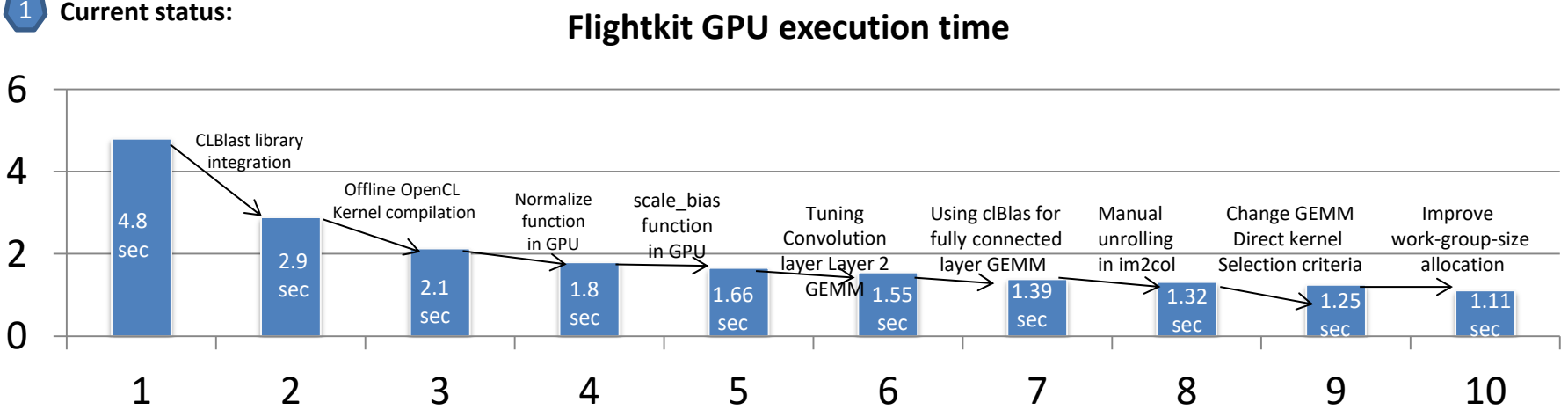
YOLO Execution speed improvement

Current status

1 **Current status:**



YOLO Execution speed improvement

Average time per layer for 100 test images:

Layer #	Type	Filters	Kernel size /Stride	Input size	Output size	Time (ms)
0	Convolution layer	16	3 x 3 / 1	448 x 448 x 3	448 x 448 x 16	220.2344
1	Max pooling layer		2 x 2 / 2	448 x 448 x 16	224 x 224 x 16	8.8364
2	Convolution layer	32	3 x 3 / 1	224 x 224 x 16	224 x 224 x 32	228.2602
3	Max pooling layer		2 x 2 / 2	224 x 224 x 32	112 x 112 x 32	4.7590
4	Convolution layer	64	3 x 3 / 1	112 x 112 x 32	112 x 112 x 64	115.8306
5	Max pooling layer		2 x 2 / 2	112 x 112 x 64	56 x 56 x 64	2.5943
6	Convolution layer	128	3 x 3 / 1	56 x 56 x 64	56 x 56 x 128	85.9564
7	Max pooling layer		2 x 2 / 2	56 x 56 x 128	28 x 28 x 128	2.0354
8	Convolution layer	256	3 x 3 / 1	28 x 28 x 128	28 x 28 x 256	89.7564
9	Max pooling layer		2 x 2 / 2	28 x 28 x 256	14 x 14 x 256	1.5034
10	Convolution layer	512	3 x 3 / 1	14 x 14 x 256	14 x 14 x 512	103.6955
11	Max pooling layer		2 x 2 / 2	14 x 14 x 512	7 x 7 x 512	1.0268
12	Convolution layer	1024	3 x 3 / 1	7 x 7 x 512	7 x 7 x 1024	168.3409
13	Convolution layer	256	3 x 3 / 1	7 x 7 x 1024	7 x 7 x 256	85.4873
14	Connected			12544	1029	16.9828
15	Detection					0.5438
				Total time / image		**1.1 sec**

Im2col = 12 ms
GEMM = 140 ms
BatchNorm = 61 ms
Activation = 8 ms

GEMM = General Matrix Multiply

YOLO Execution speed improvement

Layer #	Type	Filters	Kernel size /Stride	Input size	Output size	Time (ms)
0	Convolution layer	16	3 x 3 / 1	448 x 448 x 3	448 x 448 x 16	220.2344
1	Max pooling layer		2 x 2 / 2	448 x 448 x 16	224 x 224 x 16	8.8364
2	Convolution layer	32	3 x 3 / 1	224 x 224 x 16	224 x 224 x 32	228.2602
3	Max pooling layer		2 x 2 / 2	224 x 224 x 32	112 x 112 x 32	4.7590
4	Convolution layer	64	3 x 3 / 1	112 x 112 x 32	112 x 112 x 64	115.8306
5	Max pooling layer		2 x 2 / 2	112 x 112 x 64	56 x 56 x 64	2.5943
6	Convolution layer	128	3 x 3 / 1	56 x 56 x 64	56 x 56 x 128	85.9564
7	Max pooling layer		2 x 2 / 2	56 x 56 x 128	28 x 28 x 128	2.0354
8	Convolution layer	256	3 x 3 / 1	28 x 28 x 128	28 x 28 x 256	89.7564
9	Max pooling layer		2 x 2 / 2	28 x 28 x 256	14 x 14 x 256	1.5034
10	Convolution layer	512	3 x 3 / 1	14 x 14 x 256	14 x 14 x 512	103.6955
11	Max pooling layer		2 x 2 / 2	14 x 14 x 512	7 x 7 x 512	1.0268
12	Convolution layer	1024	3 x 3 / 1	7 x 7 x 512	7 x 7 x 1024	168.3409
13	Convolution layer	256	3 x 3 / 1	7 x 7 x 1024	7 x 7 x 256	85.4873
14	Connected layer			12544	1029	16.9828
15	Detection layer					0.5438
				Total time / image		**1.1 sec**

GEMM time > 900ms
> 90% of
total time

YOLO Execution speed improvement

Libraries used for optimizing GEMM operation

- CLBlast
- clBLAS

GPU optimization

- NNPACK — CPU optimization

Not experimented libraries

- ViennaCL

Other libraries already considered:

Library / Tool	Issue / Bottleneck faced	Remarks
Snapdragon Neural Processing Engine (SNPE)	• Supports Snapdragon 820, 835, 625, 650, 652, 653, 660, 630, 450, 829Am	FlightKit ➔ Snapdragon 801 ➔ Not supported by SNPE RS800 ➔ Snapdragon 625 ➔ Supported by SNPE
	• Requires Caffe (or) TensorFlow model	Need to Convolution layerert Darknet model to TensorFlow model (Issue link)
Adreno GPU SDK	Contains very less OpenCL implementation samples	
ARM compute library	Does not support OpenCL 1.1 embedded profile (Issue link)	FlightKit ➔ OpenCL 1.1 Embedded profile RS800 ➔ OpenCL 1.2
Tuning clBLAS	Cannot tune clBLAS on FlightKit (Similar Issue link)	
clMAGMA	Unable to install in FlightKit hardware	Very less documentation & support

YOLO Execution speed improvement

Next steps:

- Optimize GEMM function for Adreno GPU
  - Need to survey & study about Adreno 330 GPU (if available in internet)
- If possible, check execution speed on RS800 (since it has better GPU configuration)