For the second task of this assignment, initially we considered downloading images, however, after further research we discovered that downloading is primarily an I/O bound task, since there is a wait time for the images to download. Therefore, to follow the assignment requirements we decided image processing was a more appropriate task as it is mainly a CPU bound task.

We utilised the Pillow Python library to do some image processing (adding a Gaussian blur) on the pre-downloaded images. The `normal\_image\_process` and `multiprocessing\_image\_process` functions are identical except for where they save the processed images. It would have been preferable to merge this into one function which takes the folder route as a parameter, but the pool process function does not permit this as the function provided to it must only take one argument.

As can been seen in the accompanying notebook, it took 17 seconds to process the 15 images using only 1 core. When 4 cores were used the time took to process the images was 7 seconds. This increase in core utilisation lead to a 58.8% decrease in processing time. This is a significant drop in time and highlights how multiprocessing can greatly reduce the time necessary to perform CPU intensive tasks.