WINTECHCON

Edge Acceleration of Computer Vision and Deep Learning Algorithms using OpenCL

September 27, 2019

Bakshree Mishra
Intel Corporation

On the Edge : Computer Vision and Machine Learning

Significance in IOT

- Industrial automation
- Enable real time as well as offline analytics

Problem Statement

- Conveyor belt with moving parts
- Over head camera doing online analysis such as OCR
- Support high camera frame-rate

Challenges

- Real-time processing
- Variable latency of data transfer in cloud

Proposed Solution

- Custom hardware accelerator
- FPGA + OpenCL

OpenCL : Quick Overview

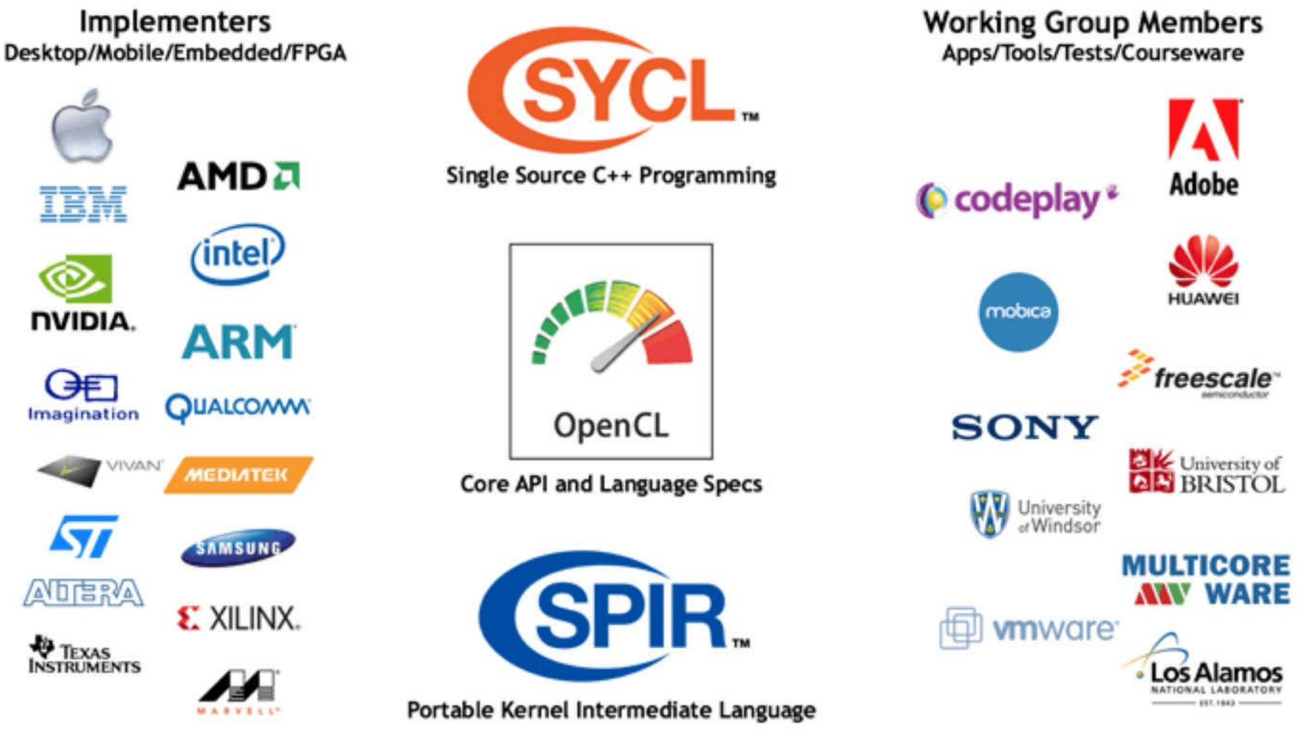
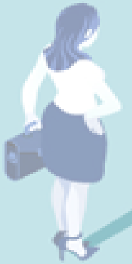


Image from https://www.khronos.org/

- Open Standard for heterogeneous and cross-platform computing
- Framework maintained by the Khronos group
- Consists of Host code and Device code
- Device code is instantiated on the accelerator/co-processor



Solution for Fast OCR

- Algorithm:
  - Sensor image pre-processing
  - Connected Components Labeling
  - CNN for OCR
  - Character stitching (post-processing)

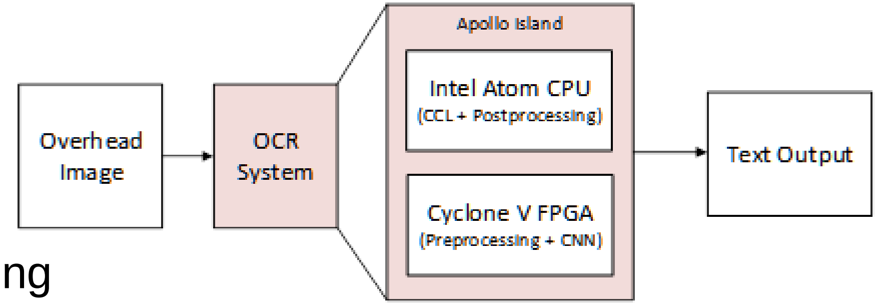


Fig. 1 Industrial Setup for fast OCR

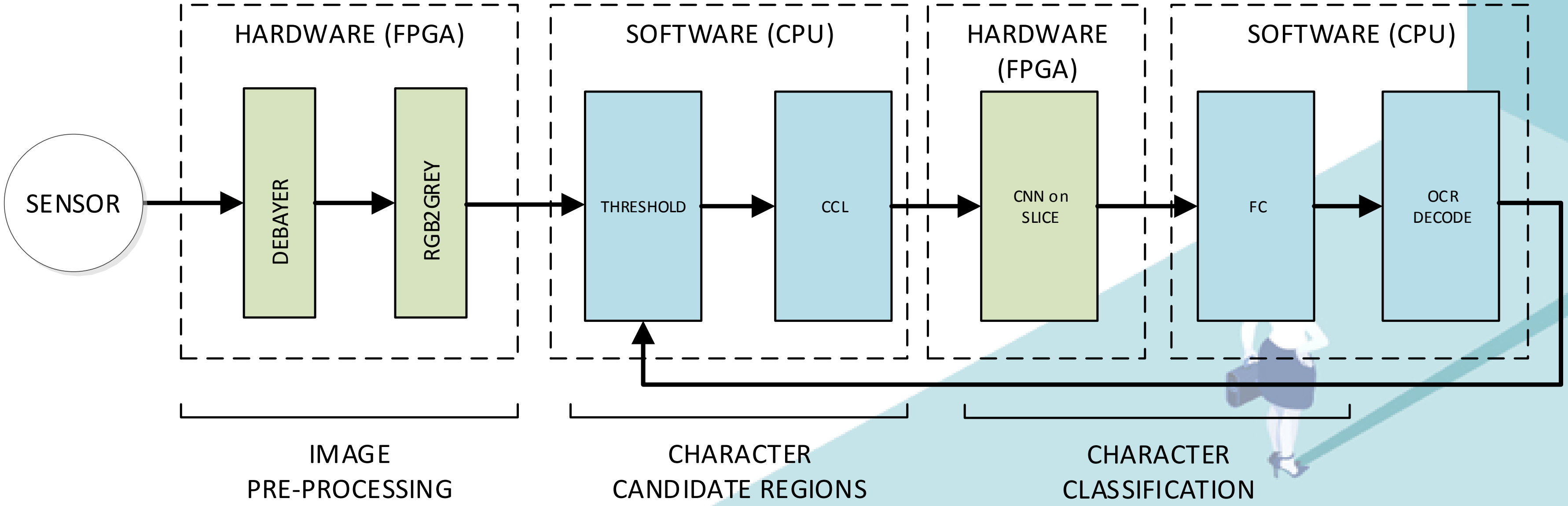


Fig. 3 Pipeline for OCR Acceleration

CNN topology and Computation analysis

- Convolutional Neural Networks (CNN) are a class of machine learning algorithms which have recently performed very well in image classification and are very widely used for machine vision.
- In OCR, the input is an image and the output is a choice among a set of characters that are to be recognized.

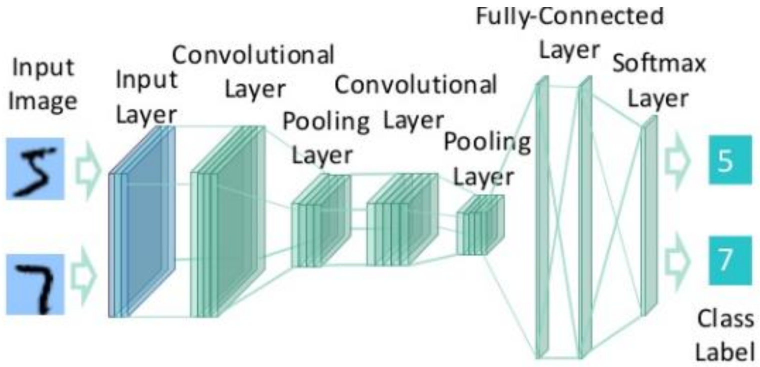


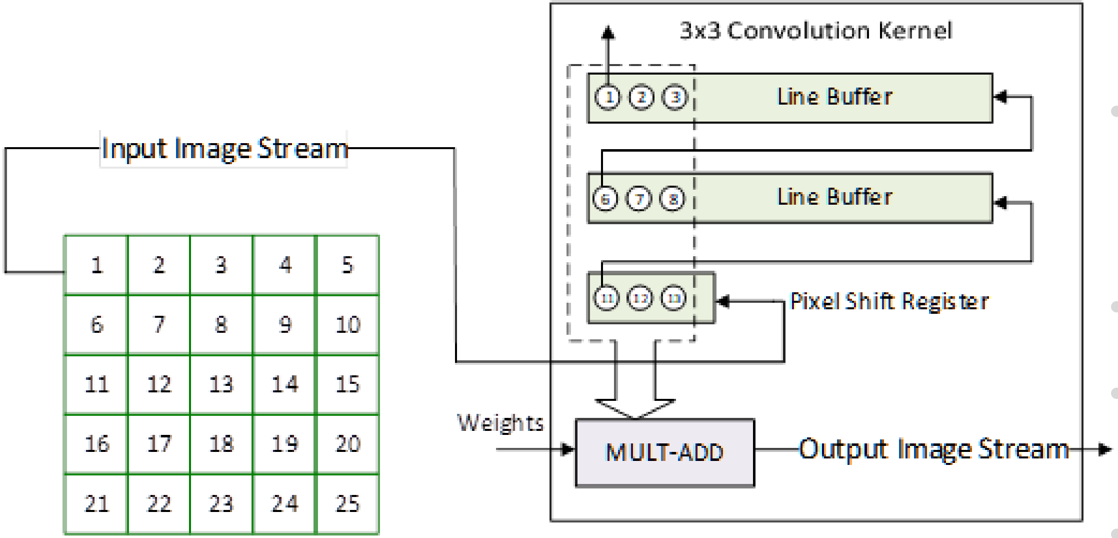
Fig. 2 CNN topology for OCR

CNN PER LAYER COMPUTE

Layer	Nodes	Input Size	Compute
Convolution Layer 1	16	16x16	36864
Pooling Layer 1	16	16x16	4096
Convolution Layer 2	64	8x8x16	589824
Pooling Layer 2	64	8x8x16	65536
Fully Connected Layer 1	128	4x4x64	131072
Fully Connected Layer 2	256	128	32768

- The network topology:
  - two convolution and pooling layers
  - two fully connected layers
  - mask size 3x3 for convolutions.

High Level Design | Convolution Kernel

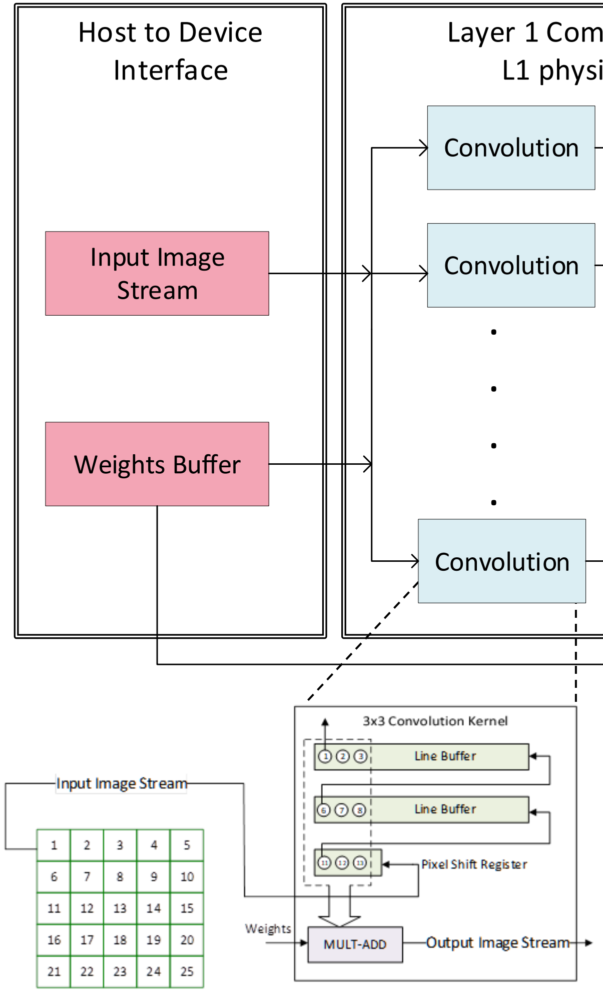


1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Figure 4 Raster Scan Architecture

- Convolutions take place in raster scan order
- Processing image slices as a 1D data stream enables bypassing the memory fetch overhead
- In OCR, the input is an image and the output is a choice among a set of characters that are to be recognized.
- The nodes are connected in a pipelined fashion
- Each node receives an input pixel and generates an output pixel every clock cycle.
- Architecture is scalable to the size of the filter as well as stride,
- Can accelerate both traditional as well as deep learning based computer vision algorithms.

High Level Design



- Modular design that can be scaled as per network topology
- Nodes pipelined to buffer pre-fetched data and compute output every clock cycle

High Level Design | Partials Compute

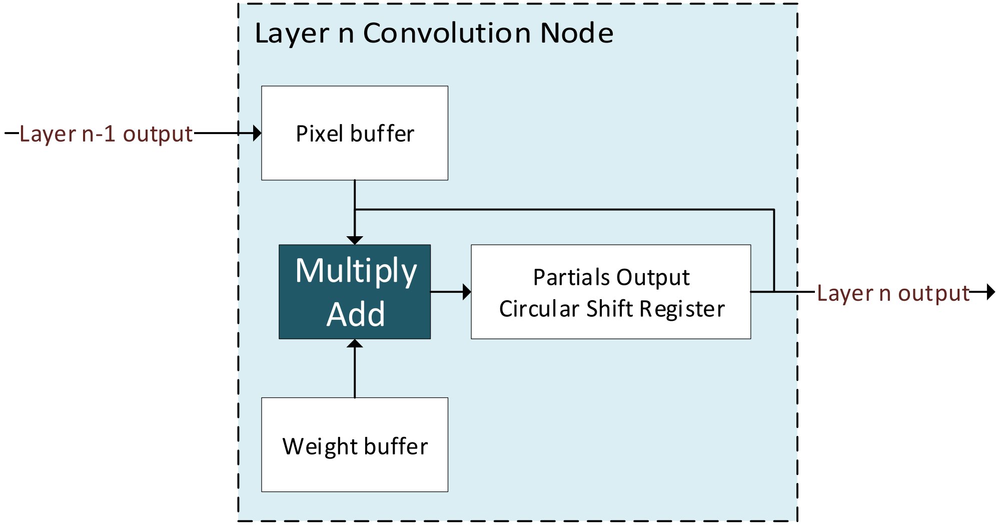


Figure 6 High Level Partial Compute Block

- Nodes in CNN layer operate on multiple nodes' outputs from previous layer
- Data transfer to and from DDR is expensive
- No-stall partials compute method to start computing for n^{th} layer
- No need to wait for all nodes in $(n\text{-}1)^{st}$ to finish

High Level Architecture and Data flow

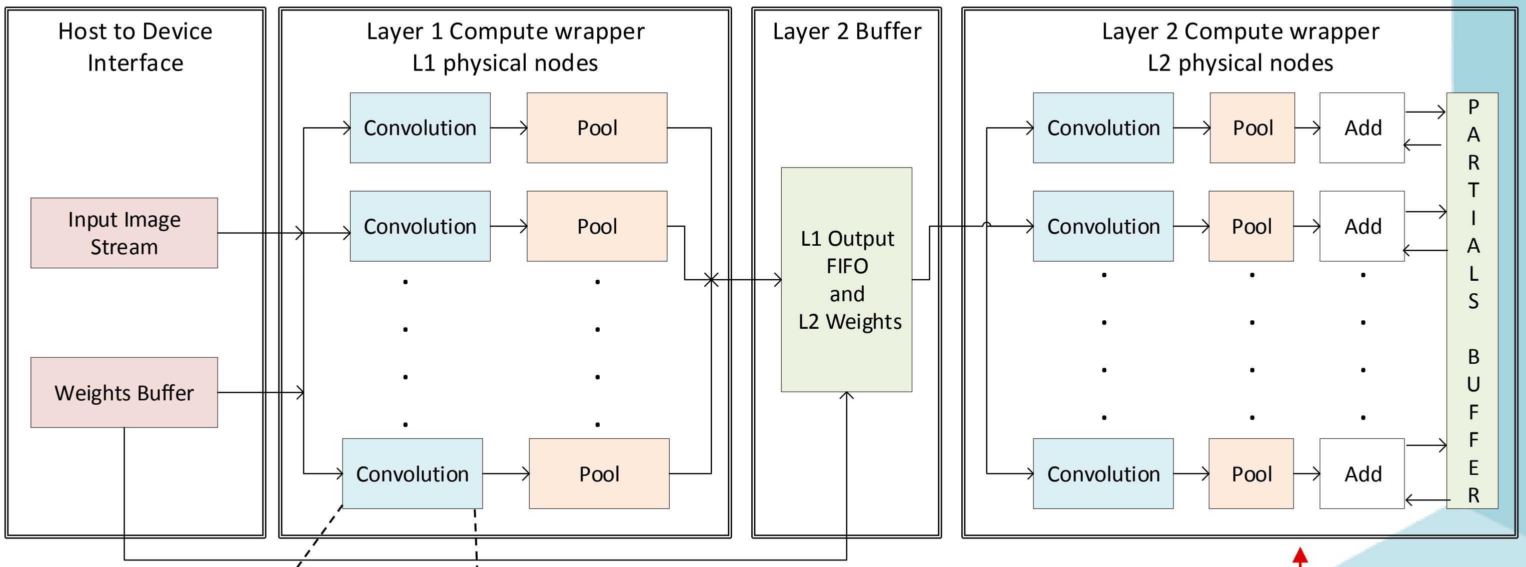
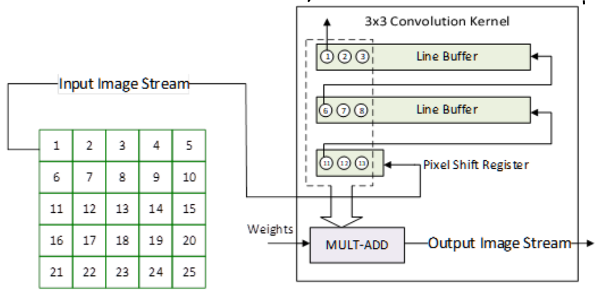
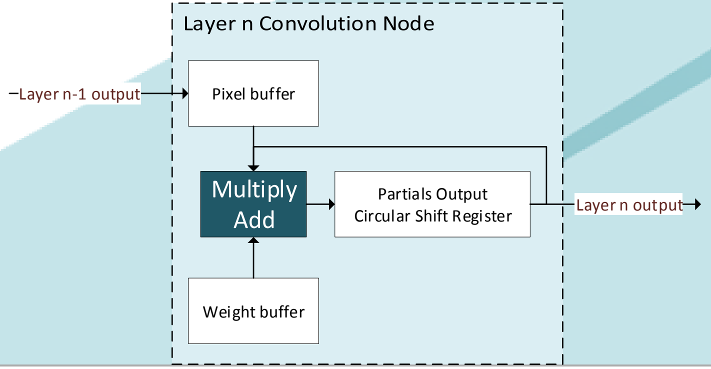


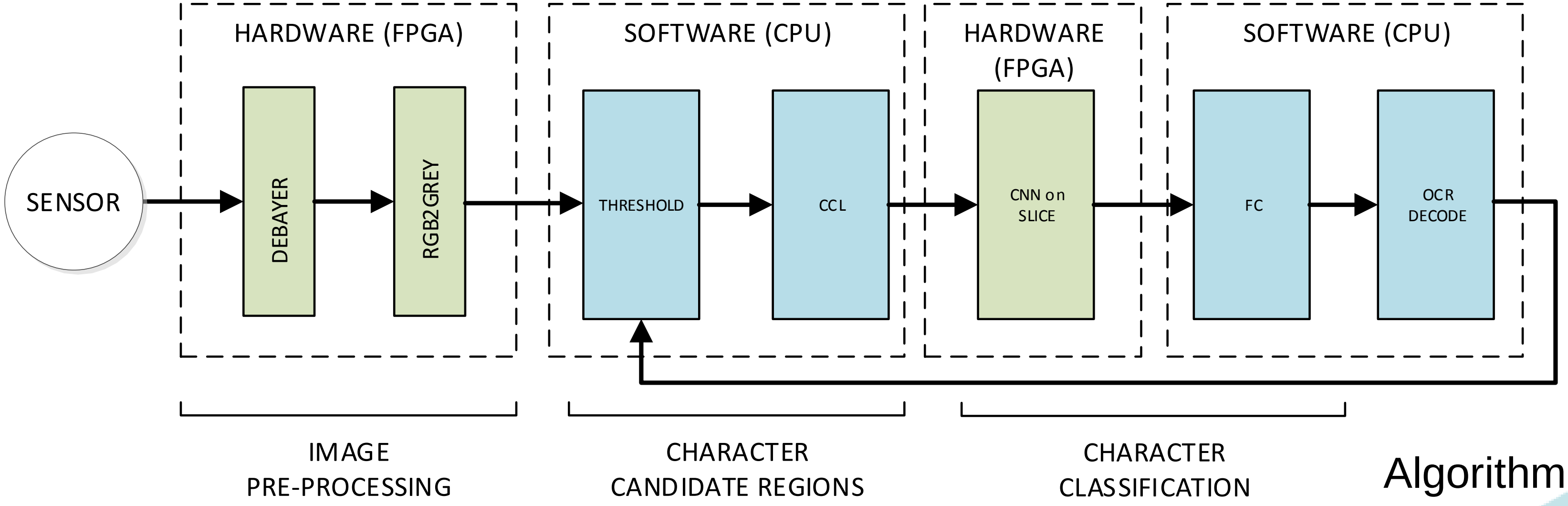
Figure 5 High Level Hardware Architecture for CNN Acceleration



1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25



Experimental Results



FPGA Resource Stats:

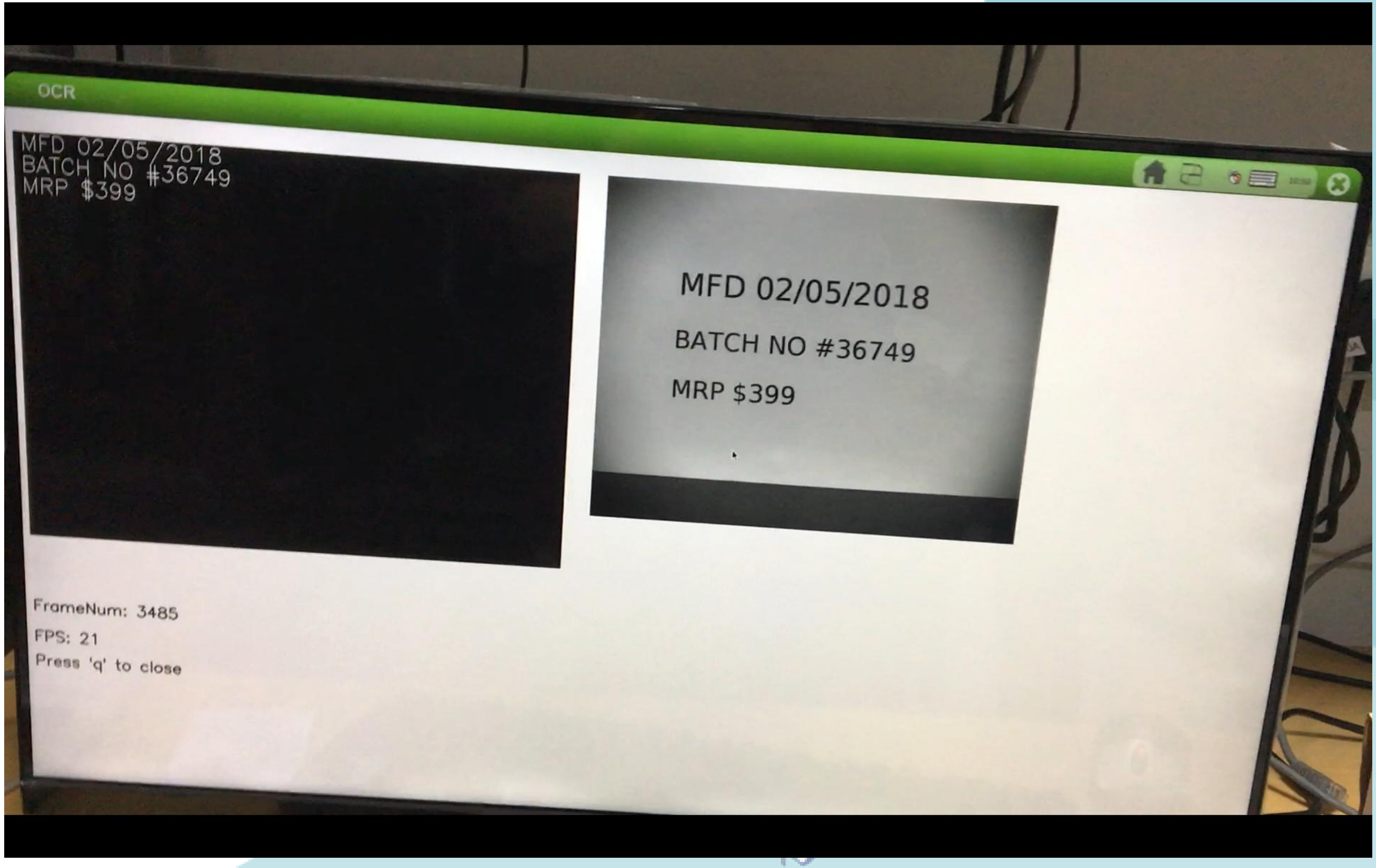
Resource	Percentage Used
ALM	88
DSP	76
M10K	44

Algorithm Acceleration per stage:

Threaded Operations	CPU (ms)	FPGA (ms)
IO Channel (FPGA)	-	8.3
CCL+ Threshold	25	-
CNN-Conv (FPGA)	200	8
CNN - FC	15	-

Experimental Results

- The hardware achieves 25x performance over convolution layers.
- The software flow could originally compute OCR at 4 FPS
- CNN accelerator boosts the end-to-end performance by 7.5X by running at 30FPS.
- Photo on the right is a snap at early stage of design



```
MFD 02/05/2018
BATCH NO #36749
MRP $399
```

MFD 02/05/2018

BATCH NO #36749

MRP $399

```
FrameNum: 3485
FPS: 21
Press 'q' to close
```

Take-aways

Complete self-sufficiency of the solution

Low cost solution for compute-on-edge industrial solution

Maximal usage of CPU and FPGA at all times

Reusable architecture for traditional Computer Vision operations as well as CNNs

Reduced engineering efforts and faster time to market by using OpenCL

RTL level maximal efficiency and performance extracted from OpenCL implementation

Discussion and Further Scope

- Using OpenCL to implement the design helped in making quick iterations and bringing up the accelerator
- The custom design for CNN was adapted for another traditional CV algorithm use-case with minor changes
- The scalable design is gated only by FPGA resource constraints
- Current design is only for CNN, other types of networks such as RNNs, LSTMs, GANs need further work
- Current methodology takes advantage of raster scan order for image processing, may need other optimizations for other kinds of inputs

References

- Abdelouahab, Kamel, et al. "Accelerating CNN inference on FPGAs: A Survey." arXiv preprint arXiv:1806.01683 (2018).
- Zhao, Wenlai, et al. "F-CNN: An FPGA-based framework for training convolutional neural networks." 2016 IEEE 27th International Conference on Application-specific Systems, Architectures and Processors (ASAP). IEEE, 2016.
- D. Wang, K. Xu and D. Jiang, "PipeCNN: An OpenCL-based open-source FPGA accelerator for convolution neural networks," 2017 International Conference on Field Programmable Technology (ICFPT), Melbourne, VIC, 2017, pp. 279-282.
- OpenVINO - Open Visual Inference and Neural Network Optimization Toolkit, Intel Corporation, https://software.intel.com/enus/openvino-toolkit.
- Zhang, Chen, et al. "Optimizing fpga-based accelerator design for deep convolutional neural networks." Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. ACM, 2015.
- Meloni, Paolo, et al. "Curbing the roofline: a scalable and flexible architecture for CNNs on FPGA." Proceedings of the ACM International Conference on Computing Frontiers. ACM, 2016.
- Liu, B.; Zou, D.; Feng, L.; Feng, S.; Fu, P.; Li, J. An FPGA-Based CNN Accelerator Integrating Depthwise Separable Convolution. Electronics 2019, 8, 281.