五種類型的圖片考驗著不同的篩選能力，像是：

### 45-公車：

公車的款式、顏色有許多種，也有車頭向左或向右的，而街景大部分所占空間少但是可能是雜亂的。

### 185-撲克牌：

撲克牌其幾乎都長得一模一樣，最大的差異就是在花色的部分，有四種圖案和兩種顏色，圖案和顏色之間的取捨在不同特徵下的選擇不同。

### 287-船：

和少部分的火車、燈塔圖片同樣有著湛藍的背景，而有的燈塔可能和船隻相同有白色的顏色，造成若不使用 LBP 去考量輪廓，雖然會篩選出類似的圖片，但主體可能不會是船隻。

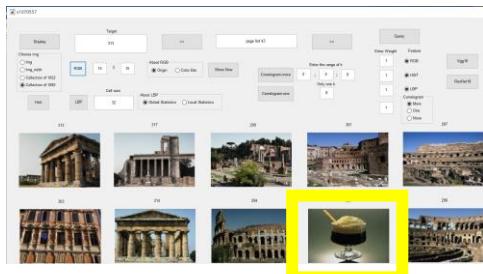
### 313-建築：

近乎沒有建築是完全相同的，就算有隨著拍攝角度的不同也會導致輪廓的差異極大，也因為如此才會在只採取一個特徵時 RGB 的顏色較優秀(建築色都是石頭色)。

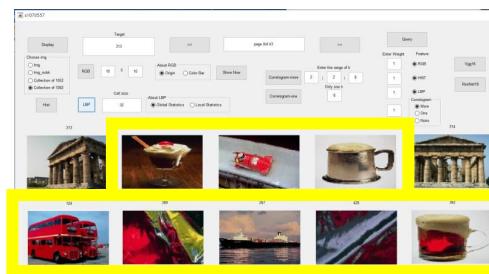
### 381-飲料：

大部分飲料的圖片背景都為純白色，而杯子的款式有許多種，裡面所裝的飲料顏色也不一樣，有的可能多了泡沫但是識別上基本上會比其他種類簡單一點。

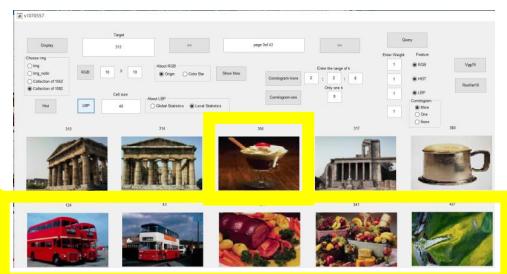
其中最感動的地方在於 313-建築的分類在一開始其實一直處於極糟的狀態，但越後面考慮越多不同特徵和特徵的混合，逐漸能篩選出我們想要的結果了，而其他類型的圖片也有進步，但是建築還是進步最多的。



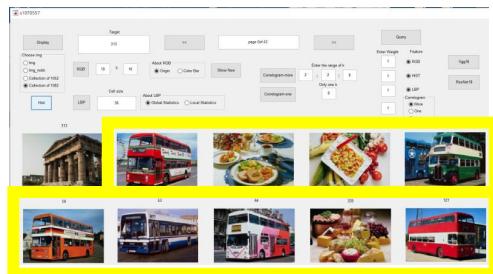
▲ RGB, 建築 313, p0



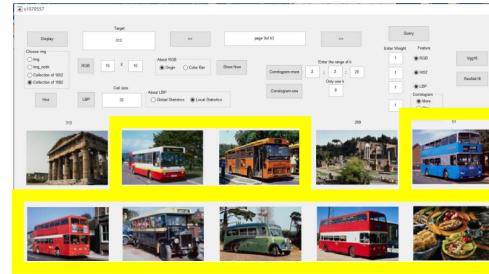
▲ LBP-Global, 建築 313, p0



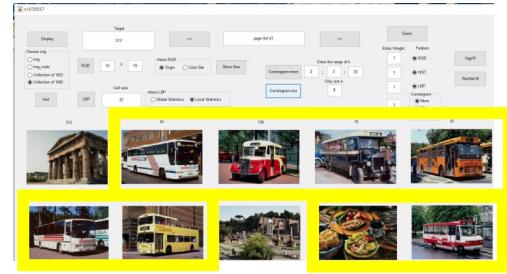
▲ Cor-More, 建築 313, p0



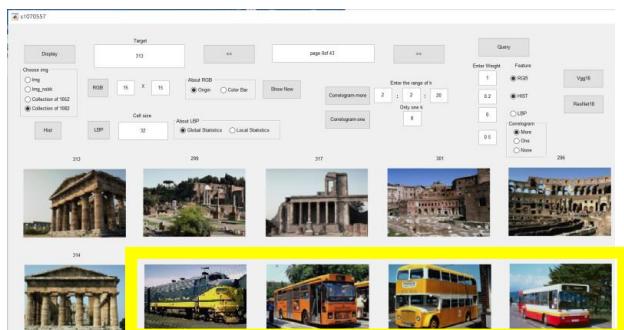
▲ HIST, 建築 313, p0



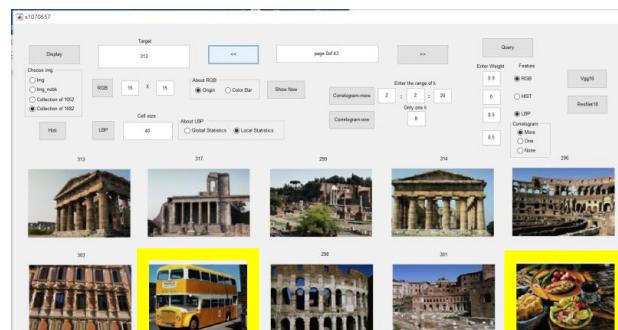
▲ LBP-Local, 建築 313, p0



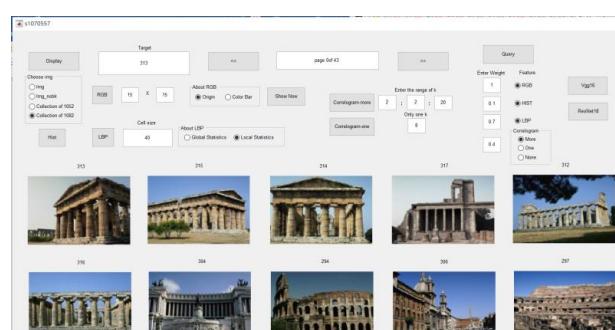
▲ Cor-One, 建築 313, p0



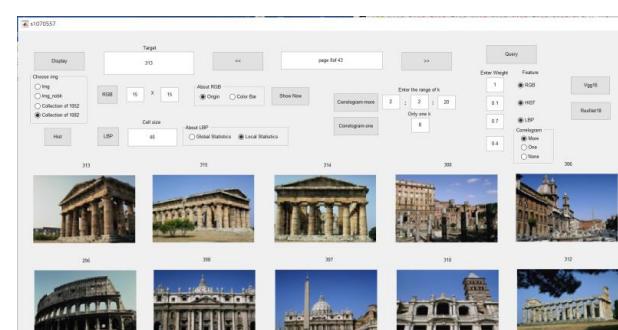
▲ 第一種, 建築 313, p0



▲ 第二種, 建築 313, p0



▲ VGG16-建築 313, p0



▲ ResNet18-建築 313, p0

藉由不斷的 try and error 能摸索出個數值的調整，最大的缺點是因為調整的過程中，相像的答案其實並非絕對的，可能包含了許多主觀看法，不一定每個人看法都相同，但我認為若不借助 VGG16 和 ResNet18 的幫助的話，是權重 A : RGB=0.9、HIST=0.15、LBP=0.5、Correlogram=0.2 的表現最好。而若是單獨去比較 VGG16 和 ResNet18 的話我會偏向 VGG16 因為它能讓 1~20 名出錯的機率較小，ResNet18 則是認為相像就有可能往前排造成可能不一樣的圖片依然排在前方，VGG16 則會將他們至於 30 名之後，看起來錯誤較小。

One-Feature	Advantages	Disadvantages
RGB	準確找到對應位置有相似顏色的圖片	可能同顏色但不同種類的圖片被納入
HIST	若篩選不複雜的圖片，有不錯的結果	似 RGB 但使用方法較粗略
LBP-global	公車、船等特定明顯主角的圖片能輕易抓取	對於局部特徵能力較弱
LBP-local	注重局部特徵，調整至合適的 CellSize 能呈現比 global 更好的效果	CellSize 數值對結果影響很大，要花時間嘗試
Cor-more-k	在適當的 k 之下，能準確抓出顏色分布相似的圖片	範圍若取不好，對結果有極大的影響，且因為考量數值多，計算較久
Cor-one-k	比起 more-k 計算速度來得更快，且 1~10 名有機會達到不錯的結果	因為只有一個 k 值能參考，不一定每種圖片最合適的 k 都一樣
VGG16	整體看起來不錯	無
ResNet18	整體看起來不錯	少數時候可能有看不出相像的圖片出現

### ▲ One-feature 優缺分析

More-Feature	Advantages	Disadvantages
$RGB(0.9)$ $+ HIST(0.2)$ $+ Cor-more-k(0.5)$	在相似圖片都擁有類似顏色的前提下，能輕鬆準確地抓出，	像是船和燈塔可能一樣背景都是藍色，而主體為白色時，容易抓錯
$RGB(0.9)$ $+ LBP-local(0.5)$ $+ Cor-more-k(0.5)$	在考量 LBP 後，可以更準確的抓出建築輪廓 提升只考量顏色而未考慮主體的結果	無較大缺點

### ▲ More-feature 優缺分析

All-Feature	Advantages	Disadvantages
$RGB(0.9)$ $+ HIST(0.15)$ $+ LBP-local(0.5)$ $+ Cor-more-k(0.2)$	普遍五個種類的圖片都能有不錯的表現	無較大缺點
$RGB(1)$ $+ HIST(0.1)$ $+ LBP-local(0.7)$ $+ Cor-more-k(0.4)$	普遍五個種類的圖片都能有不錯的表現	LBP 比重較重，在建築等圖片構圖較複雜時，容易導致誤導，造成同樣構圖複雜不同種類都排名前面

### ▲ All-feature 優缺分析