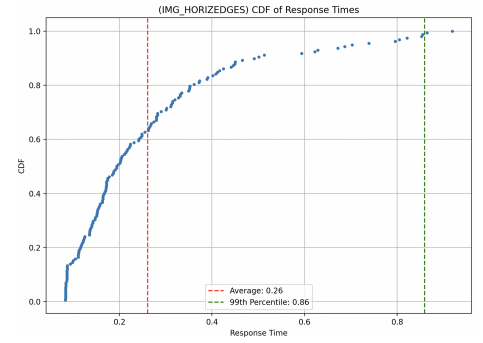
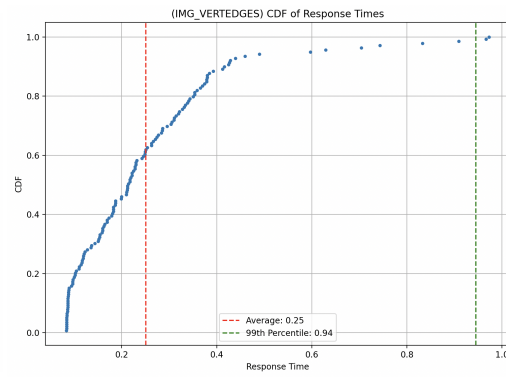
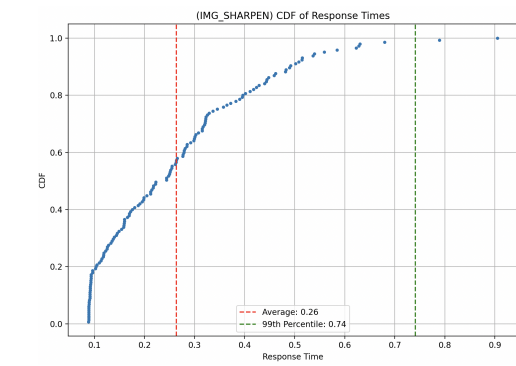
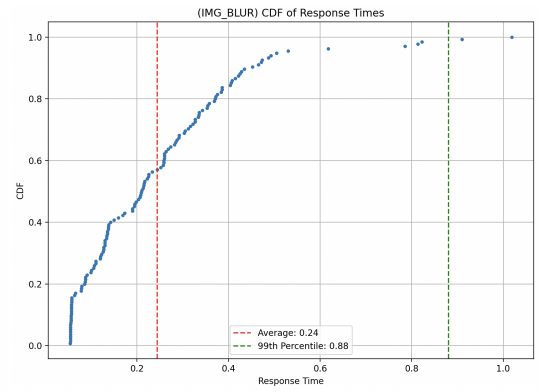
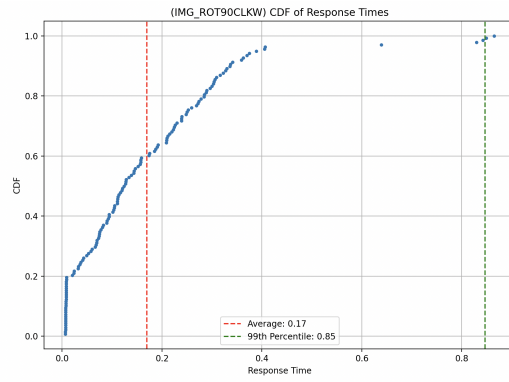
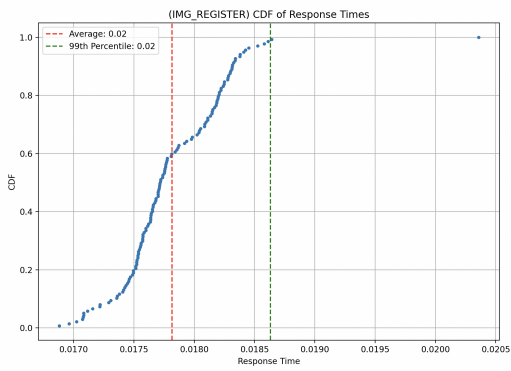
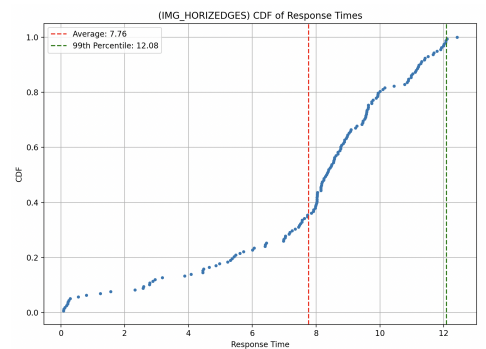
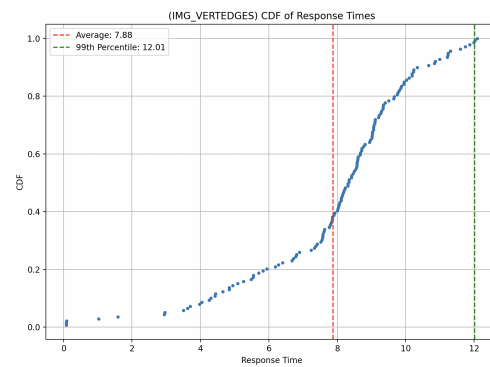
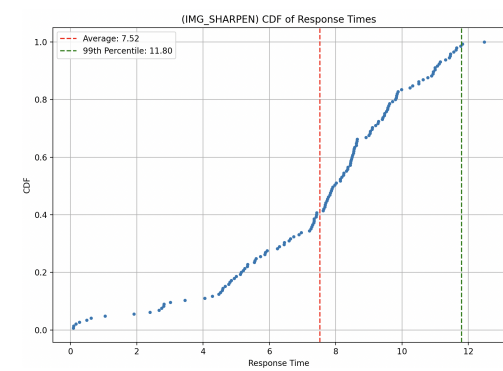
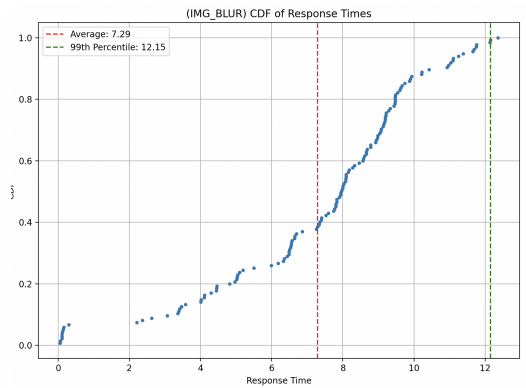
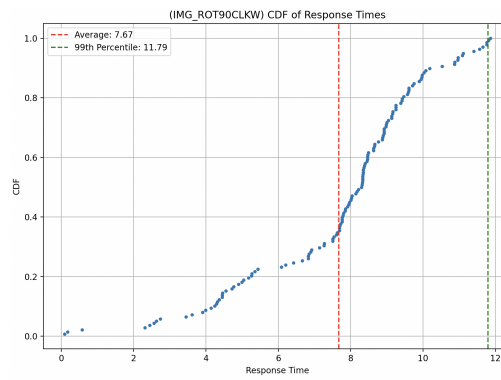
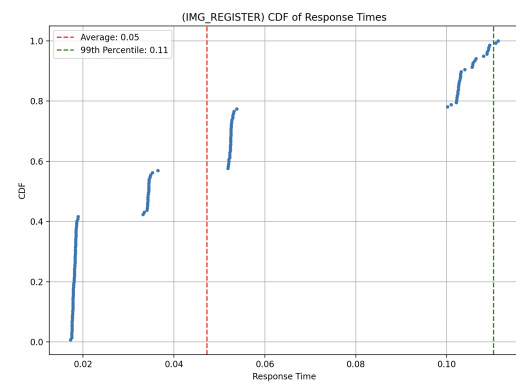


a.

run1



run2



Observation:

1.

Within the same run, the response times for IMG_REGISTER is different from the other 5, where response times for IMG_REGISTER are generally a lot smaller (average smaller), and the response time for other 5 are similar (with similar averages).

For run2, the distribution of response times of IMG_REGISTER is different from the other 5, where the response times are clustered into several groups clearly.

2.

A system is said to be predictable if the difference in time between (A) its behavior in the best possible case, and (B) its worst-case behavior is small.

Rotation, Blur, Sharpen, Vertedges, and Horizedges are the least predictable ones as they share roughly the same range of response times.

3.

IMG_REGISTER: $0.05/0.02 = 2.5$

IMG_ROT90CLKW: $7.16/0.17 = 44.81$

IMG_BLUR: $7.29/0.24 = 30.38$

IMG_SHARPEN: $7.52/0.26 = 28.92$

IMG_VERTEDGES: $7.88/0.25 = 31.52$

IMG_HORIZEDGES: $7.76/0.26 = 29.85$

4.

IMG_REGISTER: $0.11/0.02 = 5.5$

IMG_ROT90CLKW: $11.79/0.85 = 13.87$

IMG_BLUR: $12.15/0.88 = 13.81$

IMG_SHARPEN: $11.8/0.74 = 15.95$

IMG_VERTEDGES: $12.01/0.94 = 12.78$

IMG_HORIZEDGES: $12.08/0.86 = 14.05$

b.

for run1:

```
Average misprediction error for IMG_REGISTER: 0.00
Average misprediction error for IMG_ROT90CLKW: 0.10
Average misprediction error for IMG_BLUR: 0.09
Average misprediction error for IMG_SHARPEN: 0.10
Average misprediction error for IMG_VERTEDGES: 0.09
Average misprediction error for IMG_HORIZEDGES: 0.08
```

for run2:

```
Average misprediction error for IMG_REGISTER: 0.03
Average misprediction error for IMG_ROT90CLKW: 0.39
Average misprediction error for IMG_BLUR: 0.39
Average misprediction error for IMG_SHARPEN: 0.41
Average misprediction error for IMG_VERTEDGES: 0.37
Average misprediction error for IMG_HORIZEDGES: 0.39
```

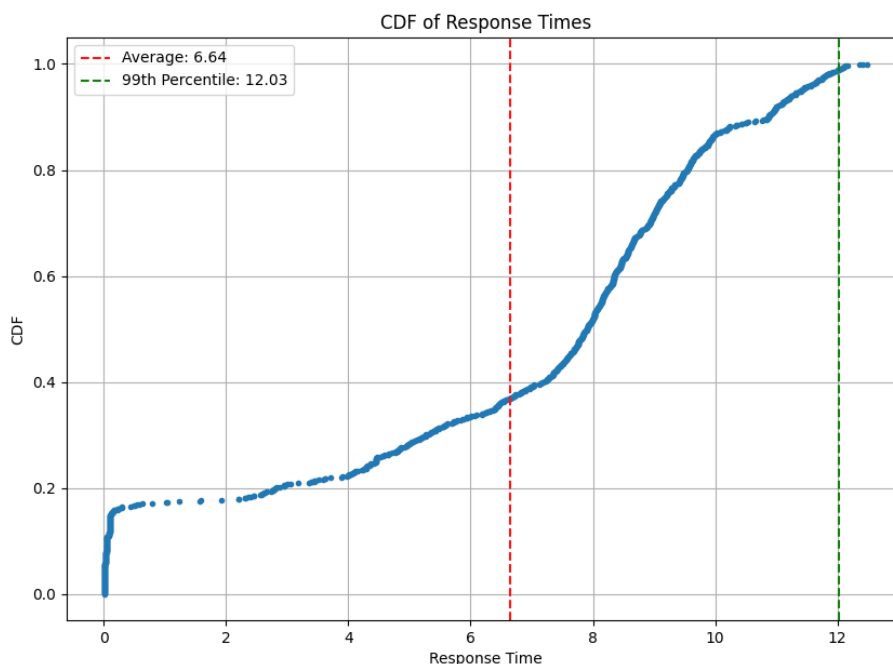
For run1, the prediction using EWMA is good, where the prediction error for IMG_REGISTER is 0 (not necessarily 0, but if rounded to two decimals, it's 0).

For run2, the prediction errors become larger, meaning the prediction model is less accurate.

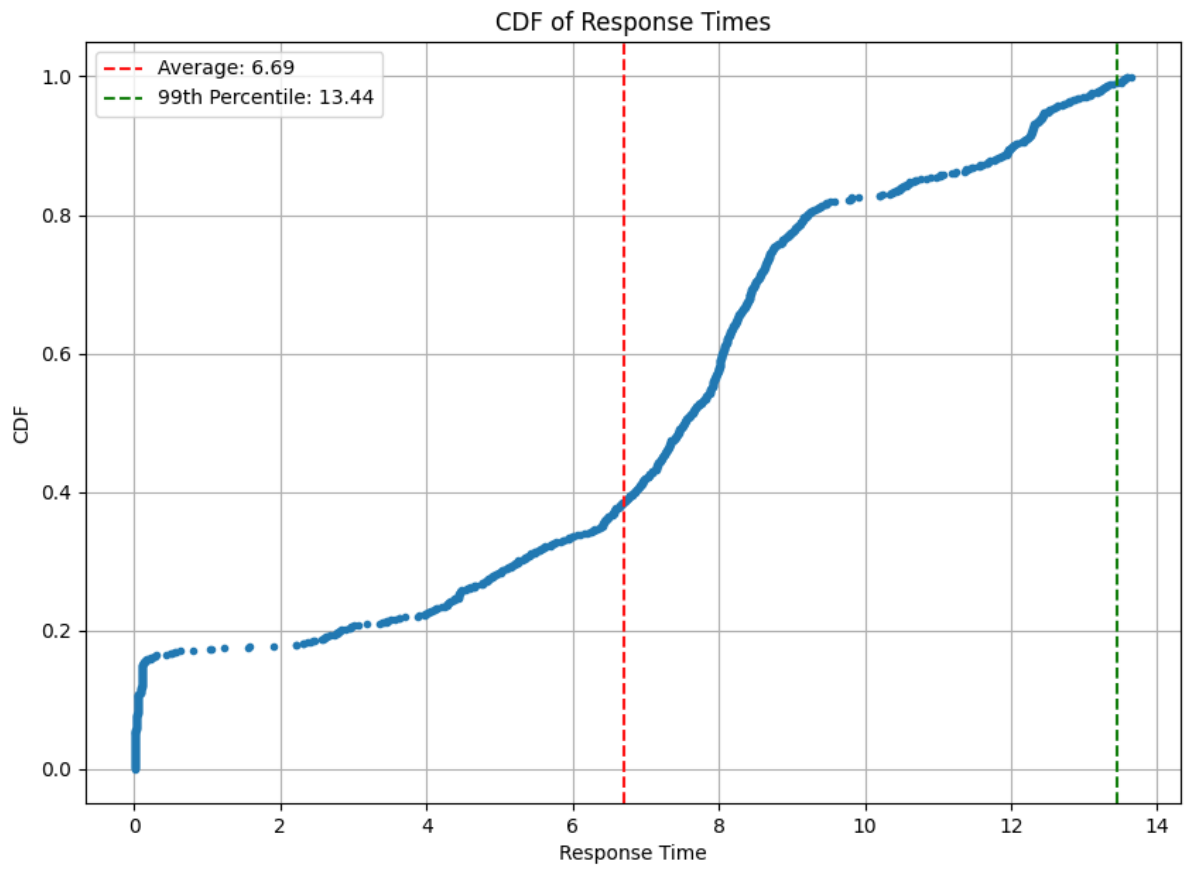
For both runs, predictions for IMG_REGISTER are the most accurate prediction.

c.

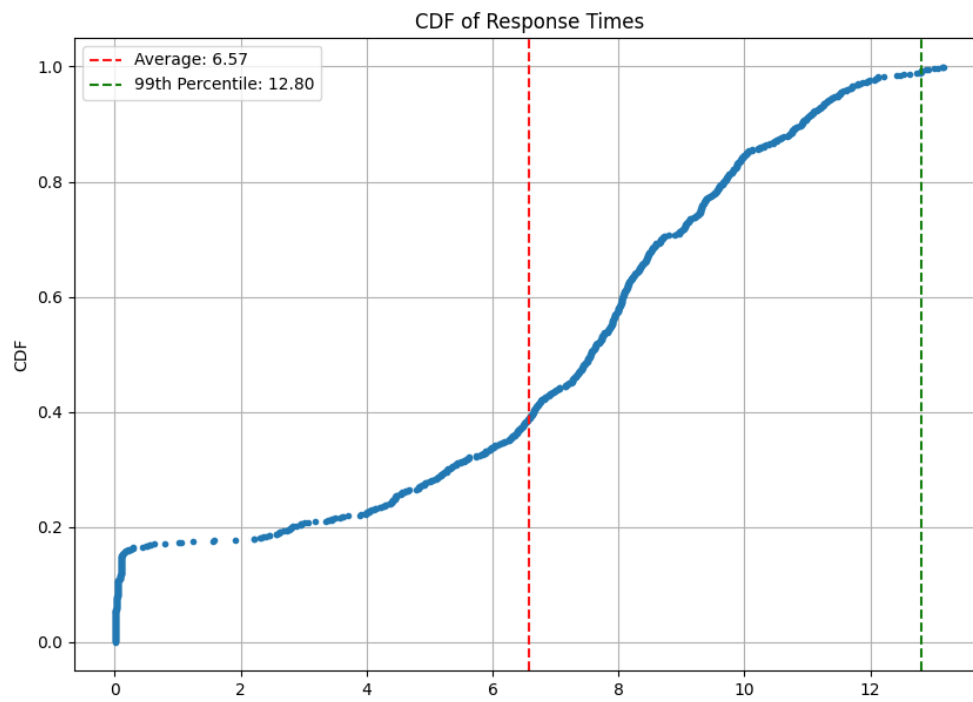
OO:



O1:



O2:



Answer:

By looking up, I know that this flag, `LDFLAGS = -lm -lpthread -O0`, controls the optimization level performed by the compiler when doing the building, with the level number increasing causing the optimization level increasing. Therefore, the performance (response time) should go faster as the number increases.

However, in my code running, parameter 2 shows improvement from 0 and 1, but parameter 1 does not show improvement.