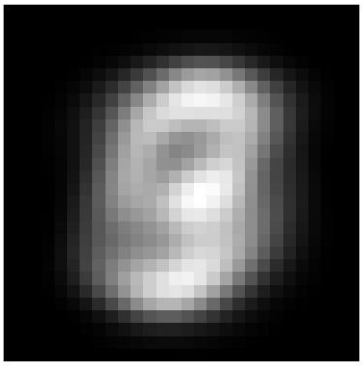
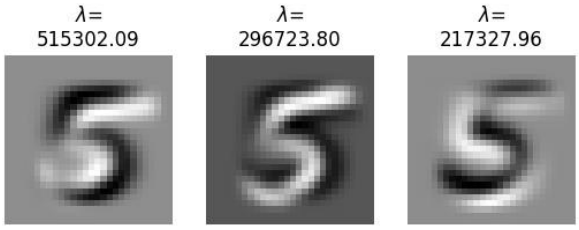
Program Homework 1 Report

NTU CSIE R10922100 許顥騰

Q1.



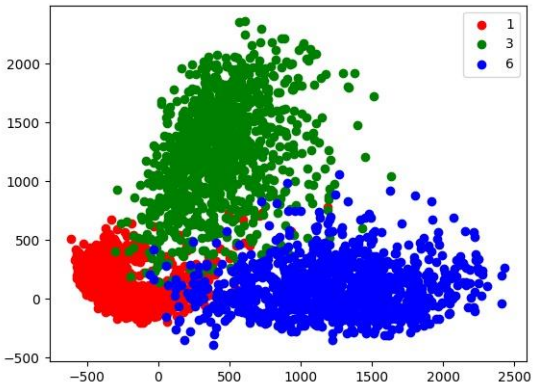
Q2.



Q3.

第一張的 5 可以看出在轉彎的地方比較銳利。
使用前三個最大的 Eigenvalues 對應的 Eigenvectors 作影像重建，他的轉彎處是較圓滑的，但隨著使用更多 Eigenvectors 來進行重建，轉彎處逐漸變得跟原始影像一樣銳利。

Q4.

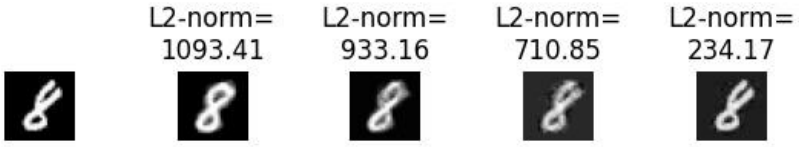


可以看出降維後的三個 cluster 重疊的部分不多，表示 1、3、6 沒有太多重複的特徵，如 1 就是一條垂直線，3 有兩個弧，6 有一個圈。重疊的部分我認為是有些 3 或 6 寫的很像 1。

Q5.

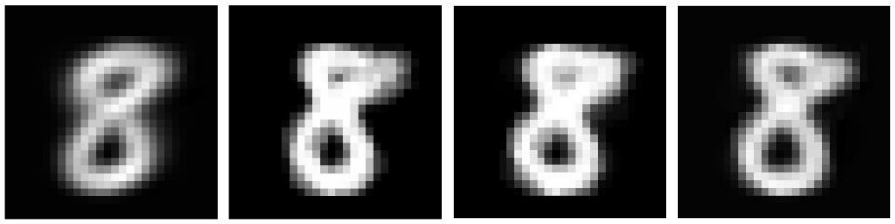
找出的五個 bases 前三個 bases 與第 10001 張的 3 非常接近，到了第四、第五個 bases 就長得比較歪了。

Q6.



第 10002 張的 8 頂部有個缺口。當 sparsity=5 時重建出來的 8 頂部是沒有缺口的，但 sparsity 到 40 時開始有出現一點缺口，最後 sparsity=100 時已經跟原影像沒什麼差異了。

Q7.



由左至右分別是第一到第四小題的結果。

PCA 跟 OMP 我是用自己手刻的。可以看出用 OMP 及 LASSO 還原出來的結果比較接近原始影像，PCA 用的 eigenvalues 不夠多沒辦法還原的比較接近。

第三、第四張使用 sklearn.linear_model 的 Lasso 來還原。Sklearn 中的參數 alpha 對應到 L1 regularization term 的 λ，預設為 1。

第三張為 alpha=1 的結果，外型與原始影像接近，上面的孔比較不明顯。
第四張為 alpha=0.1 的結果，隨著 alpha 的降低，整體的亮度稍微降低，而且上面的孔變得比原始影像還大。

除了 sklearn.linear_model 中的 Lasso，我還有試過 sklearn.decomposition 中的 sparse_encode 做 Lasso + coordinate descent，雖然 source code 中也是 call linear_model 中的 Lasso，但 sparse_encode 中的參數 alpha 還會再除以 feature 的數量，造成我一開始 alpha 設太小一直無法 converge。

Bouns:

+10

```python
def Lasso(X: np.ndarray,
          y: np.ndarray,
          alpha: float = 1.0,
          max_iter: int = 1000,
          tol: float = 1e-4) -> np.ndarray:
    """
        Implement Lasso from ppt: Sparse representation L1-norm solutions p.11

        y \approx X * coefficients

        X: n by N matrix
        y: n by 1 vector
        alpha: L1-norm penalty
        max_iter: max iterations
        tol: tolerance

        Return: coefficients(N by 1)
    """
    def S(a: float, x: float) -> float:
        return np.sign(x) * max(abs(x) - a, 0)

    N = X.shape[1]
    last_c = 0
    cur_c = np.zeros((X.shape[1], 1))
    converge = False
    iters = 1

    while not converge and iters <= max_iter:
        for i in range(cur_c.shape[0]):
            r = y - X.dot(cur_c) - cur_c[i] * X[:, i].reshape(-1, 1)
            cur_c[i] = S(alpha, X[:, i].T.dot(r) / N)
            iters += 1

        if abs(last_c - cur_c).sum() < tol:
            converge = True

        last_c = cur_c.copy()

    return cur_c
```

Bouns 的 Lasso 我是照著投影片 Sparse representation L1-norm solutions 的第 11 頁實作，中間的 while loop 即為 coordinate descent。

跟投影片中比較不一樣的是判斷收斂的方式，我是每做完一次完整的 coefficient 再判斷有沒有收斂，判斷方式是用上一次產生的 coefficient 跟這次產生的 coefficient 兩者的差的和有沒有小於一個閥值。