Q3.

(1) We compare the training time of incremental PCA and batch PCA and PCA trained only by the first subset:
When we set M_pca = 70:
Total training time of incremental PCA: 29.98634 seconds
Total training time of batch PCA: 8.70358 seconds
Total training time of PCA trained only by the first subset: 8.81949 seconds

Discuss: We can observe that PCA trained only by the first subset cost least time with only about 9 seconds, while the the incremental PCA cost the most time (about 30 seconds)

Explain:
PCA trained only by the first subset cost least time because the size of its data is just ¼ of the incremental PCA and batch PCA.

To explain why training time of incremental PCA is higher than batch PCA, we notice this:
The training time of incremental PCA include:
Time to compute eigenvalues,eigenvectors of covariance matrix of the first subset : 7.36042 seconds
Time to compute eigenvalues,eigenvectors of covariance matrix of the second subset : 7.07164 seconds
Time to compute eigenvalues,eigenvectors of covariance matrix of the third subset : 6.92605 seconds
Time to compute eigenvalues,eigenvectors of covariance matrix of the fourth subset : 7.72964 seconds
Time to merge all 4 eigenspace models above: 0.89859 seconds

The training time of batch PCA is equal to the time to compute eigenvalues, eigenvectors of covariance matrix of the training dataset (the dataset before being split into 4 subsets), and this process costs 8.70358 seconds.

From the above statistics, we can observe that in the total training time, the large amount of time comes from the process computing eigenvalues, eigenvectors of covariance matrix. But we also notice that the time to compute eigenvalues, eigenvectors of covariance matrix of the training dataset, the first subset, the second subset, the third subset, the fourth subset are nearly the same:
It costs 8.70358 seconds for the training dataset, costs 7.36042 seconds for the first subset, costs 7.07164 seconds for the second subset, costs 6.92605 seconds for the third subset, and costs 7.72964 seconds for the last subset.
These amount of times are nearly the same because the shape of covariance matrix of the training dataset, and covariance matrix of the 4 subsets all have the same shape (2576,2576). And we already know that the time to compute eigenvalues and eigenvectors of a matrix depends mainly on the shape of the matrix.
With nearly the same amount of time to compute eigenvalues and eigenvectors of covariance matrix (about 7 or 8 seconds), however in case of time of training process for batch PCA, we only add this amount once, while in case of incremental PCA, we have to add this amount 4 times for total 4 subsets. It is the reason why total training time of incremental PCA is larger than batch PCA.

However, when we consider the time to merge 4 eigenspace models in incremental PCA, it costs only 0.89859 seconds, that's small in comparison to the whole training time. This is very meaningful because let's consider this situation:
When we have 4 datasets X1,X2,X3,X4 and we also have 4 eigenspace models for these datasets. Now, we want to merge these 4 datasets X1,X2,X3,X4 into a bigger dataset and name it X, and we

want to compute the eigenspace model of X. We have two ways to do this: compute eigenspace model for X from scratch or merge 4 eigenspace models of X1,X2,X3,X4 to form the new eigenspace model for X. Obviously, the time cost for the second direction would be much smaller than the first direction. For example, for the dataset of Q3, it costs 8.70358 secondsto compute eigenspace models of the training set by batch PCA, but it costs only 0.89859 seconds if we create that eigenspace model by merging eigenspace models of 4 subset of the training dataset.

(2) We compare the average reconstruction error of incremental PCA and batch PCA and PCA trained only by the first subset:
When M_pca = 70:
Average reconstruction error by incremental PCA: 682815.785
Average reconstruction error by batch PCA: 675444.041
Average reconstruction error by PCA trained on first subset: 1242675.661

Discuss: The reconstruction error by PCA trained only on the first subset is largest, and the reconstruction error by batch PCA is smallest, but nearly equal to the reconstruction error by incremental PCA.

Explain: In case of PCA trained only on the first subset, because it uses only ¼ amount of data from the training dataset, while the remaining ¾ data from training dataset are not used, so a lot of information is lost, and as a result, the reconstruction error in this model is high.
In case of incremental PCA and batch PCA, the amount of data used for training is the same (the whole training dataset), the only difference is that incremental PCA divides the whole data into 4 subsets, and merge their eigenspace models continuously to form the final eigenspace model, but this does not lead to any loss in information for training, and therefore, its reconstruction error is nearly the same as batch PCA.

(3) We compare recognition accuracy of incremental PCA and batch PCA and PCA trained only by the first subset:
Recognition accuracy by incremental PCA: 66.34615 %
Recognition accuracy by batch PCA: 66.34615%
Recognition accuracy by PCA trained only on the first subset: 63.46153 %

Discuss: The recognition accuracy of PCA trained only on the first subset is lowest, the recognition accuracy by incremental PCA and batch PCA are the same.

Explain:
- In case of PCA trained only on the first subset, because it uses only ¼ amount of data from the training dataset, while the remaining ¾ data from training dataset are not used, so a lot of information is lost, and as a result, the recognition accuracy is lower (this maybe the problem of underfitting).
- In case of incremental PCA and batch PCA, they use the same dataset (whole training dataset), the only difference is that incremental PCA divides the whole data into 4 subsets, and merge their eigenspace models continuously to form the final eigenspace model, but in theory, this does not lead to any loss in information for training, and therefore, its recognition accuracy is the same as batch PCA.

(4) Further discuss:
I believe that my implementation for incremental PCA is quite accurate because as we can observe from above statistics, the incremental PCA gains the same recognition accuracy as batch PCA, and

in terms of reconstruction error, the incremental PCA has nearly the same reconstruction error as batch PCA.

Discuss about parameters:
M_pca (the number of PCA bases) is the only important parameter that I have to set in all methods incremental PCA, batch PCA and PCA trained on only the first subset. Here is some statistics when I change the value of M_pca:

When M_pca = 70:

	Incremental PCA	Batch PCA	PCA trained only on the first subset
Training time (seconds)	29.98634	8.70358	8.81949
Recognition accuracy (%)	66.34615	66.34615	63.46153
Reconstruction error	682815.785	675444.041	1242675.661

When M_pca = 200:

	Incremental PCA	Batch PCA	PCA trained only on the first subset
Training time (seconds)	32.56629	7.99226	8.98869
Recognition accuracy (%)	67.30769	67.30769	66.34615
Reconstruction error	428128.348	427097.128	1152000.042

Observation: When we increase M_pca, in all 3 methods, the recognition accuracy increase, while the reconstruction errors decrease and the training times increase. This is because when we increase M_pca, then less information is lost and as a result, the recognition accuracy increase and the reconstruction errors decrease. But at the same time, when we increase M_pca, then there is more computations in the implementation and lead to a slight increase in training time in overall. However, even when we change M_pca, then the trend does not change: recognition accuracy of incremental PCA and batch PCA are the same and they are higher than recognition accuracy of PCA trained only on the first subset, the reconstruction error of incremental PCA and batch PCA are nearly the same and they are lower than reconstruction error of PCA trained only on the first subset.

***Notice** : For the implementation of incremental PCA in this question, when I merge 2 eigenspace models , instead of keeping all eigenvectors from these 2 models, I will eliminate some non-important eigenvectors (eigenvectors that have zero-eigenvalues after the eigen-decomposition), and only keep important eigenvectors to form the new eigenspace model. Furthermore, for my implementation of incremental PCA, the number of eigenvectors retained in any model, including a merged model, was set to be M_pca as a maximum (M_pca is the parameter to decide the number of PCA bases).*