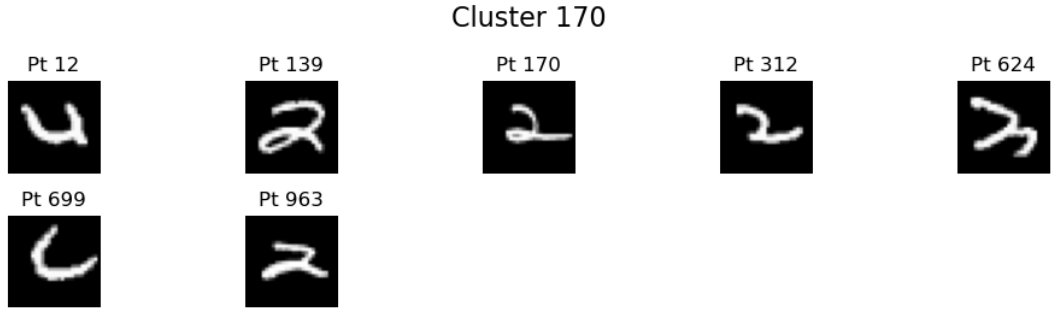
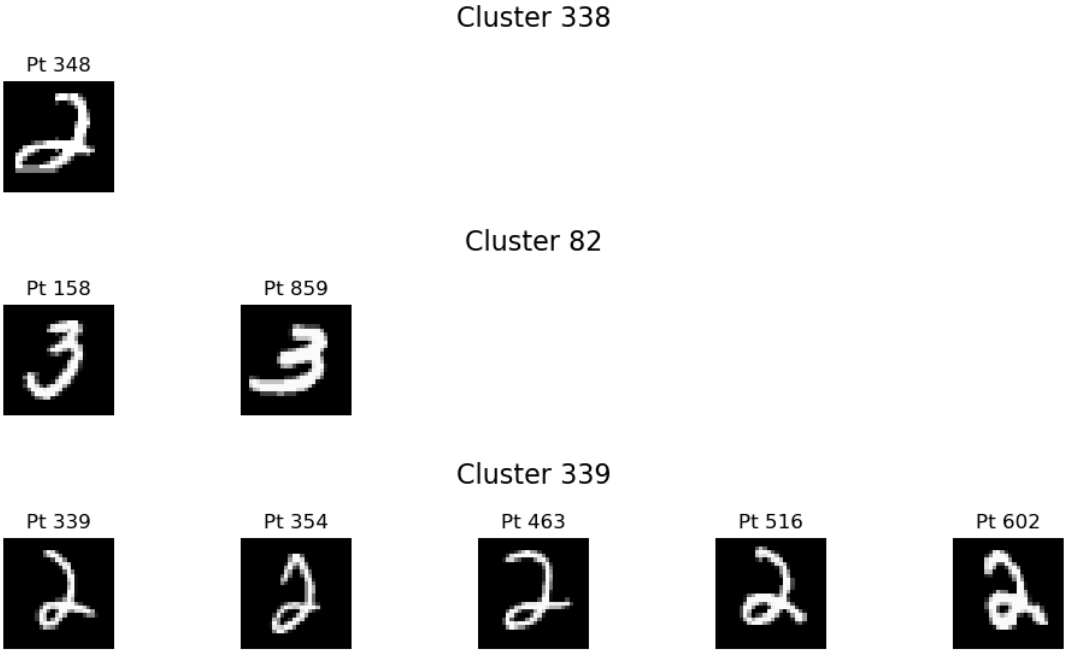
Task2 Affinity Propagation

In this project I made a parallel implementation of the affinity propagation solution. I used the multiprocessing library in python to implement a parallel solution of the iterative calculation of R and A which is the part that takes the most time. The parallel implementation makes that the matrix are separated into blocks so each processor can calculate its own block. However it turned out that the separation and then re-assembling of blocks is also time consuming and so when I was first working with only 500 samples, not dividing the matrix was faster than doing parallel calculation. But if we get to 2000 samples then separating into blocks and doing parallel calculation makes the program 3-4 times faster. But because I wanted to do a lot of iteration (about 400) to have accurate results, the program would still take a lot of time so I didn't go above 1000 samples.

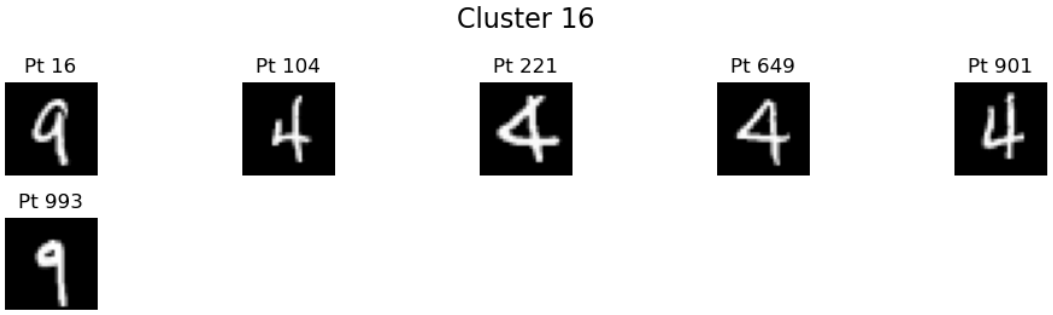
In this task the best result I was able to obtain 161 clusters out of the 1000 samples which can be a bit far from the ideal 10 clusters.
But it's understandable because some numbers are written in a very special way like in this cluster:



There are some clusters where there are only few numbers :



And that can be not really accurate.



There are also clusters with more value (but the biggest one does not exceed 30 numbers) with some that are really accurate and others that make mistakes.