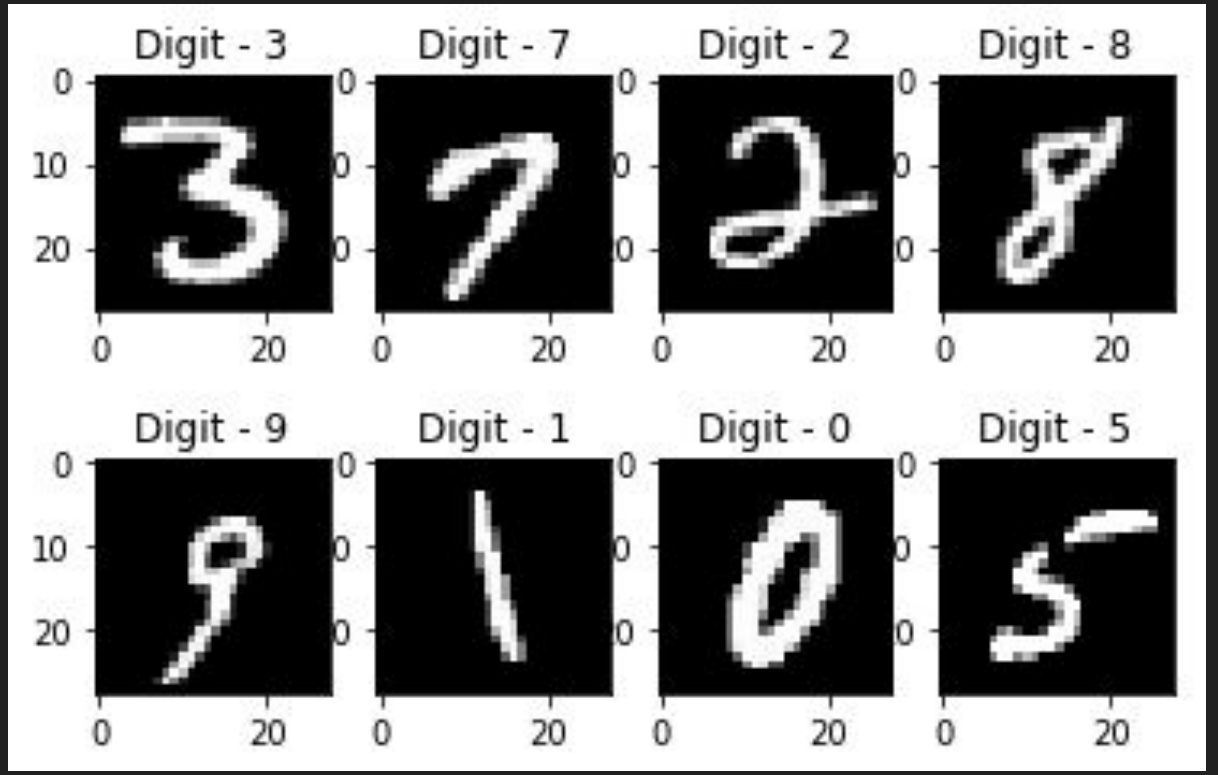
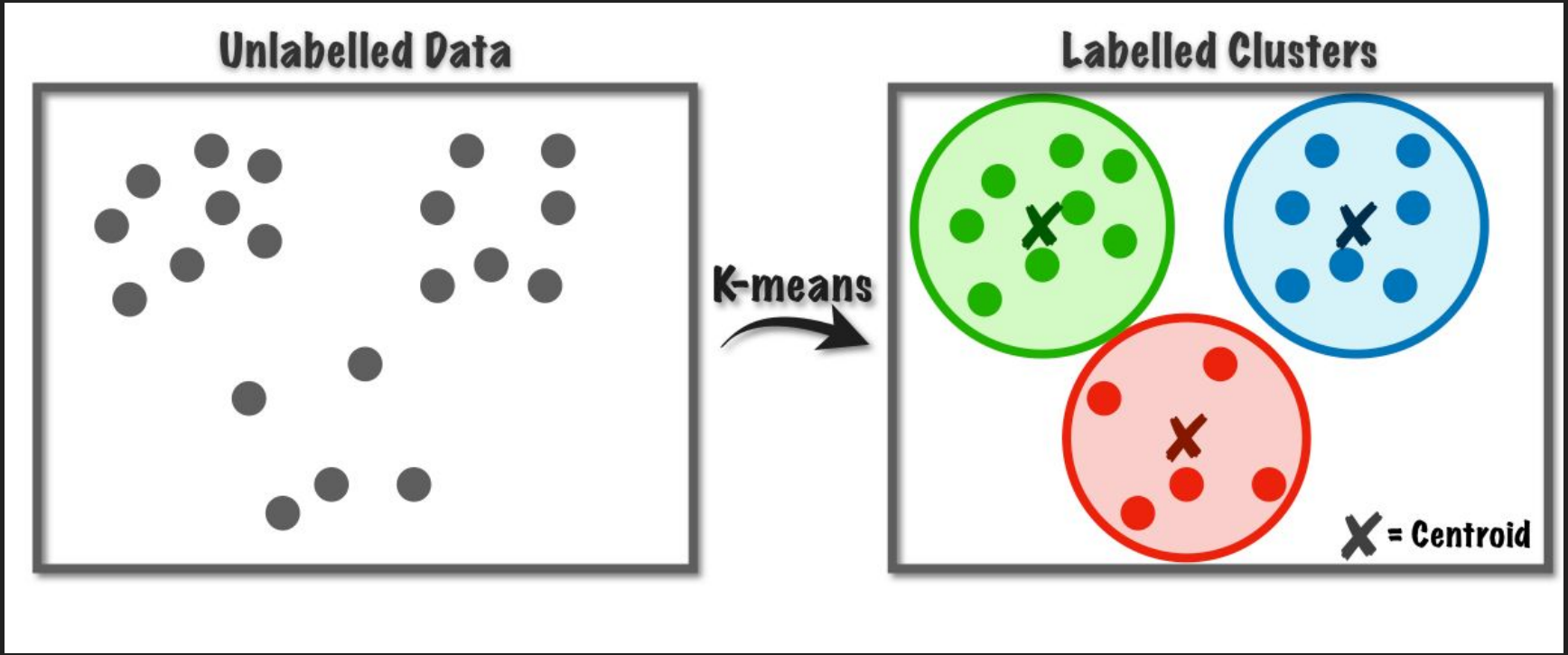
1 - Project Description

Introduction: FPGA **Accelerated Handwriting Digit Recognition**

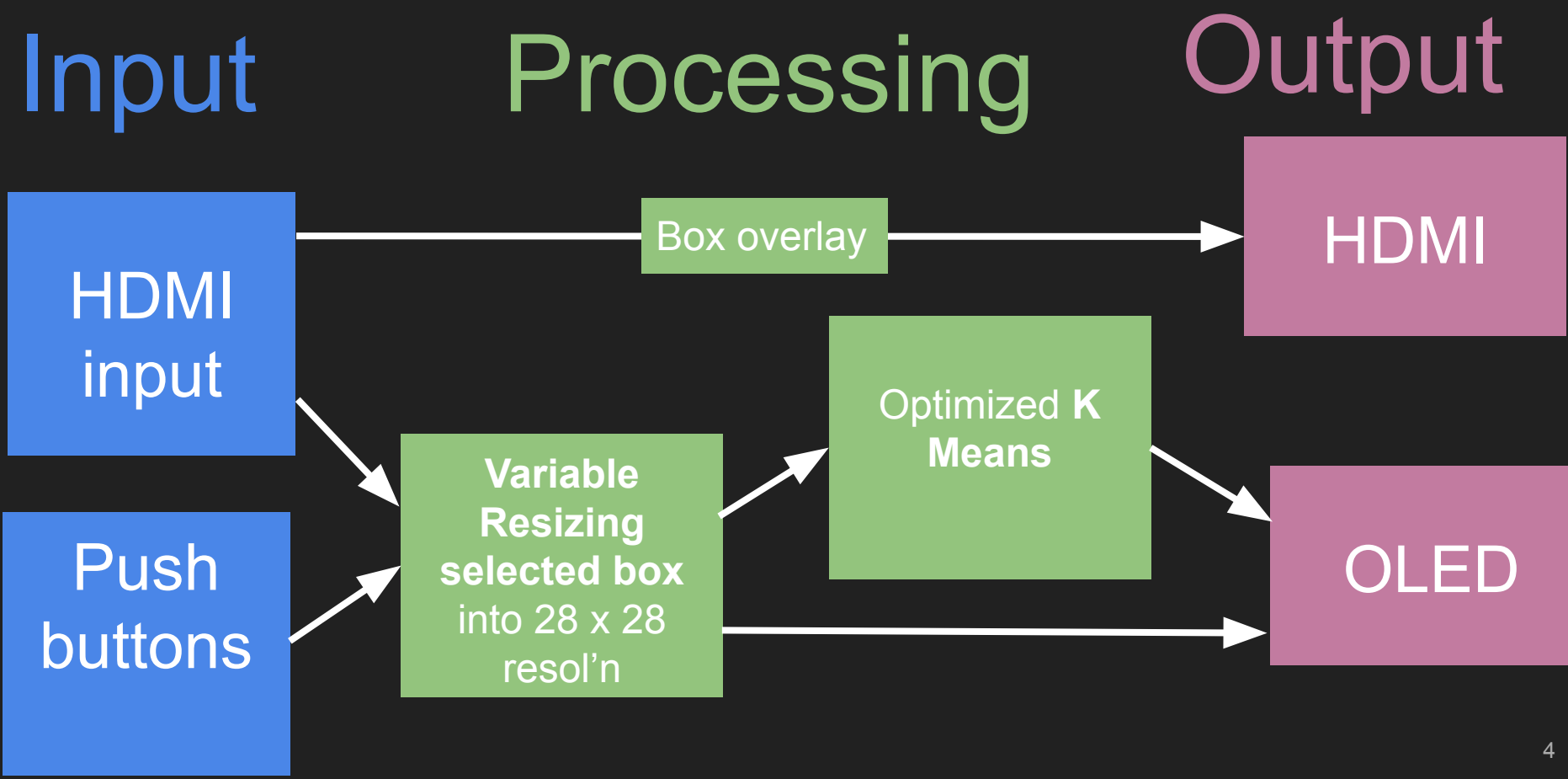


Adapted from [1]

Neural Classifier: KNN/K-Means

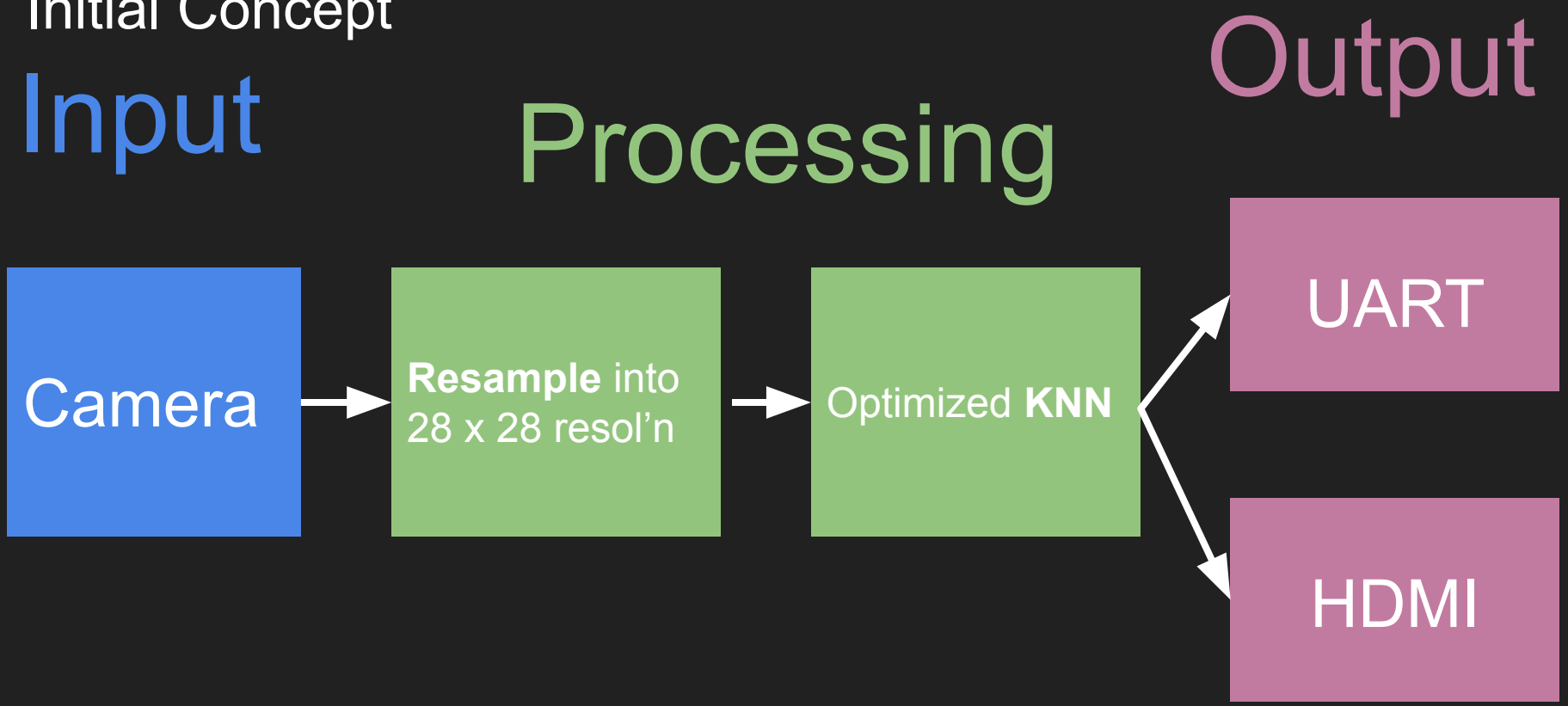


Adapted from [2]

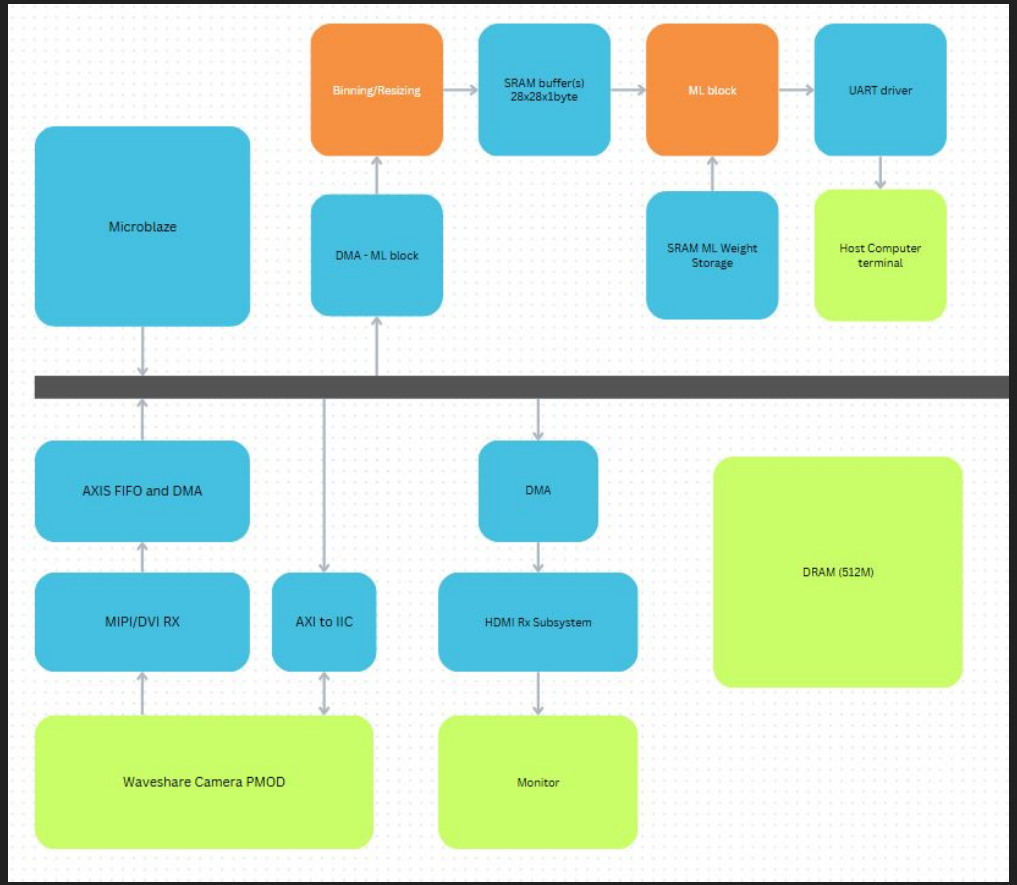

Input
Processing
Output
HDMI input
Push buttons
Box overlay
Variable Resizing selected box into 28 x 28 resol'n
Optimized K Means
HDMI
OLED

2 - Initial Goals

Initial Concept



Initial Top Level Block Diagram



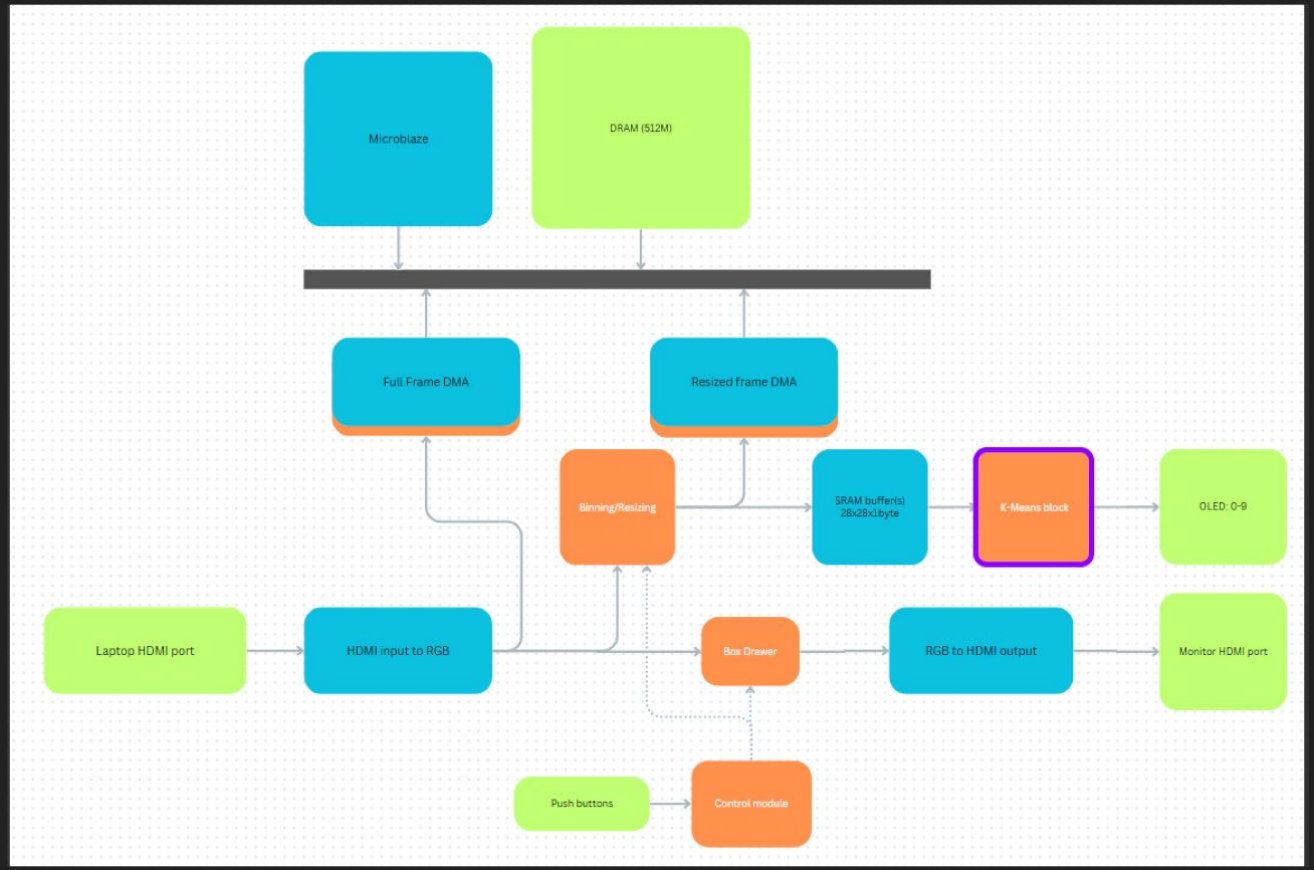
Main Changes

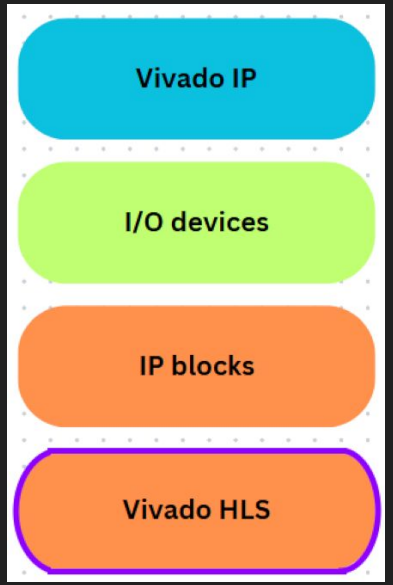
1. HDMI I/O not Camera
2. Variable input frame sizes
3. K-means using HLS
4. OLED to show results
5. Stream video instead of using

   DDR + AXI

3 - Final Project - high level block diagram

Refined Top Level Block Diagram





List of Custom IPs used

Packaged IPs

- K-Means classifier - fast + slow versions
- Resizing block
- Box controlling module
- Box drawer

Helper IP

- SW triggerable Video-2-MM-2-KMeans FSM
  - (controls interface between the K-Means + resizer)
- SW triggerable Video-2-AXIS (save 1x frame)
  - (controls interfaces with Sw)

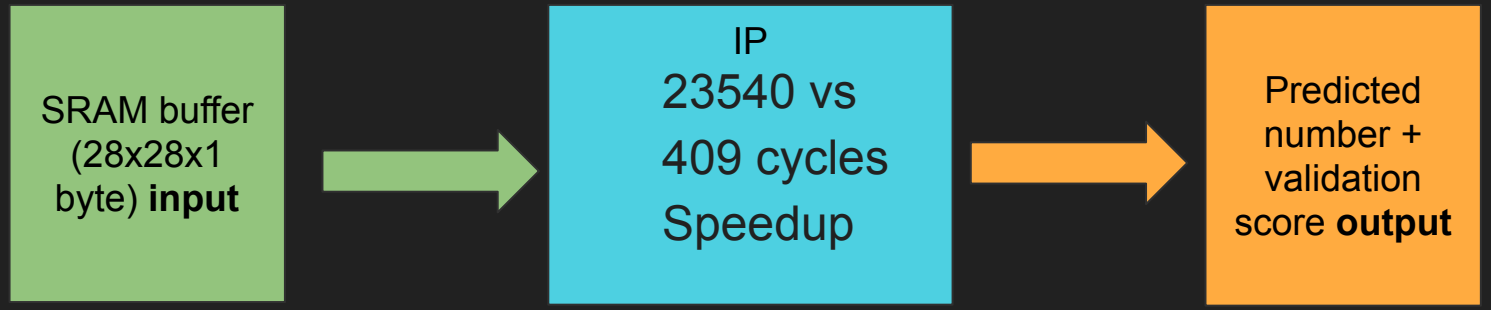
SW IP

- K-Means in SW
- Resizing in SW
- Event loop + FPS counter

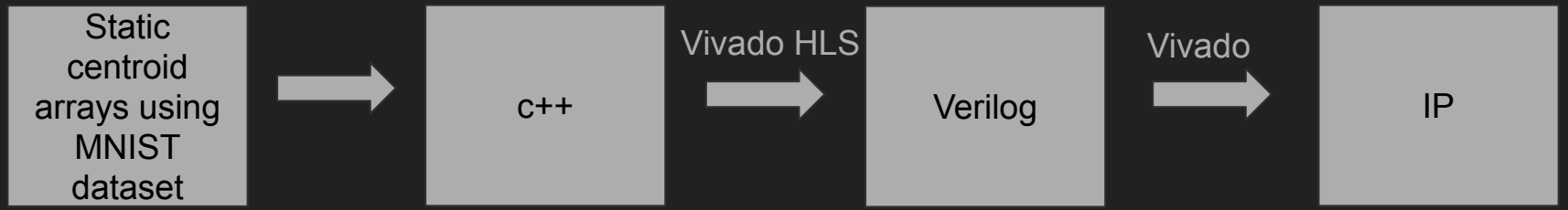
4 - Final design - Individual components

(1) Custom IP: K-Means

Input and output



IP creation



Utilization of K-means

Resource	Estimation	Available	Utilization %
LUT	23350	134600	17.35
LUTRAM	4	46200	0.01
FF	16459	269200	6.11
BRAM	784	365	214.79
DSP	740	740	100.00
IO	77	285	27.02
BUFG	1	32	3.13

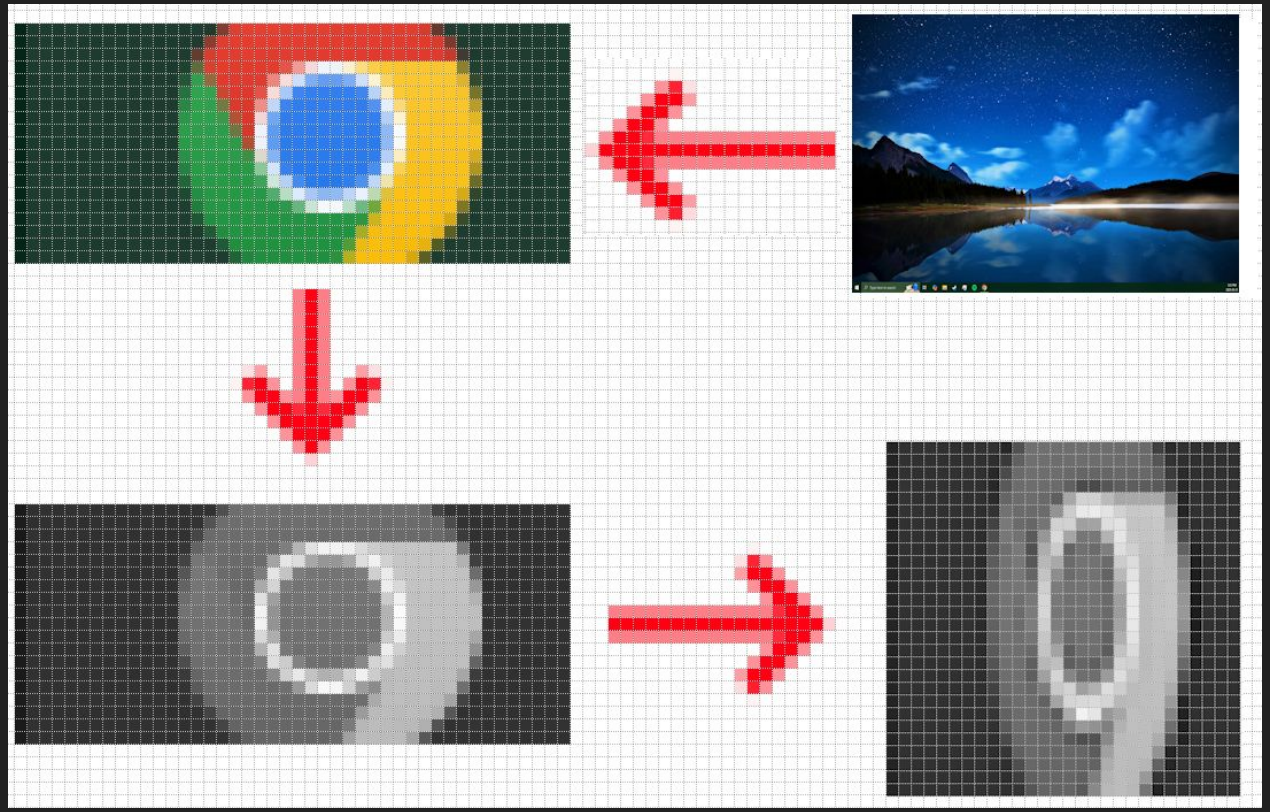
Neural Classifier Performance Comparison

Classifier type	Accuracy on MNIST	Storage requirement	Computational Requirement (per inference)
Full KNN	97.05%	45 MiB	47M MACs
Reduced KNN	81.74%	784 KiB	802K MACs
K means	82.05%	7.66 KiB	7.86K MACs

Atrix 7 BRam capacity: 13 MBits

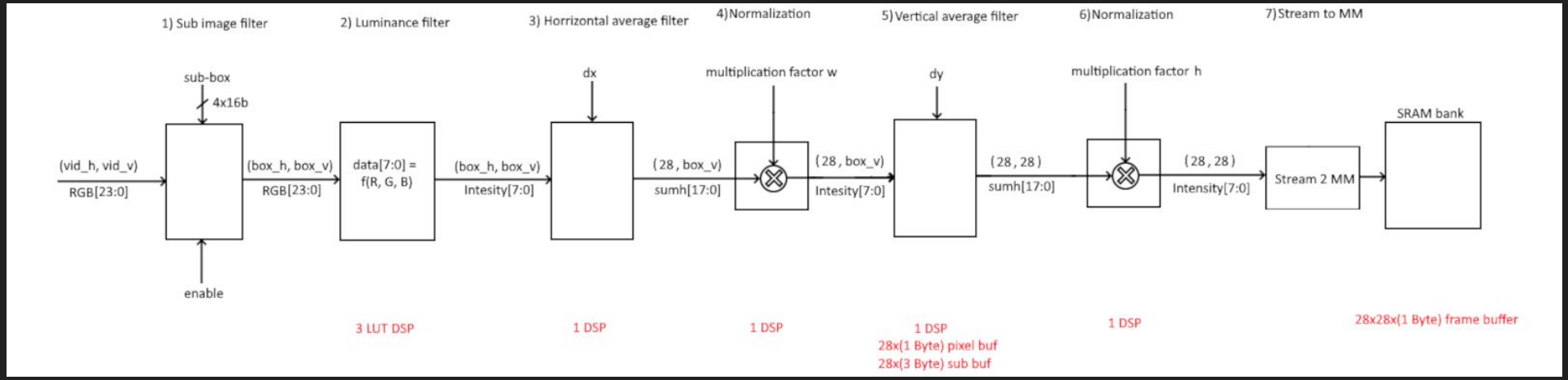
Atrix 7 DRAM capacity: 512MB

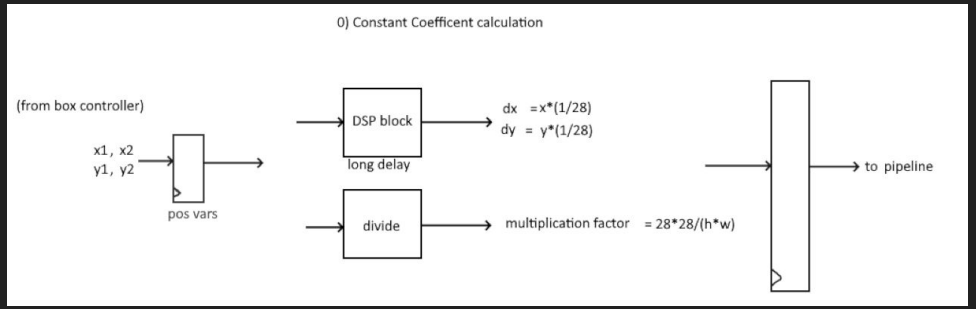
(2) Variable resizing



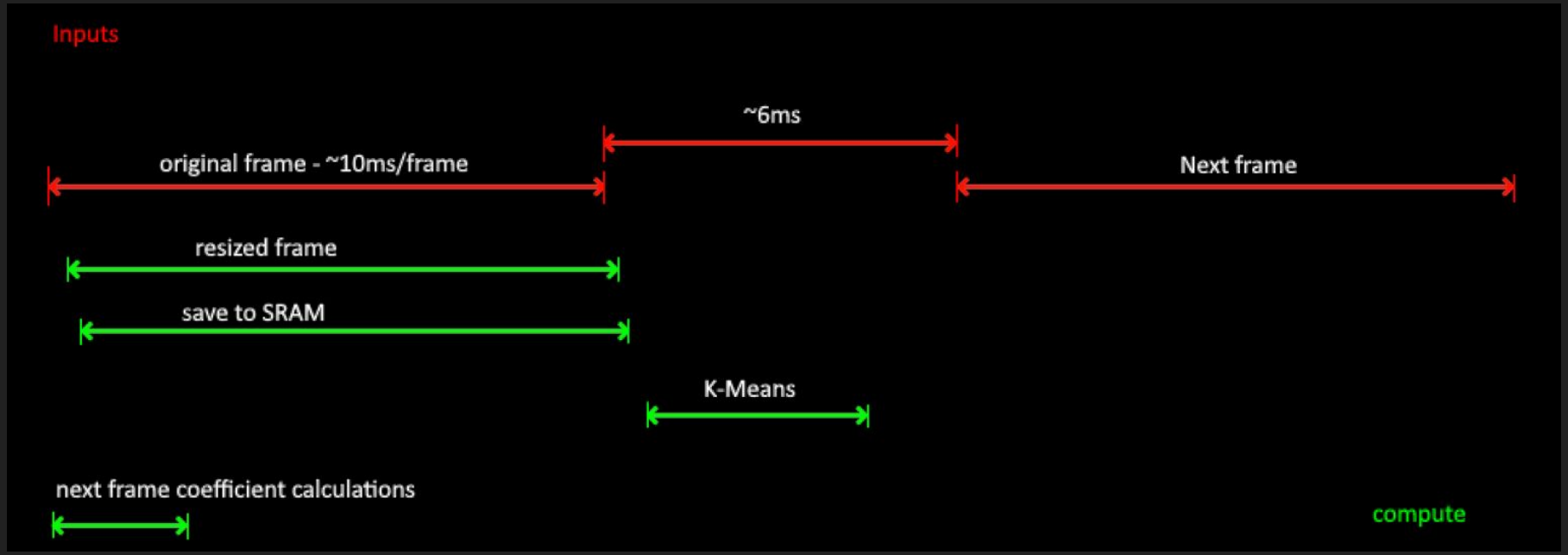
- Up sampling
- Down sampling
- Luminance
- Minimal HW resources and Latency

Resizing 7-Stage Pipeline





Timing diagram - Resizing IP

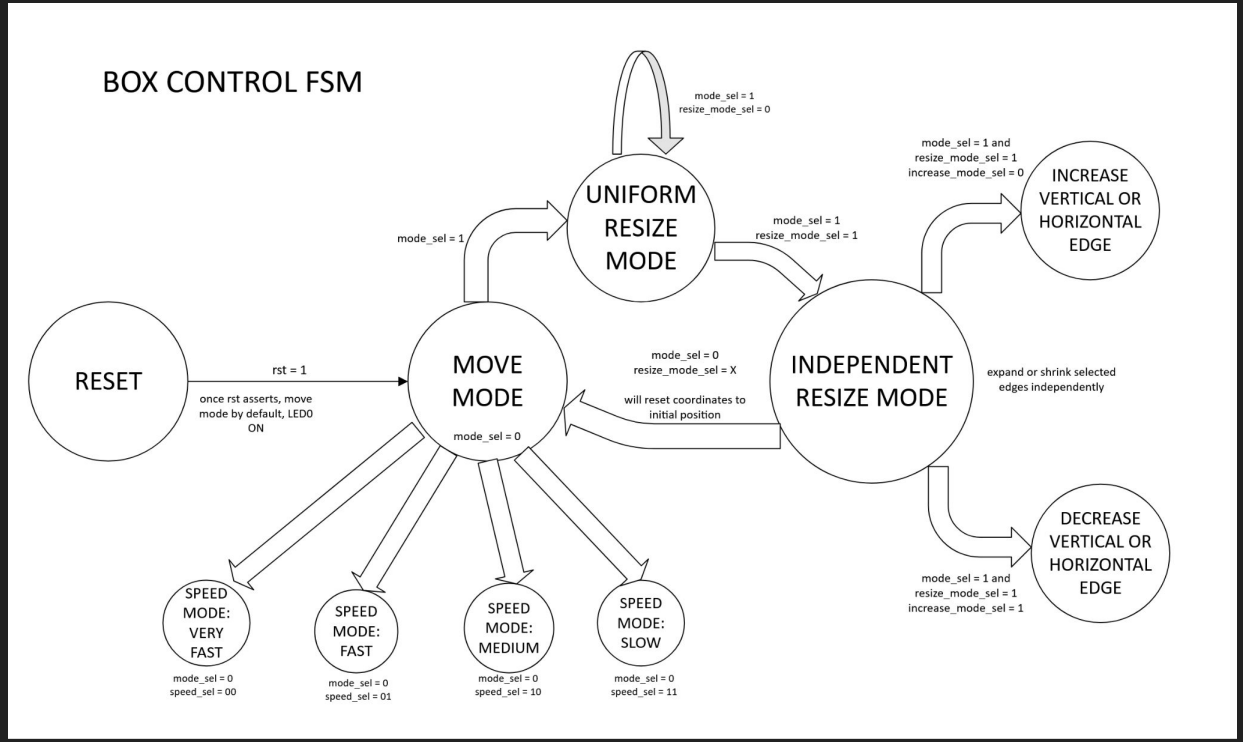


Utilization of Resizing IP

Resource	Utilization	Available	Utilization %
LUT	1153	133800	0.86
LUTRAM	48	46200	0.10
FF	1658	269200	0.62
DSP	4	740	0.54
IO	106	285	37.19
BUFG	1	32	3.13

(3) Box Control

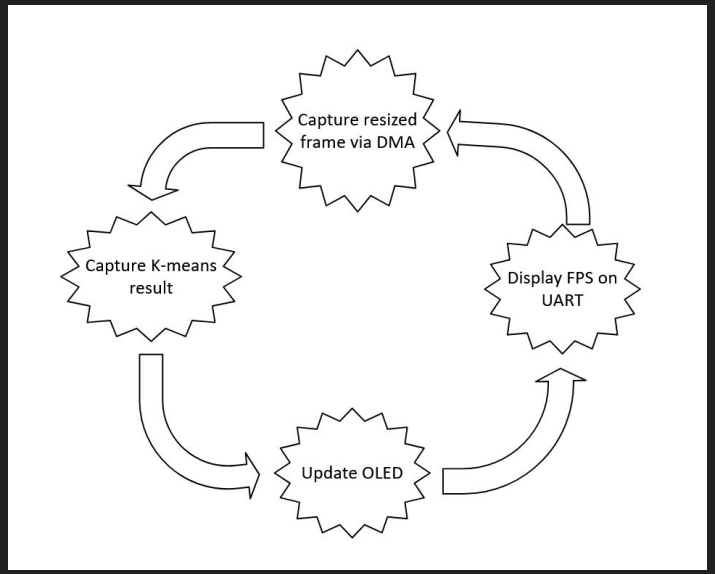
- Controls the movement and resizing of the bounding box of the image frame
- 2 modes:
  - Movement: a) fast b) slow
  - Resizing: a) uniform b) independent edge resizing
- Moderate complexity
  - Edge-sensitive (rising edge detection)
  - Time-sensitive (delay-based movement control) functionalities
- Speed selection mode can impact responsiveness and performance



Utilization of Box Control

Resource	Utilization	Available	Utilization %
LUT	232	63400	0.37
FF	95	126800	0.07
IO	78	210	37.14
BUFG	1	32	3.13

(4) Software

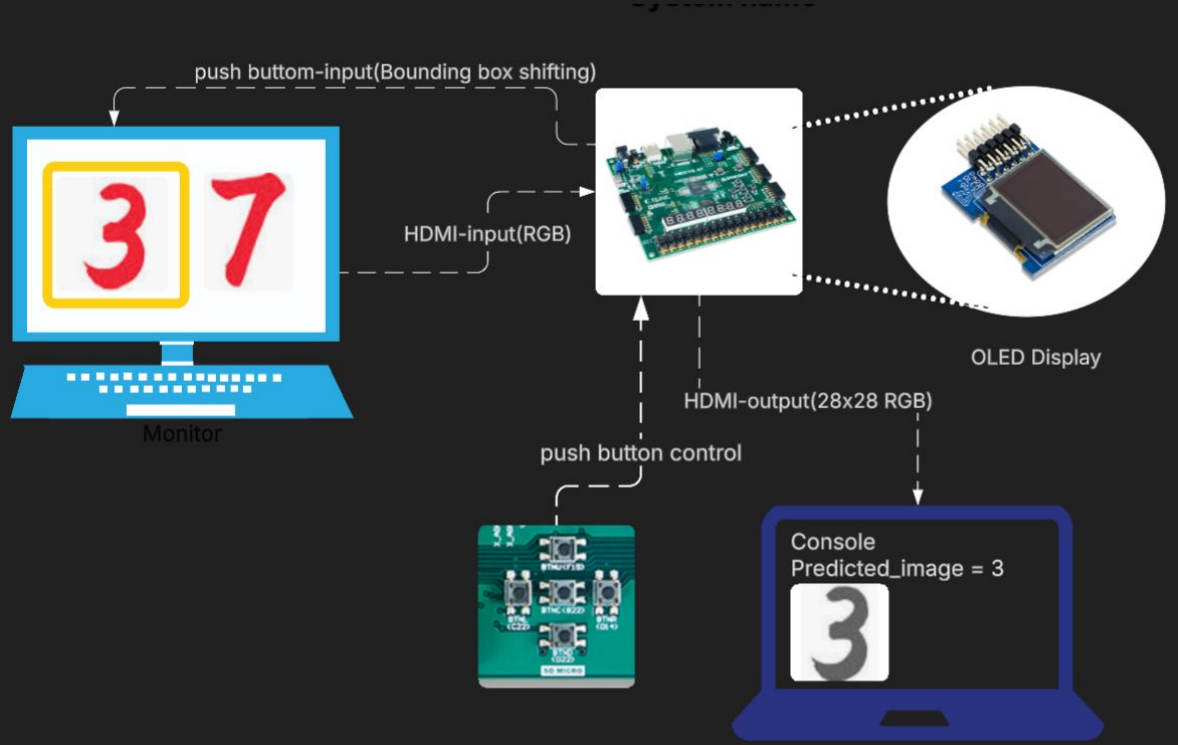


(additional)
- capture_full_frame()
- resize_image()
- infer_on_saved()

print_ascii_art()
print_saved_bitmap()

(5) Data flow + supporting blocks

- Resize/Binning Module
- HDMI to RGB Block
- RGB to HDMI Block



5 - Difficulty Score

Project Complexity Points

Binning/Resizing Module IP (HW + SW)	1.0
Knn-Means Classifier Packaged IP (HW + SW)	1.0
Box controller IP	0.5
OLED implementation	0.75
HDMI output	1.0
HDMI input	1.0
Implement performance monitoring & Meaningful visualization statistics (fps)	0.25
Total	~5.5

6 - How would we improve with more time?

Possible Improvements

1. More accurate classification
    - Use more centroids (i.e. >1) and parallelize distance calculations with centroids
    - Preprocessing images (e.g. deskew, center, remove artifacts)
    - Different distance metric
    - Feature extraction
    - (lenet)
2. Ping-pong buffering to increase throughput
3. Support for multi-character recognition and non-digit recognition

7 - The Demo

Demo Modes

1. SW resize, SW classification, and output to OLED
    a. 1.62 fps (only at low sized sub images)
2. HW resize, SW classification, and output to OLED
    a. 19.81 fps
3. HW resize, HW classification, and output to OLED
    a. 29.72 fps
4. HW resize, HW classification, only value pushed to OLED (60fps)
    a. Full 60fps

Results

Function	SW Latency (on microblaze)	HW Latency	Speedup
K-Means	15.7ms	235.4 us / 4.09 us (optimized)	99%
Update OLED with text / 28x28 image	11.9ms / 15.8ms	—	—
Resizing	10x10 input = 78.9ms 100x100 input = 1.86 s 600x600 input 56.26 s … 1280x720 = 2min 22 s	1.1 us (from after last digit input) / 16.6 ms	99%
Box control	(infinite/blocking)	(none/very small)	100%
Box drawing	8.3 ms	(none/very small)	100%

References

[1] H. Beniwal, "Handwritten digit recognition using machine learning," *Medium*, Apr. 5, 2020. [Online]. Available: https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64.

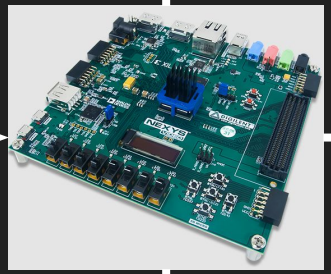
[2] S. Gupta, “Understanding how K-means Clustering Works (a detailed guide),” Medium, https://levelup.gitconnected.com/understanding-how-k-means-clustering-works-a-detailed-guide-9a2f8009a279.

Q&A Period

Demo Setup

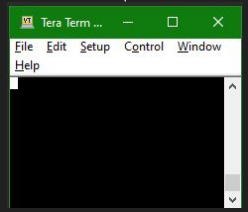
Results

HDMI



HDMI

FPS



ML Block - K-Means

- K means estimated at 235K Ops to keep up with 15 fps.

Confidence_N = |(IN - REF_N)|

(computed for each N = 0-9)