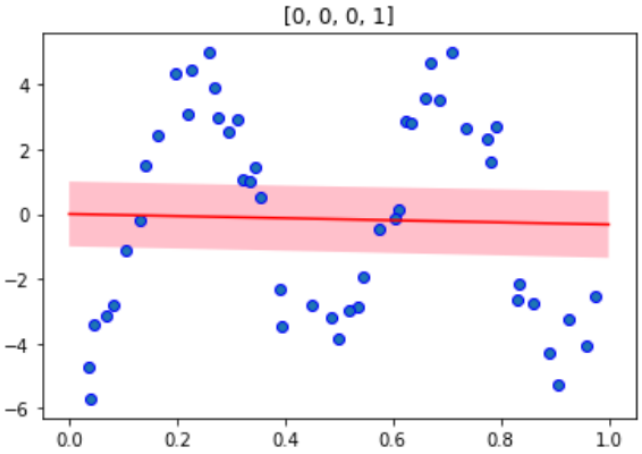
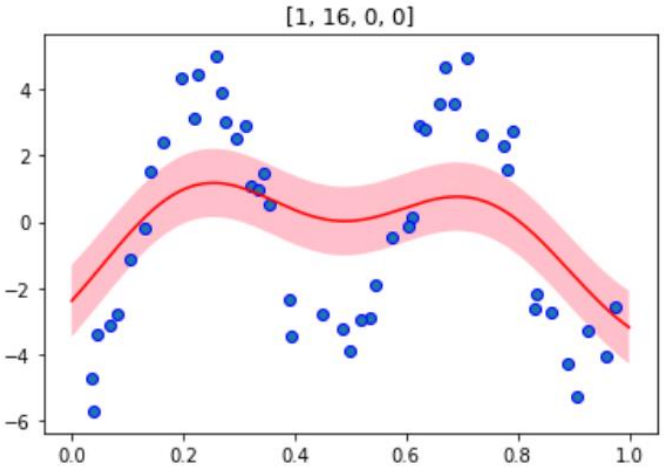
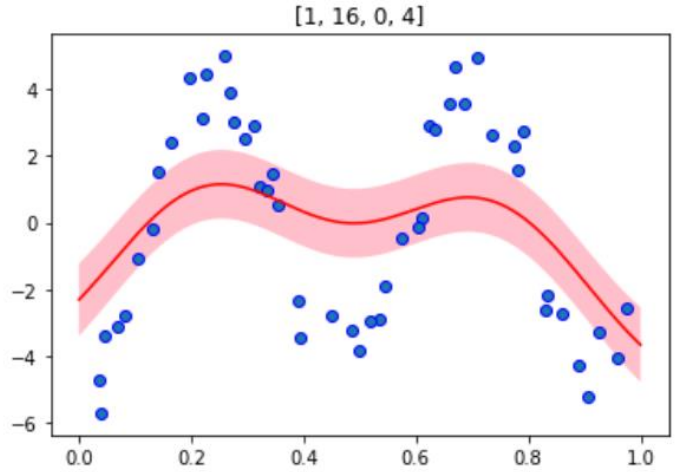
309513061 電信甲 簡詠倫

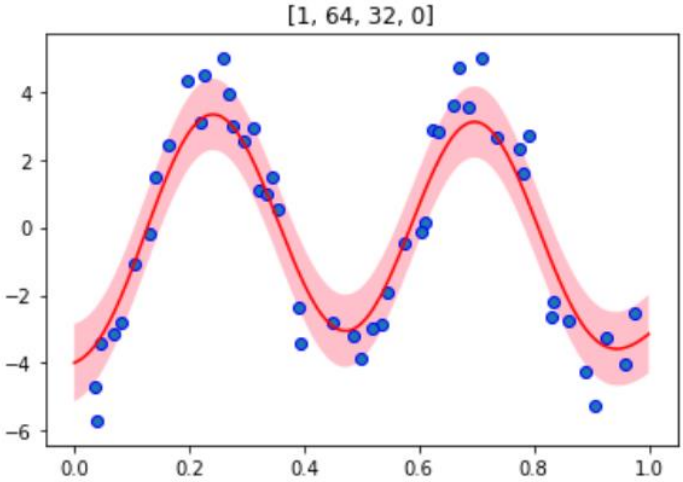
1. Gaussian Process for Regression

(1) (2) Plot the prediction result









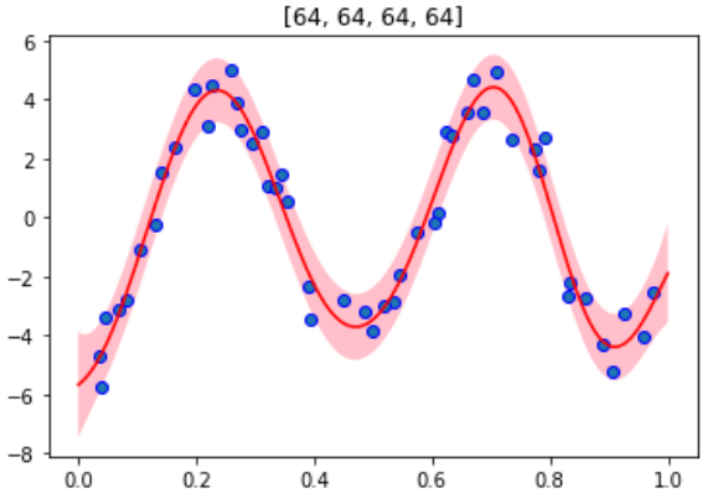
(3) Show the corresponding root-mean-square errors

```
[0, 0, 0, 1]
Training RMSE = 3.1292014298222437
Testing RMSE = 3.3443986601861146

[1, 16, 0, 0]
Training RMSE = 2.4239279278312194
Testing RMSE = 2.668051750252446

[1, 16, 0, 4]
Training RMSE = 2.4105764871252053
Testing RMSE = 2.656998000166914
```

```
[1, 64, 32, 0]
Training RMSE = 1.0428861621832162
Testing RMSE = 1.1627590936118706
```

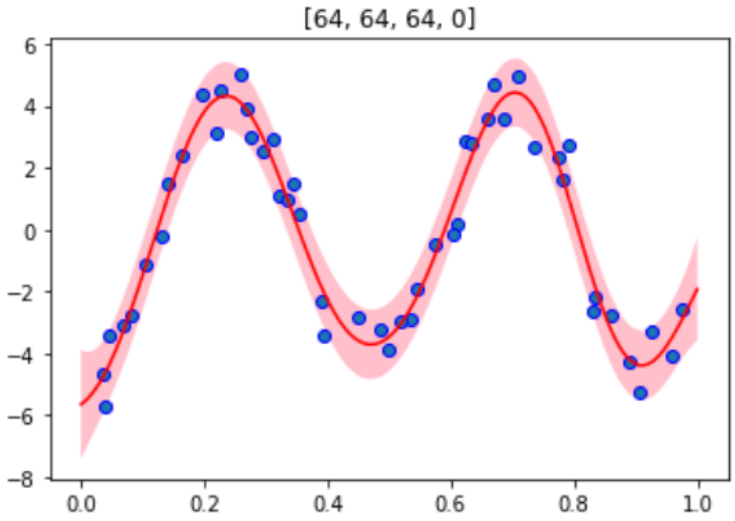
(4)

我使用的是 trial and error 來調整看看超參數。我先把全部的參數都調成 64 嘗試看看，結果如下。



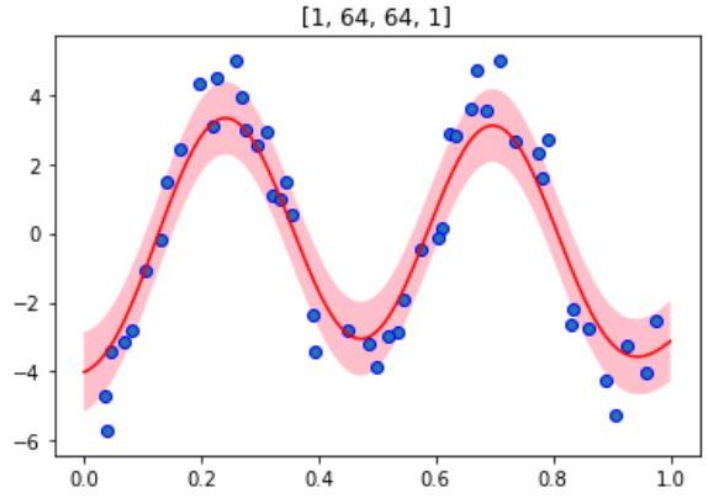

```
[64, 64, 64, 64]
Training RMSE = 0.6989589734393324
Testing RMSE = 1.0781207291422465
```

可以發現結果表現並不差。而從上題可以發現，助教給的第四個參數通常都很小，因此我懷疑它不是個重要的參數，調成 0 來嘗試看看，結果如下。



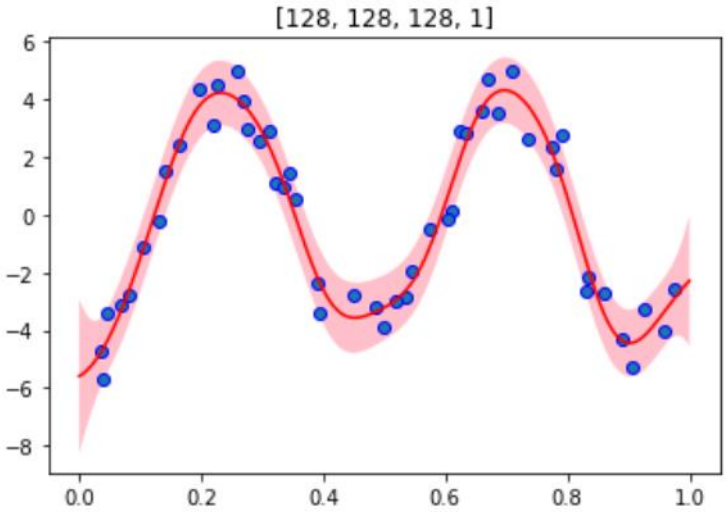

```
[64, 64, 64, 0]
Training RMSE = 0.6990244741065749
Testing RMSE = 1.0767940009808907
```

發現結果是差不多的，但根據之前所學，他很有可能是 regularization 項，因此我決定把它都設為 1。同樣的，第一個參數似乎也都不大，我嘗試看看把它設成 1，結果如下。



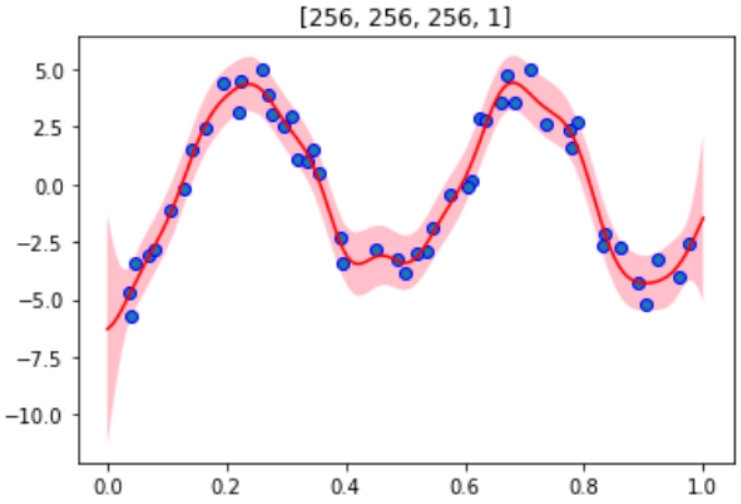

```
[1, 64, 64, 1]
Training RMSE = 1.041891402963529
Testing RMSE = 1.1618014328537425
```

結果表現變得比較差，所以我決定還是將前三個參數一起調整。而現在嘗試看看調成更大的數 128，結果如下。




```
[128, 128, 128, 1]
Training RMSE = 0.6739860328758832
Testing RMSE = 1.076624364195028
```

結果似乎又更好了一點，若是再調大數字，或許結果會更好，我再將數字調成 256，結果如下。




```
[256, 256, 256, 1]
Training RMSE = 0.6134016439363199
Testing RMSE = 1.189664930176389
```

可發現產生了 over-fitting 的現象，代表參數也不能設太大。於是我選用[128,128,128,0]作為模型較好的超參數。

(5) Explain your findings and make some discussion.

藉由實際調整參數可發現其對模型影響的重要程度，若參數太小很難去 fit training data，而太大又會造成 over-fitting 的結果，引此求得最佳超參數十分重要。第四個超參數看起來雖然不重要，但根據之前所學，或許它具有 regularization 的功能，避免模型過度擬合。

2. Support Vector Machine

(1)

one-versus-the-rest:

訓練時將某個類別的樣本歸為一類，其餘樣本歸為另一類。因此有 K 個類別就建構出 K 個 SVM 模型。分類時再計算樣本與每類的相關值，最大即為該類。此種方法可能存在 Bias。

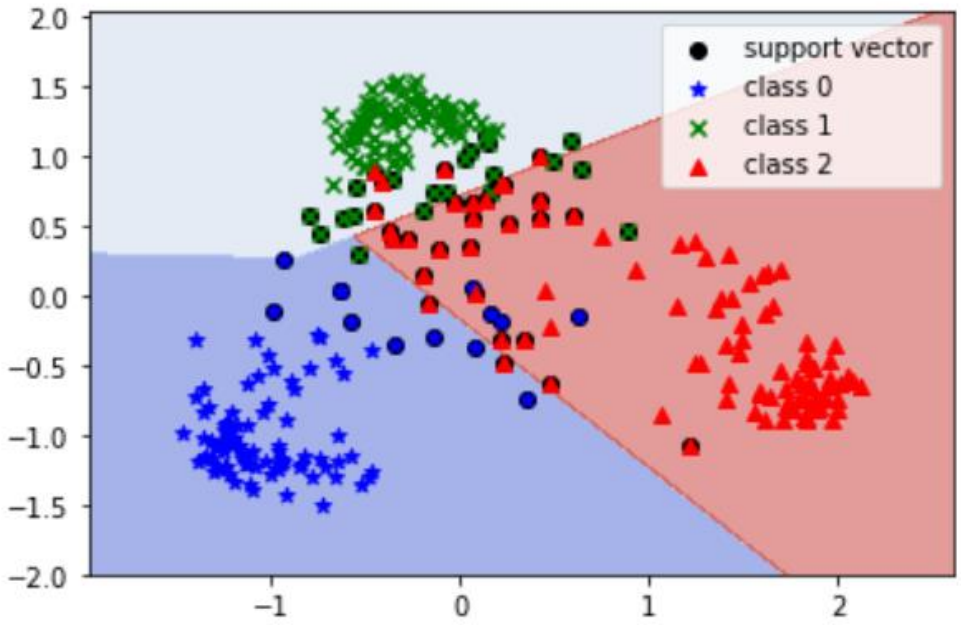
one-versus-one:

訓練時在任一兩類樣本之間設計 SVM，因此當有 K 個類別樣本時就需

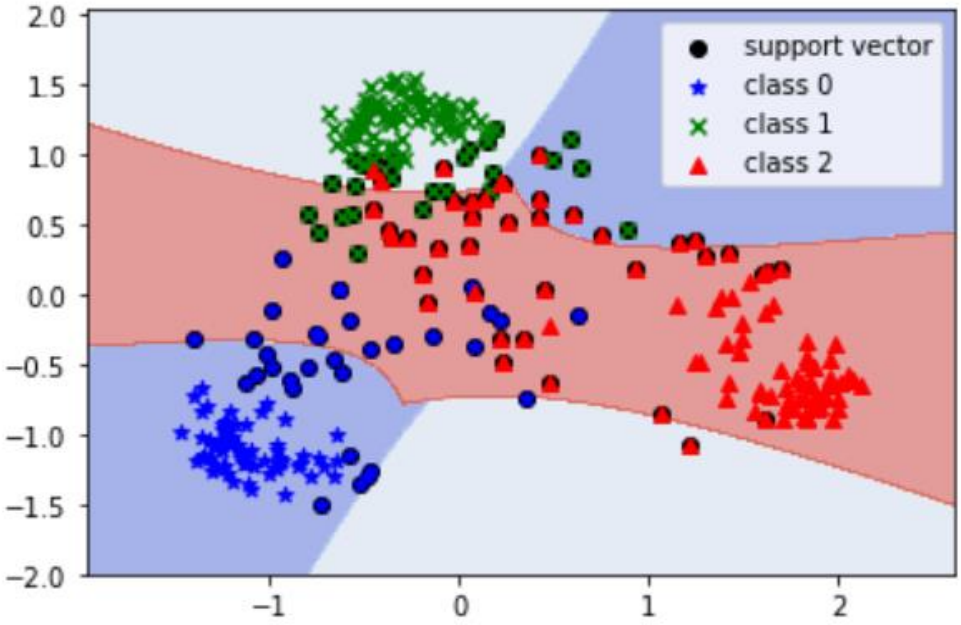
要(K-1)K/2 個 SVM 模型。分類時在各模型間計算獲得最多分數的即為該類。在類別較多時模型數跟計算量可能會較大。

在此題只有 3 類且資料量不算太大的情況下兩種方法表現是差不多的，我自己選用 ovo 來完成此題。

(2) Linear kernel



(3) Polynomial kernel (degree = 2)



(4)

比較兩結果可發現使用 polynomial kernel 在分類上有更精細的分法，而 linear 則較為單一。根據之前學到的，可能是 polynomial kernel 裡的非線性運算能讓資料在分類上，有著更好的表現。

3. Gaussian Mixture Model

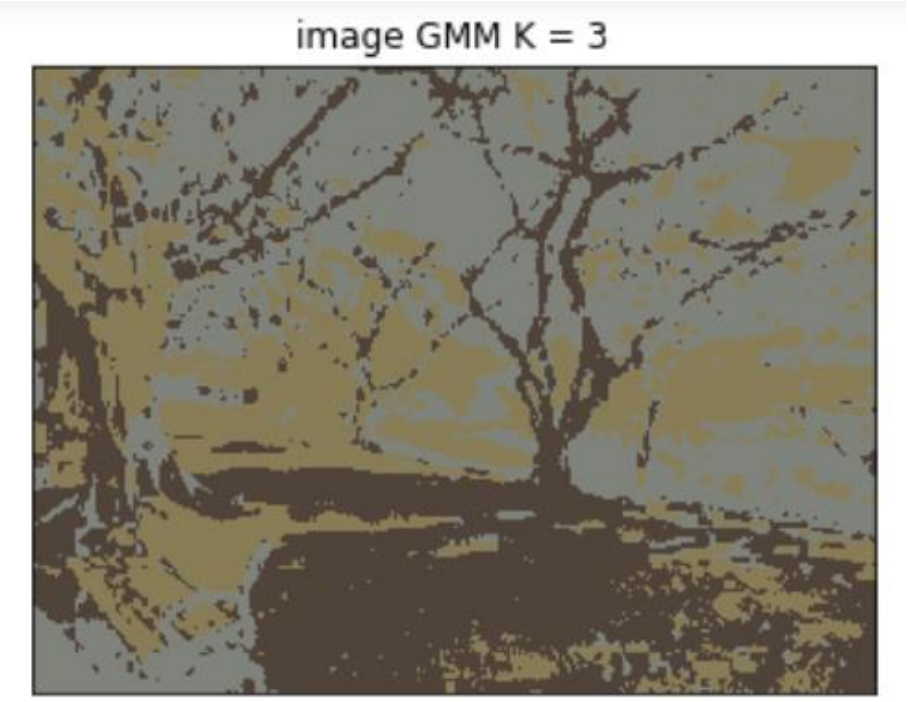
K = 3:

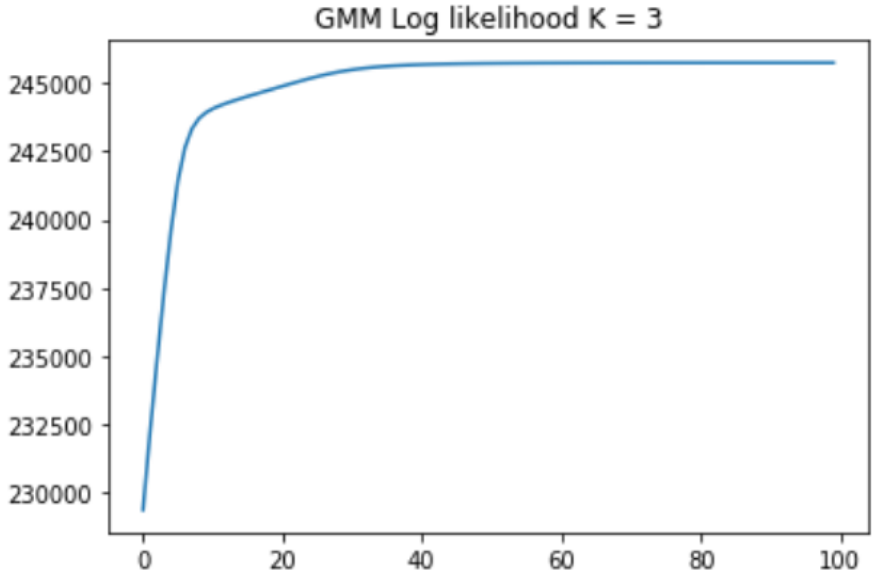
K-means K=3

	R	G	B
0	73	66	52
1	194	195	182
2	133	126	104

GMM K=3

	R	G	B
0	80	67	58
1	128	131	124
2	137	125	87





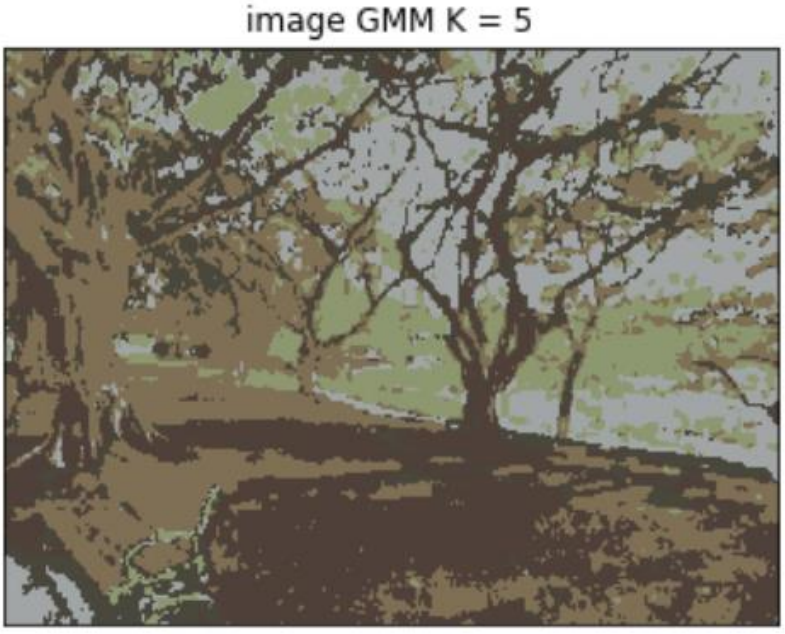
K = 5:

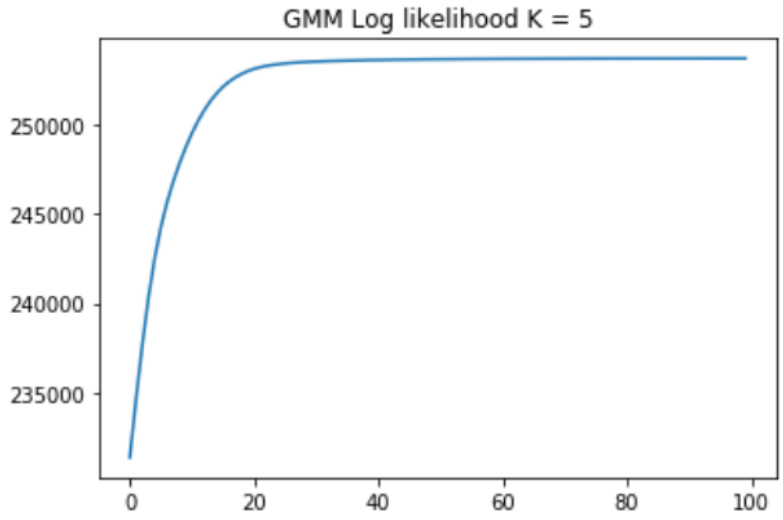
K-means K=5

	R	G	B
0	213	217	210
1	169	167	147
2	59	52	39
3	93	86	69
4	133	125	103

GMM K=5

	R	G	B
0	160	165	165
1	141	151	112
2	79	65	58
3	74	71	60
4	126	111	85





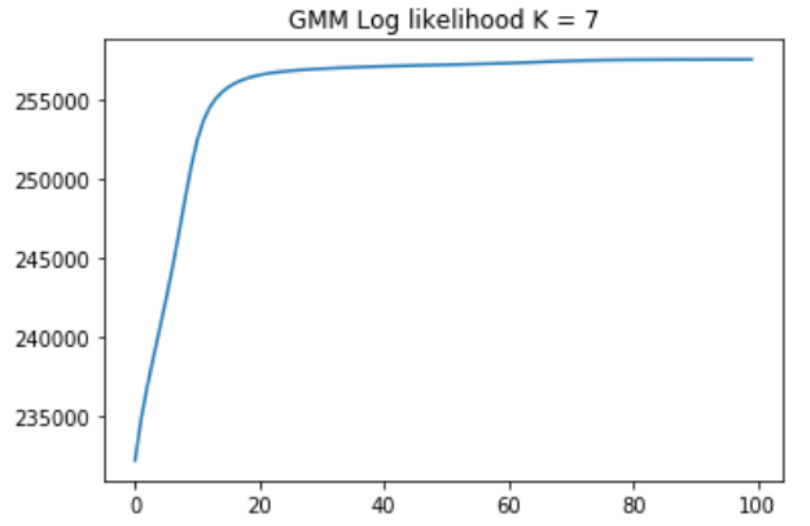
K = 7:

K-means K=7

	R	G	B
0	49	42	30
1	186	189	175
2	76	68	56
3	103	96	77
4	160	156	132
5	133	125	103
6	225	229	225

GMM K=7

	R	G	B
0	70	62	38
1	153	165	164
2	76	63	56
3	106	96	81
4	126	130	73
5	155	141	114
6	161	166	168



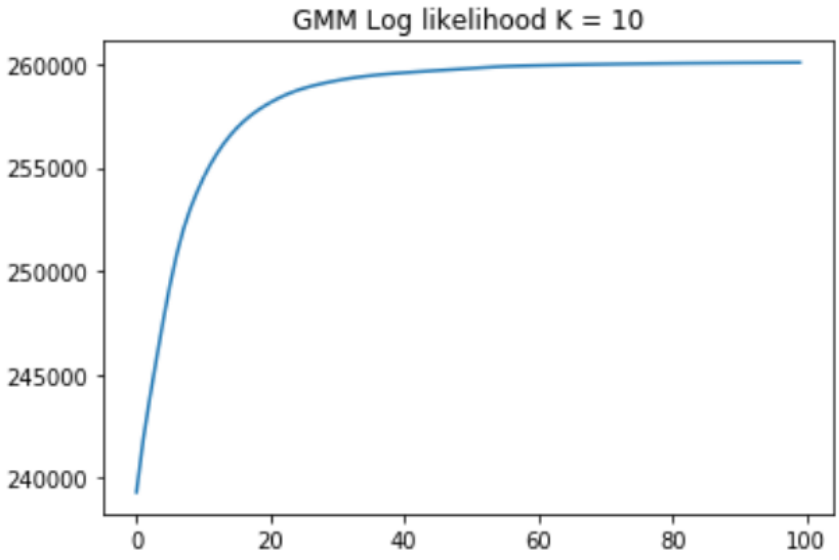
K = 10:

K-means K=10

	R	G	B
0	90	84	69
1	231	233	229
2	44	38	25
3	199	184	130
4	188	195	192
5	155	139	103
6	151	159	157
7	118	124	122
8	70	62	49
9	119	108	82

GMM K=10

	R	G	B
0	87	84	72
1	227	228	228
2	62	55	35
3	126	130	71
4	173	179	170
5	161	147	116
6	149	153	151
7	115	121	124
8	78	65	58
9	126	107	83



再補上 K-means 的圖:

image K-means K = 3

image K-means K = 5

image K-means K = 7

image K-means K = 10

(5)

由觀察可發現，K 值是影響輸出圖像非常關鍵的因素。當 K 值越來越大時，成像越清晰，Log likelihood 也較大。