embedded VISION SUMMIT 2018

Utilizing Neural Networks to Validate Display Content in Mission Critical Systems

VeriSilicon

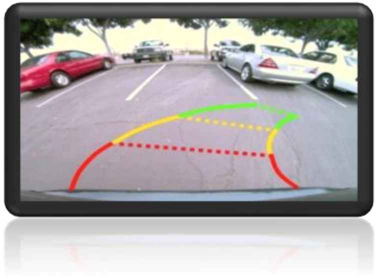
Shang-Hung Lin

May 23, 2018

The challenges of Mission Critical UI

What is Mission Critical UI?

- *Information that, if incorrect or missing, has the potential to imply risk to human life or material damage*.
- Automotive
  - Cluster
  - Mirror replacements
- Medical
  - Vital statistics
- Industrial
  - Machine status, hazards



Why go to all this trouble?

- Easy solution would be always to render all the UI in a safety certified environment; however:
  - The complexity of safety certification has an impact on SW cost – simpler is better
  - UI complexity is increasing, and safety certified renderers are trailing behind modern graphics APIs
  - UIs utilize a mix of mission critical & non-critical information
    - *The composition with bad data on a non-critical layer may cause the safety content to be unrecognizable!*

Example – Automotive Cluster

Navigation content from “unsafe” source.
Non-critical layer

Car velocity, dynamically scalable gauge with animated transitions.
Critical layer

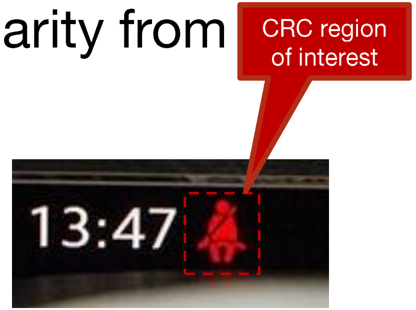
Car safety system status.
Critical layer

Content verification

- Display content must be verified to ensure that mission-critical information is *visible* and *recognizable* to humans
- Content must be verified in real time, so that safety fallback mechanisms are deployed immediately if content does not match expectation
  - Redundant UI renderer
  - System failure messages
  - System recovery
- Safety check mechanism must be isolated from primary UI system, in safety certified environment
- Governed by strict standards (ISO-26262, DO-178C)

Current solutions

- Cyclic Redundancy Checks (CRC) of Regions of Interest (ROIs) in display controller
  - Fixed-function hardware = simple, low risk
  - Alpha masking of opaque pixels allows for some level of composition with dynamic background
    - Does not prevent accidental camouflage by similarity from adjacent pixels
  - Requires pixel-perfect match per frame
    - Limits blending, dynamic colors and animation

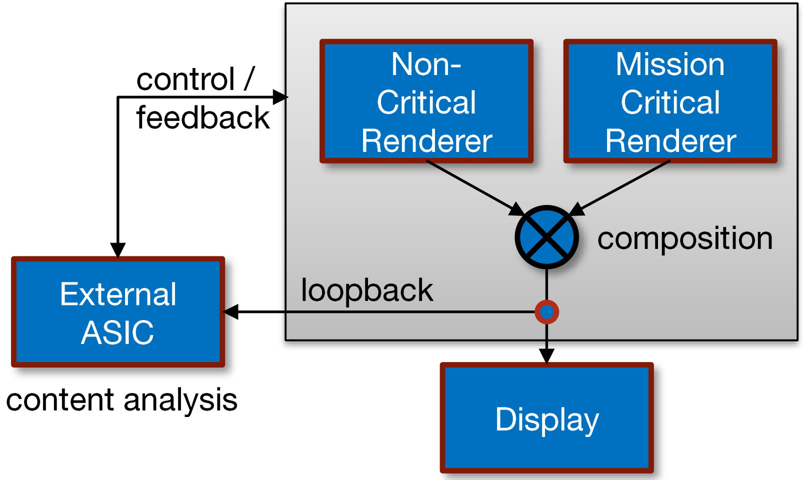


Evolving requirements

- Mission Critical UI requirements are evolving
  - Animations
  - Blending
  - Varying font parameters per language
  - User theme colors
- Pixel-perfect check by CRC is unsustainable
- Content *recognition* is needed instead

Latest attempts for meeting UI demands

- External ASIC or isolated CPU core in SoC for content analysis, fed by display signal
- Drawbacks:
    - Solution cost, complexity
    - Latency of configuration & feedback from external analysis
    - Power consumption



Solution: Embedding the content recognition in the Display Pipeline

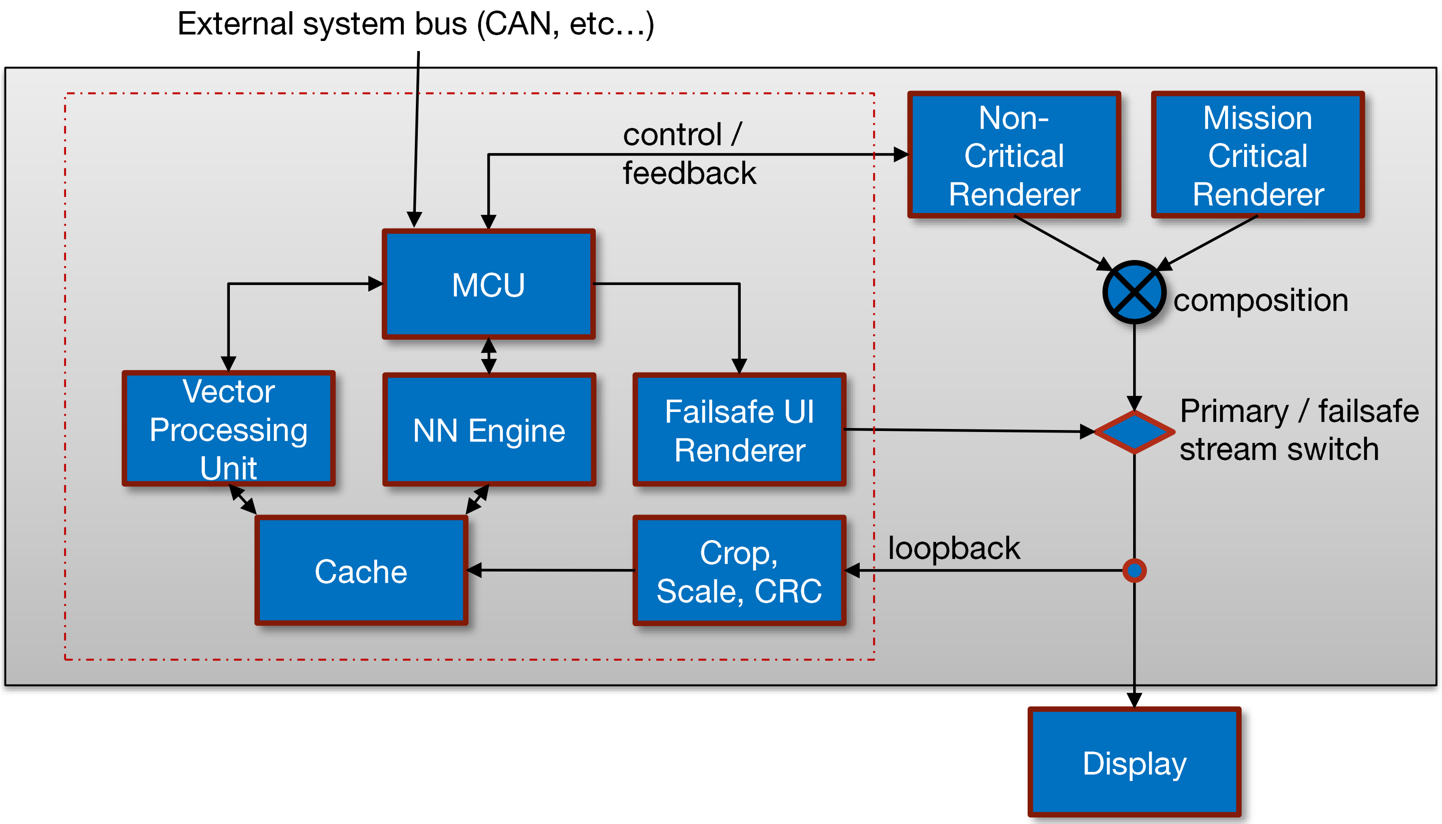
VeriSilicon IP Portfolio

- *Neural Network AI*
- *Video*
- *GPU*
- *Audio/Voice*
- *Compute*
- *ISP*
- *Display*
- *Compression/ Encryption*

Embedding a DNN accelerator in the display pipe

- Dedicated HW for content recognition in the display pipeline
- Internal loopback to basic capture interface
  - Crop and scale ROIs, CRC
- Programmable vector engine for candidate search
- DNN engine for recognition of mission-critical data
- Microcontroller for coordination and high-level logic

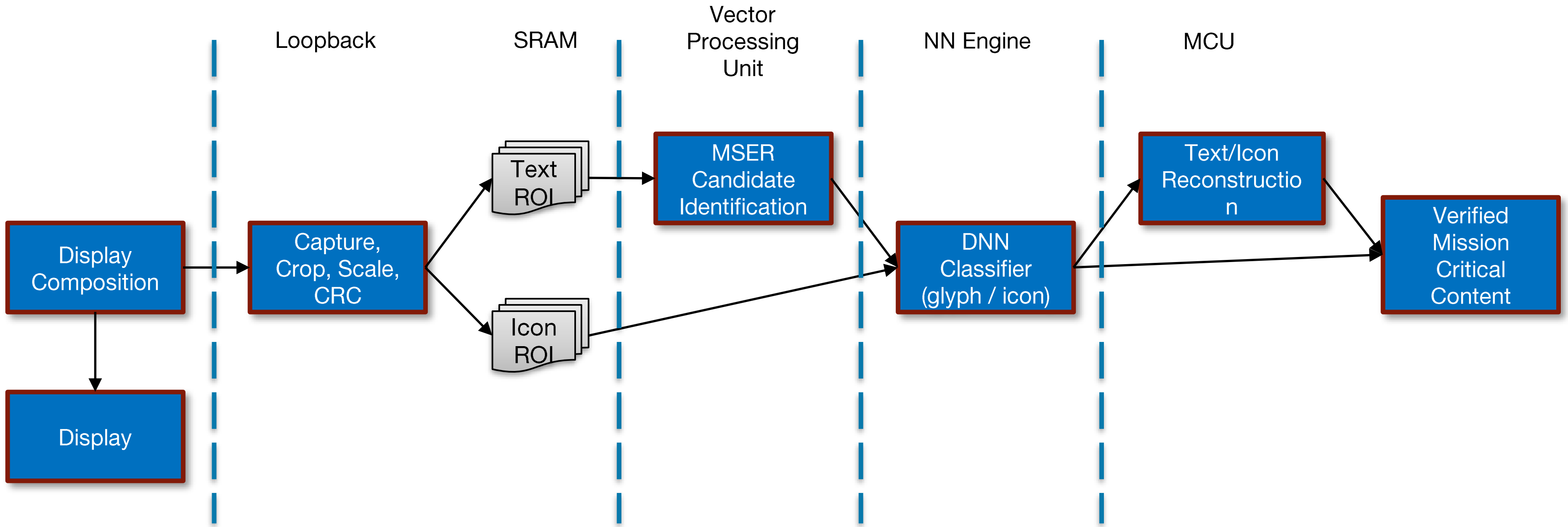
Embedding a DNN accelerator in the display pipe



Benefits

- Self-contained, safety certifiable display subsystem
- Content verification and failsafe UI renderer run within the same safety critical environment
- Minimized system cost impact for safety solution

Embedding a DNN accelerator in the display pipe



MSER candidate search

- MSER: **M**aximally **S**table **E**xtremal **R**egion
  - Blob detection method suitable for detecting character features

- MSER can be parallelized on the programmable vector engine, by slicing of the text ROI into tiles, running the algorithm separately on each tile and coalescing adjacent objects in connected tiles.

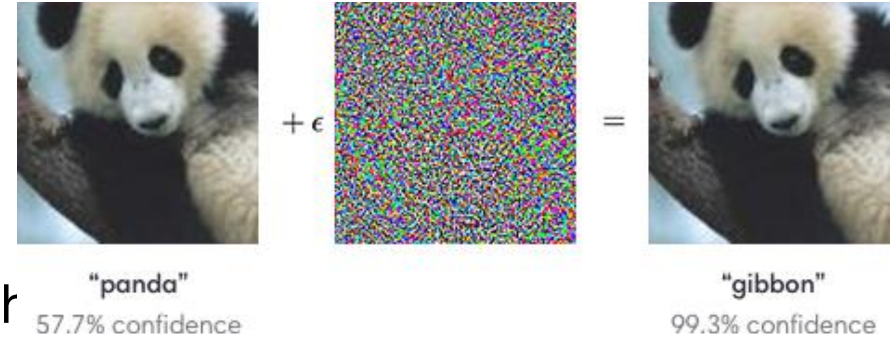
- With the DNN classifier running on the NN engine, the vector engine can be fully dedicated to the candidate search.

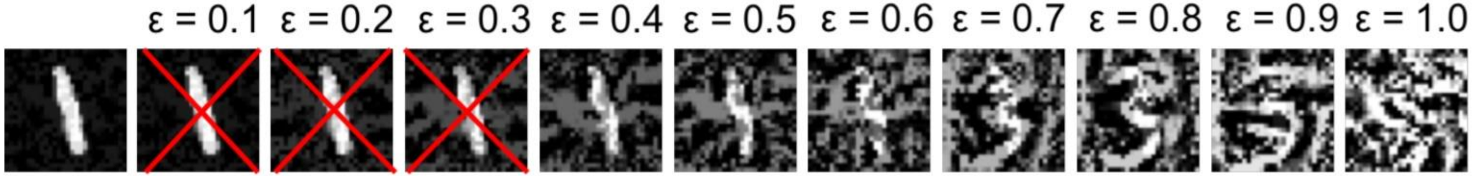
Workloads

- Known system fonts - limited training set for text
- Icons are typically monochromatic, single scale & orientation
- Augmentation of training by varying glyph background and blend levels
  - Robustness needs to be kept moderate – *must reject hard-to-see cases!*
  - Blending against noisy backgrounds should be used for training of negative classifier

Adversarial Training

- Adversarial examples: slight modification intended to fool machine learning models
- Adversarial training helps DNN classifiers to be better aligned with human perception





Adversarial examples crafted by PGD. $\varepsilon > 0.3$ are "hard-to-see" cases

Simplifying the DNN workload

- Single DNN classifier fits both text and icons, creating a font super-set
- Simplicity of features allows for network quantization and pruning with tolerable accuracy loss, targeting permanent location in SRAM
- ROIs predefined by UI layout
    - Simplifies candidate search to localized MSER for text regions
        - MSER luminance levels can be limited to known font intensities
    - Direct inference on icon ROIs

Simplifying the DNN workload

- Content verification experiment with MNIST database

LeNet-5	**FP32 No pruning**	**INT8 No pruning**	**INT8 85% pruning**	**INT4 85% pruning**
Accuracy	96.1%	95.6%	95.6%	92.7%
Model Size	1.6 MB	400 KB	73 KB	36 KB

CapsNet	**FP32 No pruning**	**INT8 No pruning**	**INT8 50% pruning**	**INT4 50% pruning**
Accuracy	98.5%	98.3%	98.3%	92.5%
Model Size	31 MB	7.8 MB	3.9 MB	1.8 MB

Concluding remarks

- Validating the display content in mission-critical systems has become a challenging task due to the advance of graphic user interface technology and modern graphics API

- Neural network approach provides adaptivity and robustness for content verification

- A self-contained, safety certifiable content verification system is proposed. The system cost impact for safety solution is minimized by embedding a neural network accelerator inside the display pipeline

Resources

- Mission-critical UI (ISO 2575): https://www.iso.org/standard/54513.html
- MSER parallelized acceleration: http://csl.cs.ucf.edu/courses/CDA6938/projects/mser.ppt
- LeNet5: http://yann.lecun.com/exdb/lenet/
- CapsNet: https://arxiv.org/pdf/1710.09829.pdf
- VeriSilicon Display Controller IPs: http://www.verisilicon.com/IPPortfolio_2_119_1_DisplayControllerIP.html
- VeriSilicon DNN AI Processor IPs: http://www.verisilicon.com/IPPortfolio_2_122_1_VisionIP.html
- Adversarial examples: https://arxiv.org/pdf/1710.10733.pdf
- Adversarial training: https://arxiv.org/pdf/1706.06083.pdf