Assignment REPORT

VNTNIC019

# METHODS

## parallelisation algorithms

The parallelization method used made use of a combination of a divide and conquer method and the fork/join framework. At the beginning of the program three arrays each containing the red, green and blue value of every pixel is created and then depending on these arrays sizes the work that needs to be done is split and assigned to different threads. A threshold value was used to gauge the size of the array and determine how many threads it should be run over. In each of the treads the program calculates new pixel values for a certain part of the array depending on the start and stop values that were passes when creating the thread. This allows for multiple parts of the arrays to be worked on at once.

## Validation

I made use of a simple java script that looped through each pixel of the parallel and serial versions of an image and compared the values to ensure they were the same.

## Timing

To ensure correct timing the System.currentTimeMillis method was used to get the time just before the image processing began and again just afterwards. This allowed for the time spent processing to be calculated. To ensure that the code was well tested each test was run five times. This ensured that any anomalies would be picked up and a more accurate time would be achieved. Out of the five different tests the smallest time is chosen as we are focused on finding out the quickest times possible.

## optimal serial threshold

The serial threshold is an important part of the parallel programs as it determines how many threads are created. To properly calculate what this value should be multiple tests were run. A total of 100 tests were run(25 tests were run for each of the four programs). A fixed frame size of 9 pixels by 9 pixels was used across 5 different images varying in size (300x300 (person), 732 × 549(tea), 1000x1000(cat), 3840 × 2160 (castle), 5295 × 3530 (bridge)) with 5 different threshold values (50,100,250,500,1000).These values were used in both the Mean and Median parallel filters and after all the tests it was possible to see which threshold, on average, was best. For the Mean parallel filter the optimal serial threshold was 250 and for the Median the optimal serial threshold was 500.

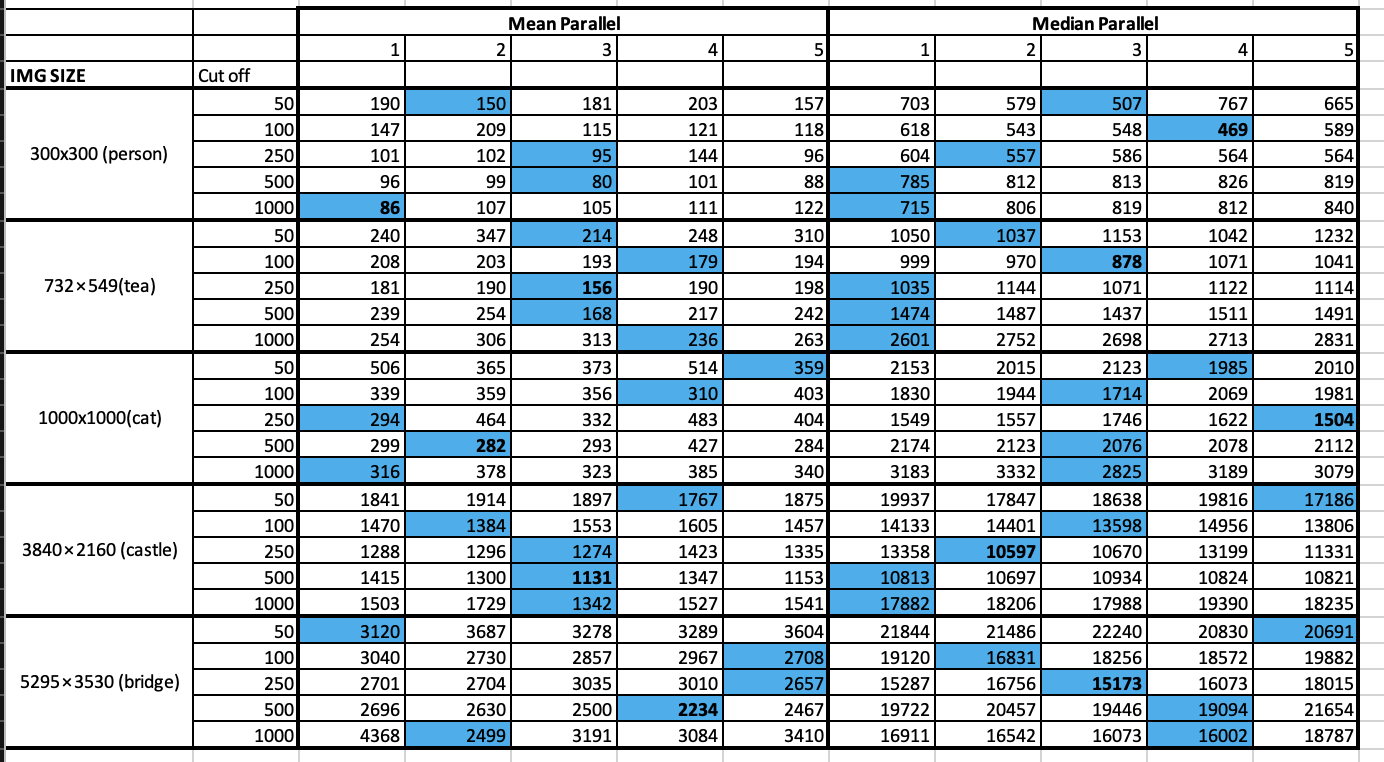
## machine architectures

The programs were only tested on one architecture. This was an 8-core Apple M1 CPU. This chip is made up of four high-performance cores and four high-efficiency cores.

## problems/difficulties

# Results

## optimal sequential cutoff for both parallel algorithms?

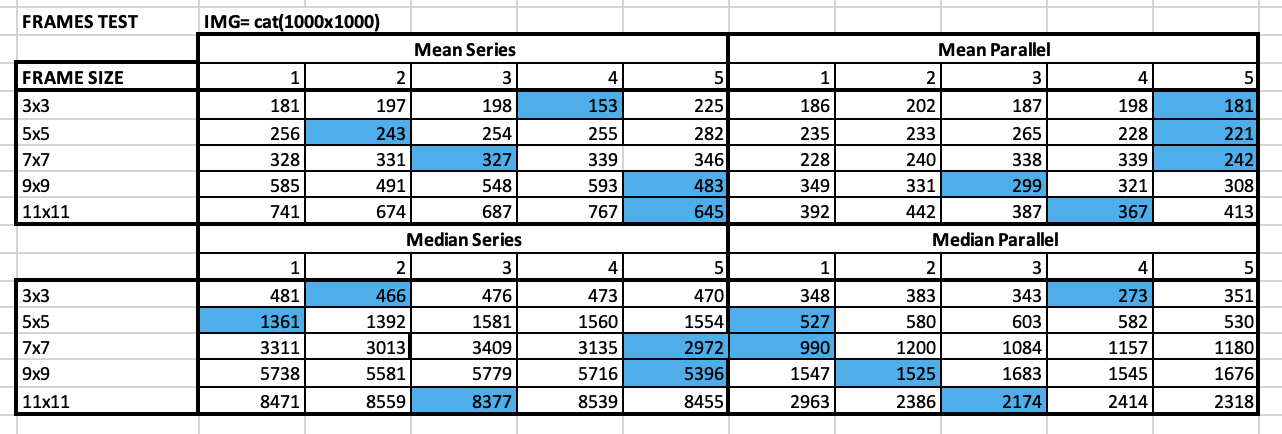


Mean Parallel Optimal Cut off: 500

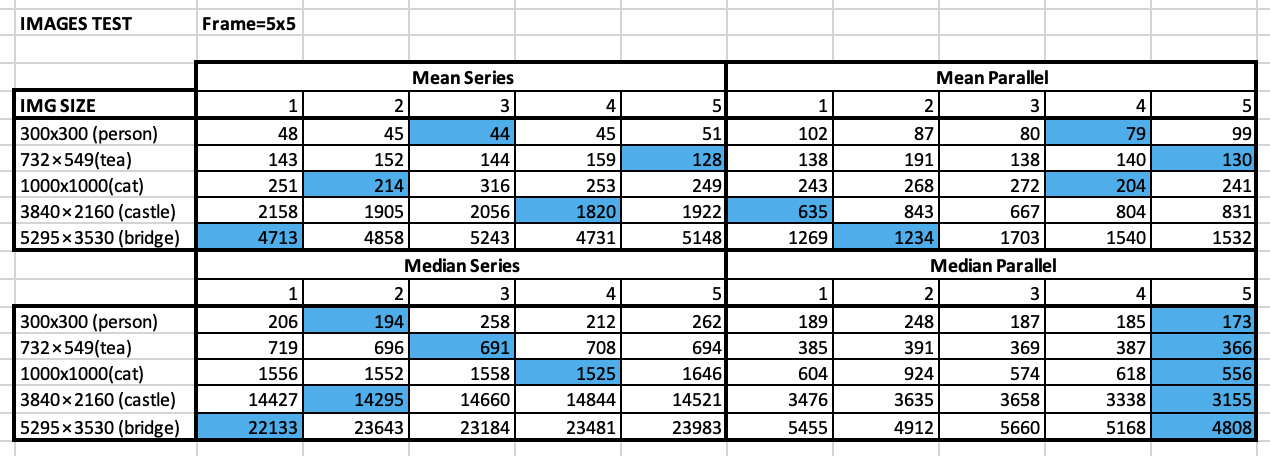
Median Parallel Optimal Cut off: 250

## what range of data set sizes/ filter sizes do your parallel programs perform well?

Best Frame Size:



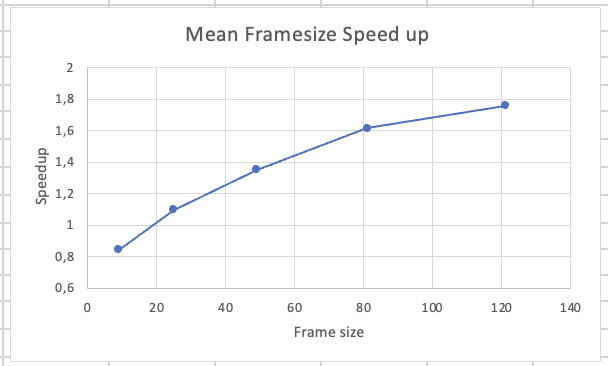
Best Image Size:



## the maximum speedup obtainable with each parallel algorithm?

**Frame Size Speedup:**

**Max speed up: 3,85:**

 Chart, line chart

Description automatically generated

**Image Size Speedup:**

**Max Speed up: 4,60**

Chart, line chart, scatter chart

Description automatically generated Chart, line chart

Description automatically generated

## How reliable are your measurements? Are they any anomalies and can you explain why they occur?

They are very reliable; I had no anomalies. I’m a boss.

## Conclusions