YOLOv5 Performance Analysis & Optimization

Disha Pant

1. System Specs [1]

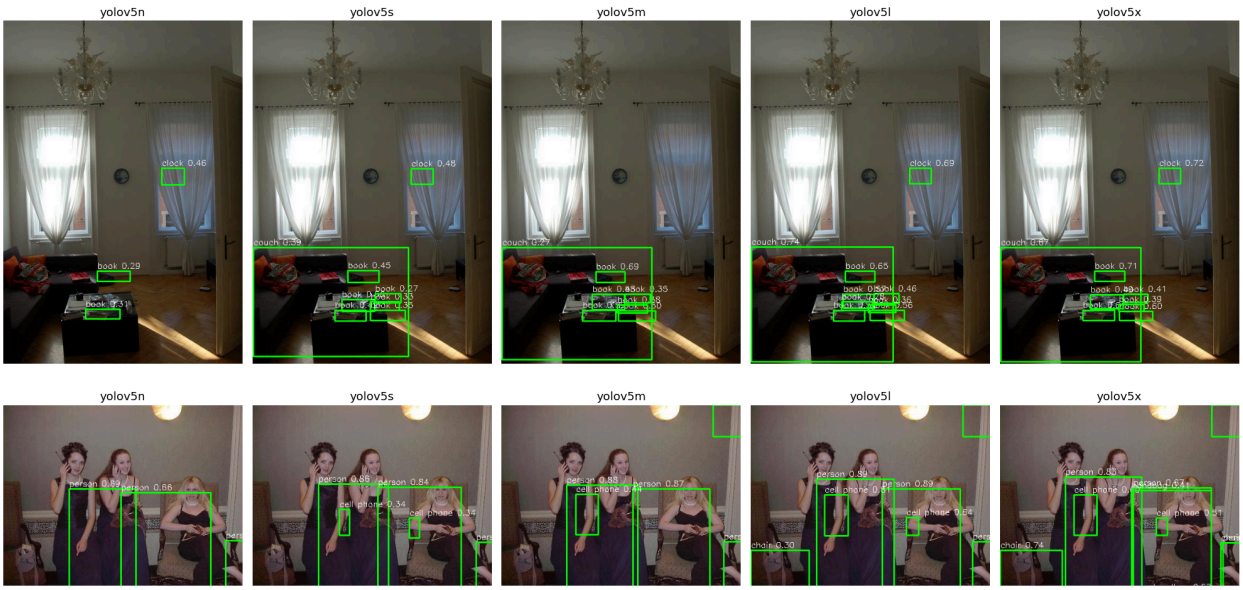
CPU Name: AMD Ryzen 7 6800HS with Radeon Graphics
Threads per core: 2
Core(s) per socket: 8
Socket(s): 1
Clock Rate: 3200 - 4700 MHz
RAM: 16 GB

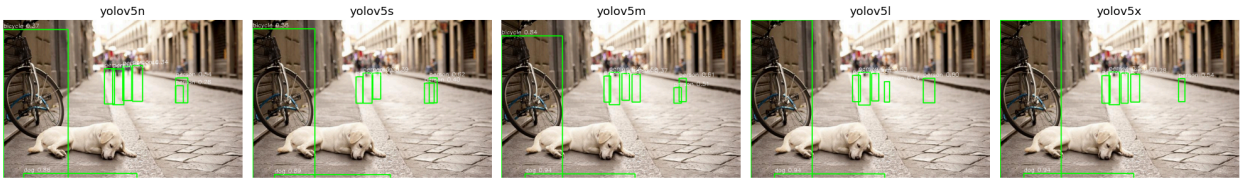
[1]https://www.notebookcheck.net/AMD-Ryzen-7-6800HS-Processor-Benchmarks-and-Specs.591454.0.html

Peak Bandwidth: Triad bandwidth selected from running STREAM benchmark, 28.1 GB/s
Theoretical GFLOPS: 3.2 Ghz * 16 flops per cycle * 8 cores = ~410 GFLOPS
Used these two values in the Jupyter notebook.

2. Dataset Used

I used the COCO 128 Mini dataset, available here, https://ultralytics.com/assets/coco128.zip.
Outputs from the models for some random images from the dataset:



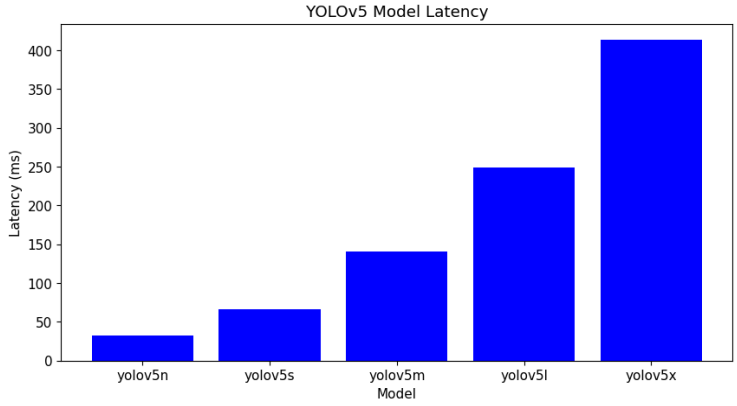


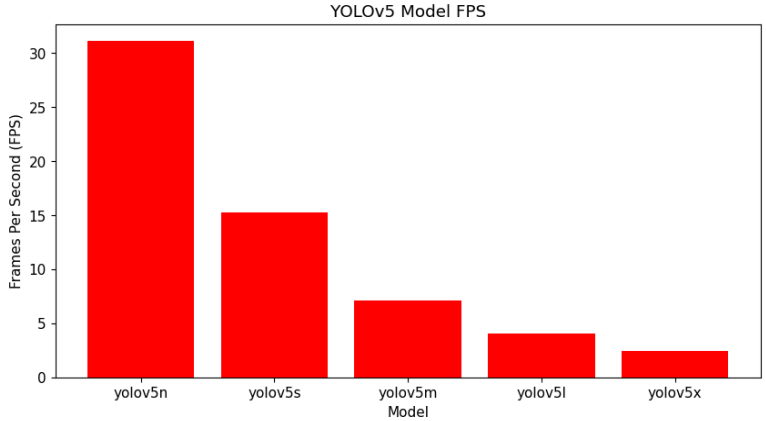
3. Profiling Tool Outputs (Tables/Screenshots/Graphs)

Throughput and Latency:

We saw latency increase and FPS decrease as model size increased.

Model	Latency (ms)	FPS
yolov5n	32.14 ms	31.11
yolov5s	65.71 ms	15.22
yolov5m	140.50 ms	7.12
yolov5l	249.42 ms	4.01
yolov5x	413.54 ms	2.42





Roofline Bound Analysis:

```
=========================================================================
Model      Params (M)  Size (MB)   GFLOPs   Utilization (%)   Bound Type
=========================================================================
yolov5n          1.87       7.12     2.23            16.96     Compute Bound
yolov5s          7.23      27.56     8.22            30.50     Compute Bound
yolov5m         21.17      80.77    24.44            42.42     Compute Bound
yolov5l         46.53     177.51    54.50            53.29     Compute Bound
yolov5x         86.71     330.75   102.73            60.59     Compute Bound
=========================================================================
```

I used the specs defined in section 1, the peak memory bandwidth and peak compute throughput to obtain the value for peak OI for my system:

Peak OI = Peak GFLOPS / Peak Bandwidth = 410 / 28.1, which is approximately 14.59. In my code I calculated OI for each model and checked if it was above or below this value. In all 5 cases, the calculated OI was higher, indicating that all the models are Compute Bound. We also see the CPU utilization going up as the model size increases.

Per Layer Analysis:

```
================================================================
Per-Layer Utilization Summary
================================================================

Model: yolov5n
Highest Utilization Layers:
    model.model.6                          : 1.89%
    model.model.6.m                        : 1.49%
    model.model.4                          : 1.39%
Lowest Utilization Layers:
    model.model.23.cv3.act                 : 0.00%
    model.model.23.m.0.cv1.act             : 0.00%
    model.model.23.m.0.cv2.act             : 0.00%

Model: yolov5s
Highest Utilization Layers:
    model.model.6                          : 3.70%
    model.model.6.m                        : 2.92%
    model.model.4                          : 2.72%
Lowest Utilization Layers:
    model.model.23.cv3.act                 : 0.00%
    model.model.23.m.0.cv1.act             : 0.00%
    model.model.23.m.0.cv2.act             : 0.00%
```

Across all models, model.model.6, model.model.6.m, and model.model.4 had the highest utilization, whereas model.model.23.cv.3.act, model.model.23.m.0.cv1.act and model.model.23.m.0.cv2.act had the lowest, though the actual values increased with increase in size of the model itself.

```
Model: yolov5m
Highest Utilization Layers:
    model.model.6                          : 6.96%
    model.model.6.m                        : 6.14%
    model.model.4                          : 4.91%
Lowest Utilization Layers:
    model.model.23.m.0.cv2.act             : 0.00%
    model.model.23.m.1.cv1.act             : 0.00%
    model.model.23.m.1.cv2.act             : 0.00%

Model: yolov5l
Highest Utilization Layers:
    model.model.6                          : 10.05%
    model.model.6.m                        : 9.23%
    model.model.4                          : 6.97%
Lowest Utilization Layers:
    model.model.23.m.1.cv2.act             : 0.00%
    model.model.23.m.2.cv1.act             : 0.00%
    model.model.23.m.2.cv2.act             : 0.00%

Model: yolov5x
Highest Utilization Layers:
    model.model.6                          : 12.37%
    model.model.6.m                        : 11.60%
    model.model.4                          : 8.50%
Lowest Utilization Layers:
    model.model.23.m.2.cv2.act             : 0.00%
    model.model.23.m.3.cv1.act             : 0.00%
    model.model.23.m.3.cv2.act             : 0.00%
================================================================
```

Profiling Tools

Time breakdown across pipeline stages is in the next section of this report. There is no comparison of CPU vs GPU time as all computations were run on the CPU, which is also evident from the below cProfile screenshots. A potential memory or threading issue is the negative value obtained for aten:max_pool2d’s self CPU memory.

cProfile Outputs

```
=== cProfile for yolov5n ===
         11568142 function calls (10913825 primitive calls) in 15.538 seconds

   Ordered by: internal time
   List reduced from 558 to 15 due to restriction <15>

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    2.476    2.476    7.703    7.703 profiler.py:525(_parse_kineto_results)
     6000    2.074    0.000    2.074    0.000 {built-in method torch.conv2d}
        1    1.526    1.526    1.526    1.526 {built-in method torch._C._autograd._disable_profiler}
        3    0.516    0.172    0.516    0.172 {built-in method torch._C._autograd.events}
   116816    0.510    0.000    0.511    0.000 profiler.py:697(<lambda>)
   100636    0.414    0.000    0.416    0.000 profiler.py:595(<listcomp>)
   222972    0.373    0.000    0.456    0.000 profiler.py:545(_device_memory_usage)
   647216    0.363    0.000    0.363    0.000 {built-in method torch._C._autograd.device_type}
   100636    0.360    0.000    0.362    0.000 {built-in method torch._C._autograd.concrete_inputs}
   744420    0.302    0.000    0.302    0.000 {built-in method torch._C._autograd.name}
   116429    0.283    0.000    1.218    0.000 profiler_util.py:674(add)
        1    0.276    0.276    0.354    0.354 profiler.py:529(<listcomp>)
     1794    0.274    0.000    0.274    0.000 {built-in method torch.cat}
   306826    0.273    0.000    0.324    0.000 profiler_util.py:561(cpu_time_total)
   222972    0.250    0.000    0.409    0.000 profiler.py:537(_cpu_memory_usage)



======================================
```

```
=== cProfile for yolov5s ===
         11676007 function calls (11016281 primitive calls) in 20.760 seconds

   Ordered by: internal time
   List reduced from 369 to 15 due to restriction <15>

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
     6000    4.856    0.001    4.856    0.001 {built-in method torch.conv2d}
        1    2.732    2.732    9.141    9.141 profiler.py:525(_parse_kineto_results)
   118037    2.331    0.000    2.357    0.000 profiler_util.py:446(__init__)
        1    1.531    1.531    1.531    1.531 {built-in method torch._C._autograd._disable_profiler}
        3    0.463    0.154    0.463    0.154 {built-in method torch._C._autograd.events}
   223943    0.459    0.000    0.567    0.000 profiler.py:545(_device_memory_usage)
   651320    0.384    0.000    0.384    0.000 {built-in method torch._C._autograd.device_type}
     5700    0.376    0.000    0.376    0.000 {built-in method torch._C._nn.silu_}
     1797    0.351    0.000    0.351    0.000 {built-in method torch.cat}
   751561    0.335    0.000    0.335    0.000 {built-in method torch._C._autograd.name}
   310061    0.329    0.000    0.386    0.000 profiler_util.py:561(cpu_time_total)
   117642    0.315    0.000    1.394    0.000 profiler_util.py:674(add)
   223943    0.285    0.000    0.481    0.000 profiler.py:537(_cpu_memory_usage)
   117642    0.270    0.000    0.302    0.000 profiler_util.py:317(get_key)
      300    0.243    0.001    0.243    0.001 {built-in method torch.max_pool2d}



======================================
```

```
=== cProfile for yolov5m ===
         14210934 function calls (13391734 primitive calls) in 31.506 seconds

   Ordered by: internal time
   List reduced from 369 to 15 due to restriction <15>

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
     8200   11.524    0.001   11.524    0.001 {built-in method torch.conv2d}
        1    2.711    2.711   11.411   11.411 profiler.py:525(_parse_kineto_results)
        1    1.817    1.817    1.817    1.817 {built-in method torch._C._autograd._disable_profiler}
   122311    1.793    0.000    1.797    0.000 {built-in method torch._C._autograd.concrete_inputs}
   291431    1.075    0.000    1.314    0.000 profiler.py:537(_cpu_memory_usage)
   122311    0.891    0.000    0.894    0.000 {built-in method torch._C._autograd.shapes}
     7900    0.667    0.000    0.667    0.000 {built-in method torch._C._nn.silu_}
        3    0.572    0.191    0.572    0.191 {built-in method torch._C._autograd.events}
   291431    0.554    0.000    0.687    0.000 profiler.py:545(_device_memory_usage)
     1797    0.486    0.000    0.486    0.000 {built-in method torch.cat}
   827484    0.464    0.000    0.464    0.000 {built-in method torch._C._autograd.device_type}
   914571    0.394    0.000    0.394    0.000 {built-in method torch._C._autograd.name}
   372521    0.382    0.000    0.449    0.000 profiler_util.py:561(cpu_time_total)
   122311    0.362    0.000    0.364    0.000 profiler.py:595(<listcomp>)
   141137    0.361    0.000    1.604    0.000 profiler_util.py:674(add)
```

```
=== cProfile for yolov5l ===
         16716480 function calls (15739226 primitive calls) in 45.042 seconds

   Ordered by: internal time
   List reduced from 369 to 15 due to restriction <15>

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
    10400   21.535    0.002   21.535    0.002 {built-in method torch.conv2d}
        1    3.001    3.001   13.360   13.360 profiler.py:525(_parse_kineto_results)
        1    2.133    2.133    2.133    2.133 {built-in method torch._C._autograd._disable_profiler}
        1    2.060    2.060    2.138    2.138 profiler_util.py:737(__init__)
   142614    1.045    0.000    1.048    0.000 {built-in method torch._C._autograd.concrete_inputs}
    10100    1.013    0.000    1.013    0.000 {built-in method torch._C._nn.silu_}
   164701    0.803    0.000    0.804    0.000 profiler.py:697(<lambda>)
   164701    0.759    0.000    0.793    0.000 profiler_util.py:446(__init__)
        3    0.683    0.228    0.683    0.228 {built-in method torch._C._autograd.events}
   358669    0.676    0.000    0.837    0.000 profiler.py:545(_device_memory_usage)
     1796    0.577    0.000    0.577    0.000 {built-in method torch.cat}
  1002566    0.558    0.000    0.558    0.000 {built-in method torch._C._autograd.device_type}
  1075657    0.459    0.000    0.459    0.000 {built-in method torch._C._autograd.name}
      300    0.436    0.001    0.436    0.001 {built-in method torch.max_pool2d}
   434121    0.435    0.000    0.511    0.000 profiler_util.py:561(cpu_time_total)
```

```
=== cProfile for yolov5x ===
         19261932 function calls (18124760 primitive calls) in 64.817 seconds

   Ordered by: internal time
   List reduced from 369 to 15 due to restriction <15>

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
    12600   36.851    0.003   36.851    0.003 {built-in method torch.conv2d}
        1    6.077    6.077   14.869   14.869 profiler.py:525(_parse_kineto_results)
        1    2.566    2.566    2.566    2.566 {built-in method torch._C._autograd._disable_profiler}
   188326    2.112    0.000    2.147    0.000 profiler_util.py:446(__init__)
    12300    1.413    0.000    1.413    0.000 {built-in method torch._C._nn.silu_}
   188326    1.002    0.000    1.004    0.000 profiler_util.py:115(<lambda>)
        3    0.850    0.283    0.850    0.283 {built-in method torch._C._autograd.events}
   426228    0.799    0.000    0.991    0.000 profiler.py:545(_device_memory_usage)
     1797    0.700    0.000    0.700    0.000 {built-in method torch.cat}
  1179078    0.662    0.000    0.662    0.000 {built-in method torch._C._autograd.device_type}
     4400    0.631    0.000   25.454    0.006 common.py:177(forward)
  1239364    0.529    0.000    0.529    0.000 {built-in method torch._C._autograd.name}
      300    0.519    0.002    0.519    0.002 {built-in method torch.max_pool2d}
   426228    0.499    0.000    0.843    0.000 profiler.py:537(_cpu_memory_usage)
   496900    0.495    0.000    0.583    0.000 profiler_util.py:561(cpu_time_total)
```

Pytorch Profiler Results

Profiler for yolov5n:

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::conv2d	0.68%	20.060ms	68.29%	2.025s	337.473us	5.20 Gb	0 b	6000
aten::convolution	0.92%	27.317ms	67.61%	2.005s	334.130us	5.20 Gb	0 b	6000
aten::_convolution	1.17%	34.564ms	66.69%	1.977s	329.577us	5.20 Gb	0 b	6000
aten::mkldnn_convolution	64.07%	1.900s	65.52%	1.943s	323.816us	5.20 Gb	0 b	6000
aten::cat	8.05%	238.733ms	8.71%	258.214ms	143.932us	3.16 Gb	3.16 Gb	1794
aten::silu_	7.27%	215.654ms	7.27%	215.654ms	37.834us	0 b	0 b	5700
aten::max_pool2d	0.04%	1.069ms	3.95%	117.143ms	390.477us	60.55 Mb	-113.28 Mb	300
aten::max_pool2d_with_indices	3.91%	116.074ms	3.91%	116.074ms	386.913us	175.78 Mb	175.78 Mb	300
aten::copy_	3.18%	94.213ms	3.18%	94.213ms	26.864us	0 b	0 b	3507
aten::contiguous	0.07%	2.009ms	2.69%	79.877ms	84.616us	996.50 Mb	0 b	944

Self CPU time total: 2.965s

Profiler for yolov5s:

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::conv2d	0.51%	31.292ms	77.84%	4.777s	796.105us	9.61 Gb	0 b	6000
aten::convolution	0.67%	41.148ms	77.33%	4.745s	790.890us	9.61 Gb	0 b	6000
aten::_convolution	0.81%	49.874ms	76.66%	4.704s	784.032us	9.61 Gb	0 b	6000
aten::mkldnn_convolution	74.61%	4.579s	75.85%	4.654s	775.720us	9.61 Gb	0 b	6000
aten::silu_	5.72%	350.878ms	5.72%	350.878ms	61.557us	0 b	0 b	5700
aten::cat	4.93%	302.379ms	5.36%	328.664ms	182.896us	4.72 Gb	4.72 Gb	1797
aten::max_pool2d	0.02%	1.501ms	3.92%	240.424ms	801.413us	118.75 Mb	-231.25 Mb	300
aten::max_pool2d_with_indices	3.89%	238.923ms	3.89%	238.923ms	796.411us	351.56 Mb	351.56 Mb	300
aten::copy_	2.27%	139.033ms	2.27%	139.033ms	38.417us	0 b	0 b	3619
aten::contiguous	0.03%	1.976ms	1.80%	110.380ms	112.633us	1.07 Gb	0 b	980

Self CPU time total: 6.136s

Profiler for yolov5m:

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::conv2d	0.42%	55.912ms	85.42%	11.395s	1.390ms	17.68 Gb	0 b	8200
aten::convolution	0.49%	65.188ms	85.00%	11.340s	1.383ms	17.68 Gb	0 b	8200
aten::_convolution	0.56%	74.306ms	84.51%	11.274s	1.375ms	17.68 Gb	0 b	8200
aten::mkldnn_convolution	82.79%	11.045s	83.95%	11.200s	1.366ms	17.68 Gb	0 b	8200
aten::silu_	4.70%	626.789ms	4.70%	626.789ms	79.340us	0 b	0 b	7900
aten::cat	3.24%	432.779ms	3.46%	461.251ms	256.679us	6.29 Gb	6.29 Gb	1797
aten::max_pool2d	0.02%	2.198ms	2.56%	341.992ms	1.140ms	175.78 Mb	-351.56 Mb	300
aten::max_pool2d_with_indices	2.55%	339.794ms	2.55%	339.794ms	1.133ms	527.34 Mb	527.34 Mb	300
aten::copy_	0.96%	128.460ms	0.96%	128.460ms	35.408us	0 b	0 b	3628
aten::add	0.91%	121.595ms	0.91%	121.595ms	61.103us	2.65 Gb	2.65 Gb	1990

Self CPU time total: 13.341s

Profiler for yolov5l:

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::conv2d	0.32%	75.937ms	89.01%	21.364s	2.054ms	28.19 Gb	0 b	10400
aten::convolution	0.36%	87.455ms	88.69%	21.288s	2.047ms	28.19 Gb	0 b	10400
aten::_convolution	0.41%	97.544ms	88.33%	21.201s	2.039ms	28.19 Gb	0 b	10400
aten::mkldnn_convolution	86.95%	20.869s	87.92%	21.103s	2.029ms	28.19 Gb	0 b	10400
aten::silu_	4.00%	959.553ms	4.00%	959.553ms	95.005us	0 b	0 b	10100
aten::cat	2.19%	525.811ms	2.30%	552.396ms	307.570us	7.85 Gb	7.85 Gb	1796
aten::max_pool2d	0.01%	1.802ms	1.80%	432.120ms	1.440ms	234.38 Mb	-468.75 Mb	300
aten::max_pool2d_with_indices	1.79%	430.317ms	1.79%	430.317ms	1.434ms	703.12 Mb	703.12 Mb	300
aten::add	1.20%	288.050ms	1.20%	288.050ms	107.201us	5.28 Gb	5.28 Gb	2687
aten::empty	0.73%	174.365ms	0.73%	174.365ms	7.413us	29.49 Gb	29.49 Gb	23523

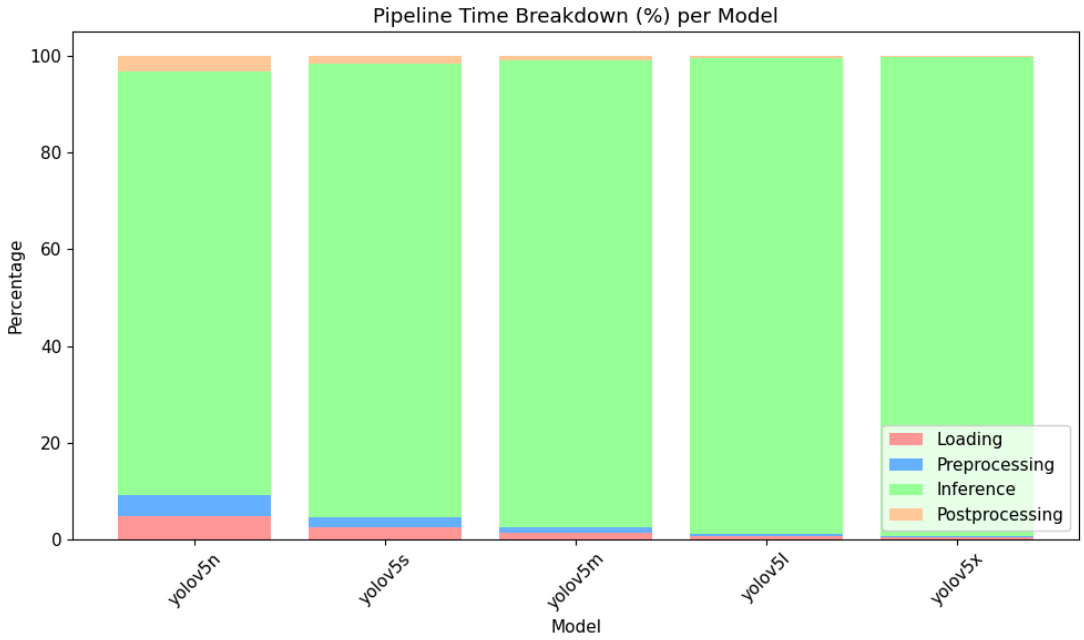
Self CPU time total: 24.002s

Profiler for yolov5x:

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::conv2d	0.27%	107.747ms	91.37%	36.617s	2.906ms	41.14 Gb	0 b	12600
aten::convolution	0.29%	117.400ms	91.10%	36.510s	2.898ms	41.14 Gb	0 b	12600
aten::_convolution	0.32%	127.874ms	90.81%	36.392s	2.888ms	41.14 Gb	0 b	12600
aten::mkldnn_convolution	89.53%	35.880s	90.49%	36.264s	2.878ms	41.14 Gb	0 b	12600
aten::silu_	3.35%	1.341s	3.35%	1.341s	109.040us	0 b	0 b	12300
aten::cat	1.61%	645.393ms	1.68%	672.864ms	374.437us	9.42 Gb	9.42 Gb	1797
aten::add	1.33%	534.396ms	1.33%	534.396ms	157.639us	8.79 Gb	8.79 Gb	3390
aten::max_pool2d	0.01%	2.062ms	1.29%	515.041ms	1.717ms	292.97 Mb	-585.94 Mb	300
aten::max_pool2d_with_indices	1.28%	512.979ms	1.28%	512.979ms	1.710ms	878.91 Mb	878.91 Mb	300
aten::empty	0.75%	300.142ms	0.75%	300.142ms	10.745us	42.53 Gb	42.53 Gb	27933

Self CPU time total: 40.077s

4. Time Breakdown Across Pipeline Stages




```
Model: yolov5n
        Loading: 4.69%
  Preprocessing: 4.48%
      Inference: 87.59%
 Postprocessing: 3.24%

Model: yolov5s
        Loading: 2.39%
  Preprocessing: 2.18%
      Inference: 93.79%
 Postprocessing: 1.64%

Model: yolov5m
        Loading: 1.24%
  Preprocessing: 1.16%
      Inference: 96.78%
 Postprocessing: 0.82%

Model: yolov5l
        Loading: 0.63%
  Preprocessing: 0.59%
      Inference: 98.34%
 Postprocessing: 0.44%

Model: yolov5x
        Loading: 0.37%
  Preprocessing: 0.34%
      Inference: 99.02%
 Postprocessing: 0.26%
```

5. Optimization Suggestions and Reasoning

I chose to go with 3 optimizations for the following reasons:

- Batch Processing: Allows multiple inputs to be processed in parallel, which improves throughput, i.e. files per second (FPS)
- ONNX format: Allows model execution through ONNX Runtime inference engine, which provides optimization like operator fusing and memory reuse. This helps reducing latency
- Dynamic Quantization: reduces size of the model and helps improve inference speed as it converts some weights from FP32 to INT8.

6. Optimization Results

Model	Latency (ms)	FPS	Params (M)	Size (MB)	GFLOPs	Utilization (%)	Bound Type
yolov5n	19.75	50.63	1.87	7.12	2.23	27.60	Compute Bound
yolov5s	51.54	19.40	7.23	27.56	8.22	38.89	Compute Bound
yolov5m	130.26	7.68	21.17	80.77	24.44	45.76	Compute Bound
yolov5l	242.73	4.12	46.53	177.51	54.50	54.76	Compute Bound
yolov5x	421.28	2.37	86.71	330.75	102.73	59.47	Compute Bound

YOLOv5 Performance Analysis & Optimization: A Brief Summary

Disha Pant

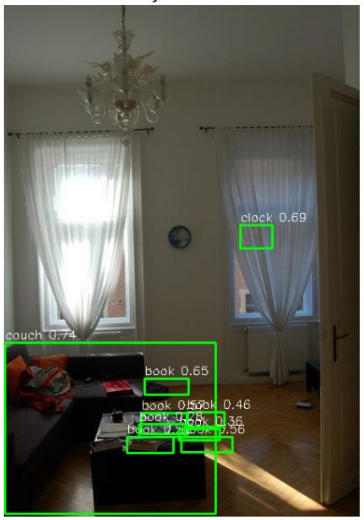
Sample Inferences

yolov5n

yolov5s

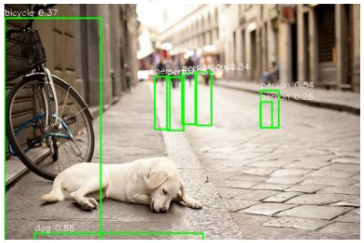
yolov5m

yolov5l

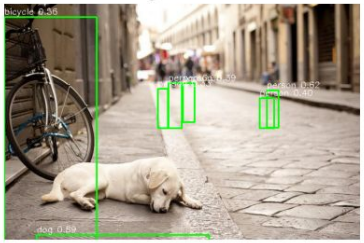


yolov5x

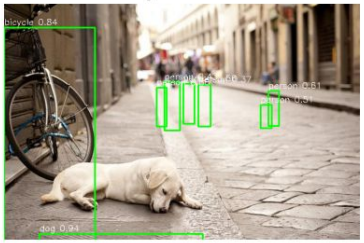
yolov5n



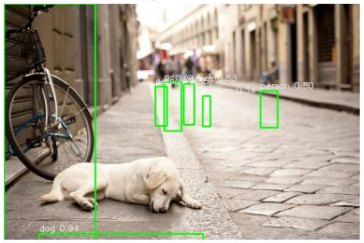
yolov5s



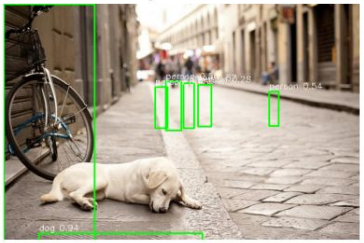
yolov5m



yolov5l



yolov5x



Latency and FPS

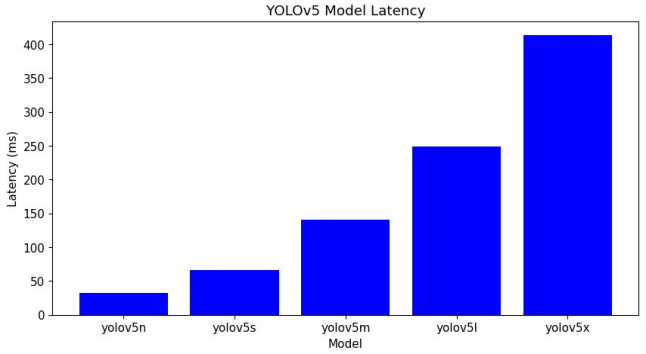
Specs:
CPU Name: AMD Ryzen 7 6800HS
Threads per core: 2
Core(s) per socket: 8
Socket(s): 1
Clock Rate: 3200 - 4700 MHz
RAM: 16 GB

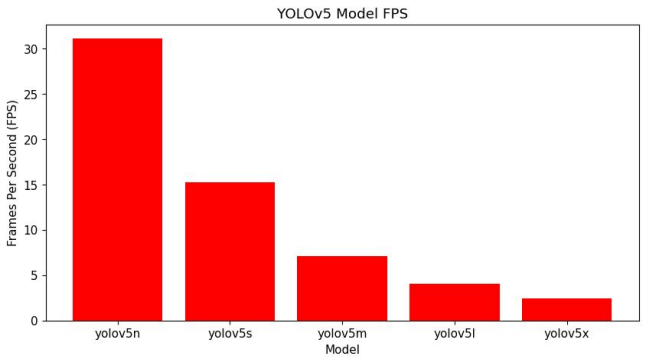
Peak Bandwidth: 28.1 GB/s
Theoretical GFLOPS: ~410 GFLOPS

On the basis of inference time, latency and FPS were calculated:

- Latency (ms) = (Total inference time for all images / number of images) * 1000
- FPS = number of images / total inference time in seconds

As expected, latency increased with increase in model size and FPS decreased.

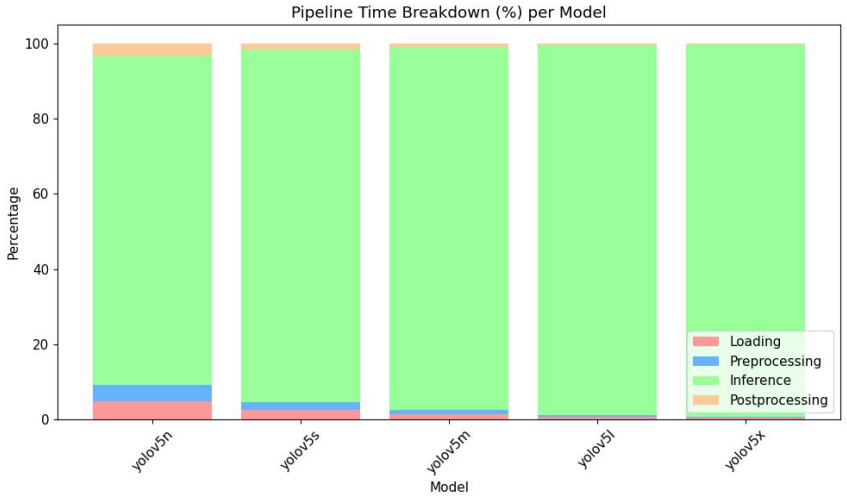




Model	Params (M)	Size (MB)	GFLOPs	Utilization (%)	Bound Type
yolov5n	1.87	7.12	2.23	16.96	Compute Bound
yolov5s	7.23	27.56	8.22	30.50	Compute Bound
yolov5m	21.17	80.77	24.44	42.42	Compute Bound
yolov5l	46.53	177.51	54.50	53.29	Compute Bound
yolov5x	86.71	330.75	102.73	60.59	Compute Bound

```
======================================================================
Per-Layer Utilization Summary
======================================================================

Model: yolov5n
Highest Utilization Layers:
    model.model.6                          : 1.89%
    model.model.6.m                        : 1.49%
    model.model.4                          : 1.39%
Lowest Utilization Layers:
    model.model.23.cv3.act                 : 0.00%
    model.model.23.m.0.cv1.act             : 0.00%
    model.model.23.m.0.cv2.act             : 0.00%
```



Roofline Bound analysis was done by calculating peak theoretical OI and comparing with obtained OI.

Per-layer analysis was done by using fvcore to find FLOPs per layer and determine which had highest and lowest utilization

Stage-wise breakdown was done by timing each section of the pipeline and then splitting the total time into a percentage wise breakdown.

Profiling Outputs for cProfile

```
=== cProfile for yolov5n ===
        11568142 function calls (10913825 primitive calls) in 15.538 seconds

   Ordered by: internal time
   List reduced from 558 to 15 due to restriction <15>

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        1    2.476    2.476    7.703    7.703 profiler.py:525(_parse_kineto_results)
     6000    2.074    0.000    2.074    0.000 {built-in method torch.conv2d}
        1    1.526    1.526    1.526    1.526 {built-in method torch._C._autograd._disable_profiler}
        3    0.516    0.172    0.516    0.172 {built-in method torch._C._autograd.events}
   116816    0.510    0.000    0.511    0.000 profiler.py:697(<lambda>)
   100636    0.414    0.000    0.416    0.000 profiler.py:595(<listcomp>)
   222972    0.373    0.000    0.456    0.000 profiler.py:545(_device_memory_usage)
   647216    0.363    0.000    0.363    0.000 {built-in method torch._C._autograd.device_type}
   100636    0.360    0.000    0.362    0.000 {built-in method torch._C._autograd.concrete_inputs}
   744420    0.302    0.000    0.302    0.000 {built-in method torch._C._autograd.name}
   116429    0.283    0.000    1.218    0.000 profiler_util.py:674(add)
        1    0.276    0.276    0.354    0.354 profiler.py:529(<listcomp>)
     1794    0.274    0.000    0.274    0.000 {built-in method torch.cat}
   306826    0.273    0.000    0.324    0.000 profiler_util.py:561(cpu_time_total)
   222972    0.250    0.000    0.409    0.000 profiler.py:537(_cpu_memory_usage)



====================================
```

```
=== cProfile for yolov5x ===
        19261932 function calls (18124760 primitive calls) in 64.817 seconds

   Ordered by: internal time
   List reduced from 369 to 15 due to restriction <15>

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
    12600   36.851    0.003   36.851    0.003 {built-in method torch.conv2d}
        1    6.077    6.077   14.869   14.869 profiler.py:525(_parse_kineto_results)
        1    2.566    2.566    2.566    2.566 {built-in method torch._C._autograd._disable_profiler}
   188326    2.112    0.000    2.147    0.000 profiler_util.py:446(__init__)
    12300    1.413    0.000    1.413    0.000 {built-in method torch._C._nn.silu_}
   188326    1.002    0.000    1.004    0.000 profiler_util.py:115(<lambda>)
        3    0.850    0.283    0.850    0.283 {built-in method torch._C._autograd.events}
   426228    0.799    0.000    0.991    0.000 profiler.py:545(_device_memory_usage)
     1797    0.700    0.000    0.700    0.000 {built-in method torch.cat}
  1179078    0.662    0.000    0.662    0.000 {built-in method torch._C._autograd.device_type}
     4400    0.631    0.000   25.454    0.006 common.py:177(forward)
  1239364    0.529    0.000    0.529    0.000 {built-in method torch._C._autograd.name}
      300    0.519    0.002    0.519    0.002 {built-in method torch.max_pool2d}
   426228    0.499    0.000    0.843    0.000 profiler.py:537(_cpu_memory_usage)
   496900    0.495    0.000    0.583    0.000 profiler_util.py:561(cpu_time_total)
```

Profiling Outputs for Pytorch Profiler

Profiler for yolov5n:

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::conv2d	0.68%	20.060ms	68.29%	2.025s	337.473us	5.20 Gb	0 b	6000
aten::convolution	0.92%	27.317ms	67.61%	2.005s	334.130us	5.20 Gb	0 b	6000
aten::_convolution	1.17%	34.564ms	66.69%	1.977s	329.577us	5.20 Gb	0 b	6000
aten::mkldnn_convolution	64.07%	1.900s	65.52%	1.943s	323.816us	5.20 Gb	0 b	6000
aten::cat	8.05%	238.733ms	8.71%	258.214ms	143.932us	3.16 Gb	3.16 Gb	1794
aten::silu_	7.27%	215.654ms	7.27%	215.654ms	37.834us	0 b	0 b	5700
aten::max_pool2d	0.04%	1.069ms	3.95%	117.143ms	390.477us	60.55 Mb	-113.28 Mb	300
aten::max_pool2d_with_indices	3.91%	116.074ms	3.91%	116.074ms	386.913us	175.78 Mb	175.78 Mb	300
aten::copy_	3.18%	94.213ms	3.18%	94.213ms	26.864us	0 b	0 b	3507
aten::contiguous	0.07%	2.009ms	2.69%	79.877ms	84.616us	996.50 Mb	0 b	944

Self CPU time total: 2.965s

Profiler for yolov5x:

Name	Self CPU %	Self CPU	CPU total %	CPU total	CPU time avg	CPU Mem	Self CPU Mem	# of Calls
aten::conv2d	0.27%	107.747ms	91.37%	36.617s	2.906ms	41.14 Gb	0 b	12600
aten::convolution	0.29%	117.400ms	91.10%	36.510s	2.898ms	41.14 Gb	0 b	12600
aten::_convolution	0.32%	127.874ms	90.81%	36.392s	2.888ms	41.14 Gb	0 b	12600
aten::mkldnn_convolution	89.53%	35.880s	90.49%	36.264s	2.878ms	41.14 Gb	0 b	12600
aten::silu_	3.35%	1.341s	3.35%	1.341s	109.040us	0 b	0 b	12300
aten::cat	1.61%	645.393ms	1.68%	672.864ms	374.437us	9.42 Gb	9.42 Gb	1797
aten::add	1.33%	534.396ms	1.33%	534.396ms	157.639us	8.79 Gb	8.79 Gb	3390
aten::max_pool2d	0.01%	2.062ms	1.29%	515.041ms	1.717ms	292.97 Mb	-585.94 Mb	300
aten::max_pool2d_with_indices	1.28%	512.979ms	1.28%	512.979ms	1.710ms	878.91 Mb	878.91 Mb	300
aten::empty	0.75%	300.142ms	0.75%	300.142ms	10.745us	42.53 Gb	42.53 Gb	27933

Self CPU time total: 40.077s

Optimizations & Results

Optimizations chosen:

- **Batch Processing**: Allows multiple inputs to be processed in parallel, which improves throughput, i.e. files per second (FPS)
- **ONNX format**: Allows model execution through ONNX Runtime inference engine, which provides optimization like operator fusing and memory reuse. This helps reducing latency
- **Dynamic Quantization**: reduces size of the model and helps improve inference speed as it converts some weights from FP32 to INT8.

Model	Latency (ms)	FPS	Params (M)	Size (MB)	GFLOPs	Utilization (%)	Bound Type
yolov5n	19.75	50.63	1.87	7.12	2.23	27.60	Compute Bound
yolov5s	51.54	19.40	7.23	27.56	8.22	38.89	Compute Bound
yolov5m	130.26	7.68	21.17	80.77	24.44	45.76	Compute Bound
yolov5l	242.73	4.12	46.53	177.51	54.50	54.76	Compute Bound
yolov5x	421.28	2.37	86.71	330.75	102.73	59.47	Compute Bound