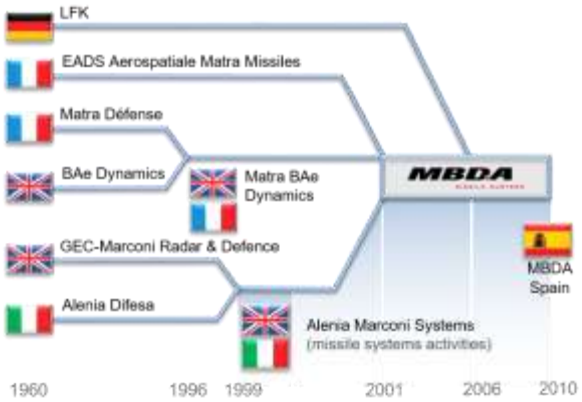
The challenges of AI and multi-camera imaging: An MBDA perspective

30th November 2022 – UDRC Themed Workshop in *Algorithm Implementation and Low SWAP Challenges*

Nikki Easton and Dean Goff

MBDA
MISSILE SYSTEMS

- **Created in 2001 from a series of mergers**
- **The largest European company in the missile systems sector**
- **The only European company able to meet the whole range of complex weapons needs of the three armed forces**
- **More than 12,000 people worldwide**
  - 60% in Technical/Engineering functions

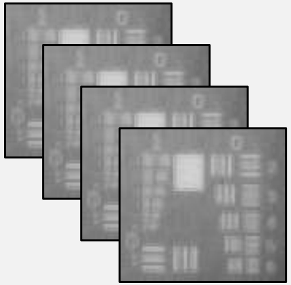


Correct as of March 2021

Challenges of AI Industrialisation

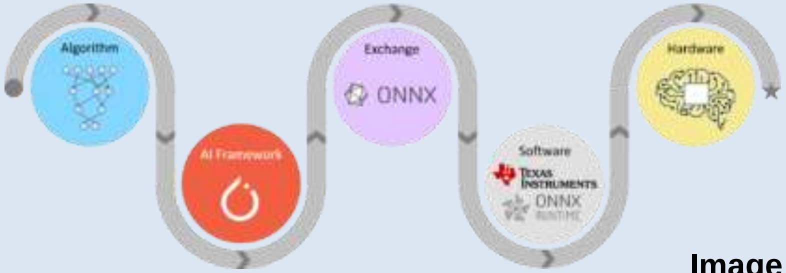
Nikki Easton
Image Processing Department

Challenges of Multi-Camera Imaging



Dean Goff
Electro-Optics Department

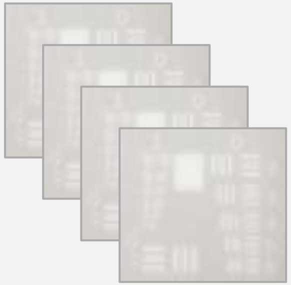
Challenges of AI Industrialisation



Nikki Easton

Image Processing Department

Challenges of Multi-Camera Imaging



Dean Goff

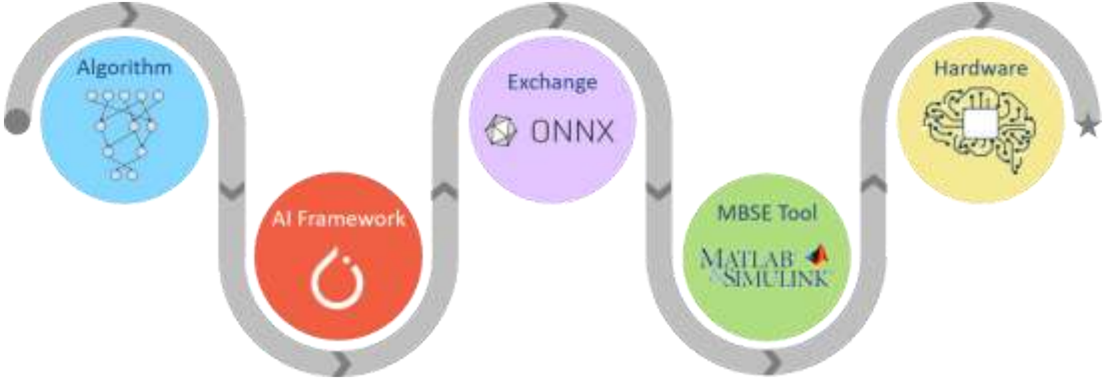
Electro-Optics Department

AI gives a huge leap forward in computer vision performance on a range of tasks

Challenge: Exploitation on the edge

A major obstacle in increasing Technology Readiness Levels of AI

Work in 2021 – Real time ATR bench demonstrator



Computing challenge of AI

- Huge number of matrix operations
- Unsuitable for regular CPUs

Embedded AI processing platform examples

- NVIDIA® Jetson™ – not suitable
- Xilinx Versal®
- NXP i.MX 8M Plus ®
- Texas Instruments Jacinto™ 7

TEXAS INSTRUMENTS

Jacinto™ 7
SK-TDA4VM
TDA4VM processor starter kit for Edge AI vision systems

8 TOPS MMA
2x ARM® Cortex®-A72

Challenges of TI Jacinto Development Workflow

AI framework

PyTorch

Develop and train bespoke algorithm

Exchange

ONNX

Transfer trained model between tools

MBSE Tool

MATLAB & SIMULINK

Integrate into full system models

Embodiment

ONNX RUNTIME

Texas Instruments

TIDL-RT Edge-AI-Tools

Deployment

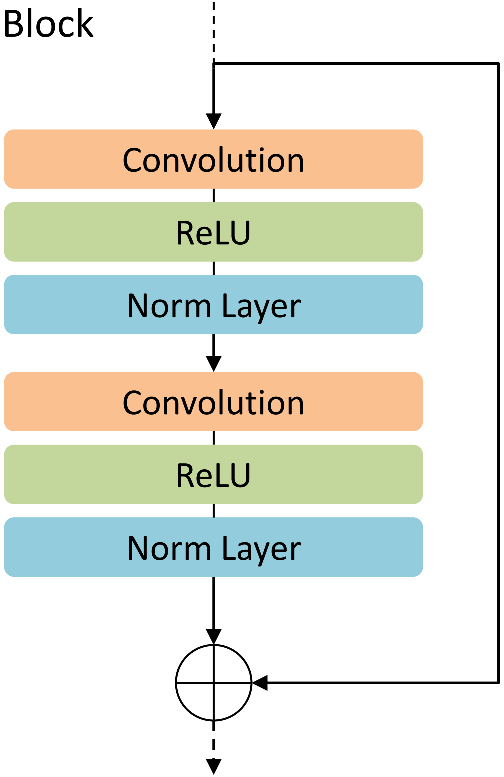
Texas Instruments

Jacinto™ 7

SK-TDA4VM

ResNet Basic Block

(example)



Embody a custom algorithm through the workflow

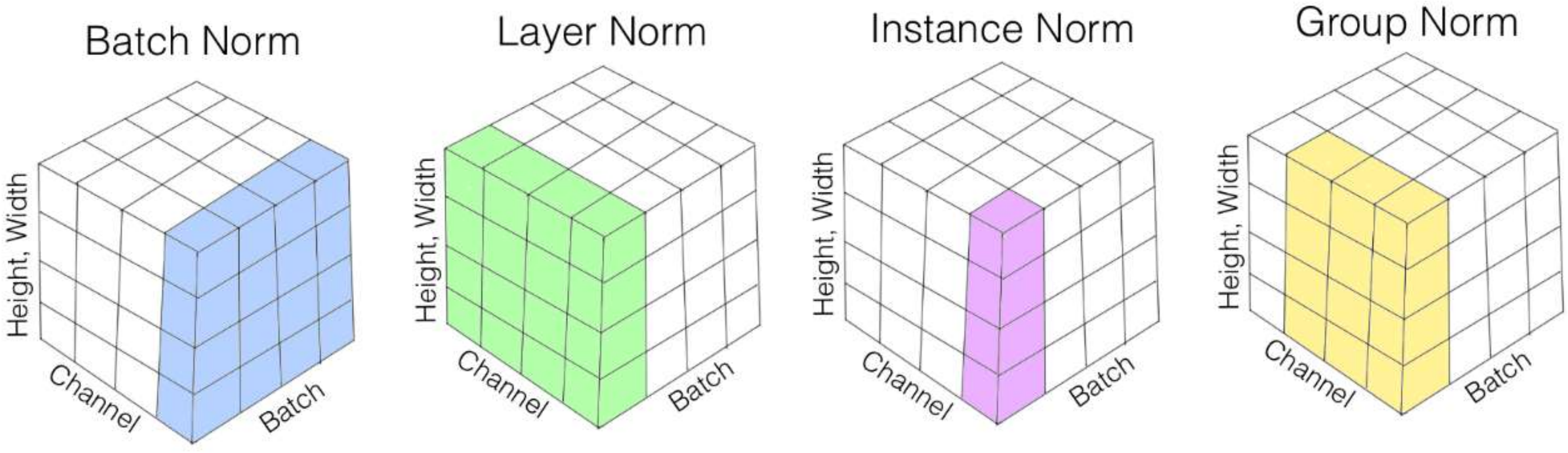
Classification problem using a ResNet-18

This mostly contains standard layers

Adapted architecture with a different norm layer

A normalisation layer just normalises the input, with regards to the chosen dimensions

$$\widetilde{x_{\blacksquare}} = \frac{x - \mu_{\blacksquare}}{\sqrt{\sigma_{\blacksquare}^{2} + \epsilon}}, \qquad where\ \blacksquare = B, L, I\ or\ G$$



Convolution Layers

Pooling Layers

- Global Pooling (Average, Max)
- Spatial Pooling (Average, Max)

Activation Functions

- ReLU Layer
- PReLU Layer
- ReLU6 Layer
- Sigmoid Layer
- Leaky ReLU Layer

Output Layers

- Fully-connected Layer
- Soft Max Layer

Norm Layers

- Batch Norm Layer (inference mode only)
- Instance Norm Layer
- Layer Norm, Group Norm Layers

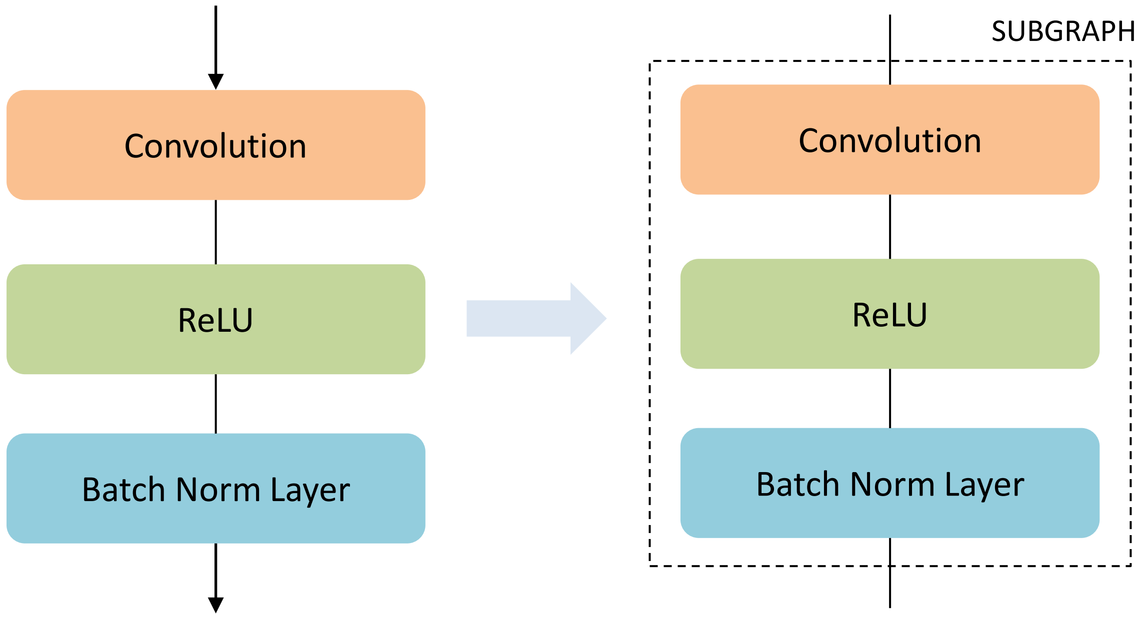
General Layers

- Element-wise Layers (add, subtract, multiplication)
- Inner Product Layers
- Bias Layer
- Concatenate Layer
- Scale Layer
- Re-size Layer
- Arg Max Layer
- Slice Layer
- Crop Layer
- Flatten Layer
- Shuffle-Channel Layer
- … and 5 more general layers

Many more layer types and custom layers

Red = unsupported

The above is correct as of TIDL-RT version 08_01_00_05: https://software-dl.ti.com/jacinto7/esd/processor-sdk-rtos-jacinto7/08_01_00_13/exports/docs/tidl_j7_08_01_00_05/ti_dl/docs/user_guide_html/md_tidl_layers_info.html



SUPPORTED BY TIDL-RT
Accelerated on MMA

Results

ResNet-18

Batch norm in inference mode only

1 Subgraph

588 Hz

How does TIDL-RT work?
Instance Norm Layers

Convolution

ReLU

Instance Norm Layer

Convolution

ReLU

SUPPORTED BY TIDL-RT
Could be accelerated on MMA
So executed on CPU

Instance Norm Layer

NOT SUPPORTED BY TIDL-RT
Executed on CPU

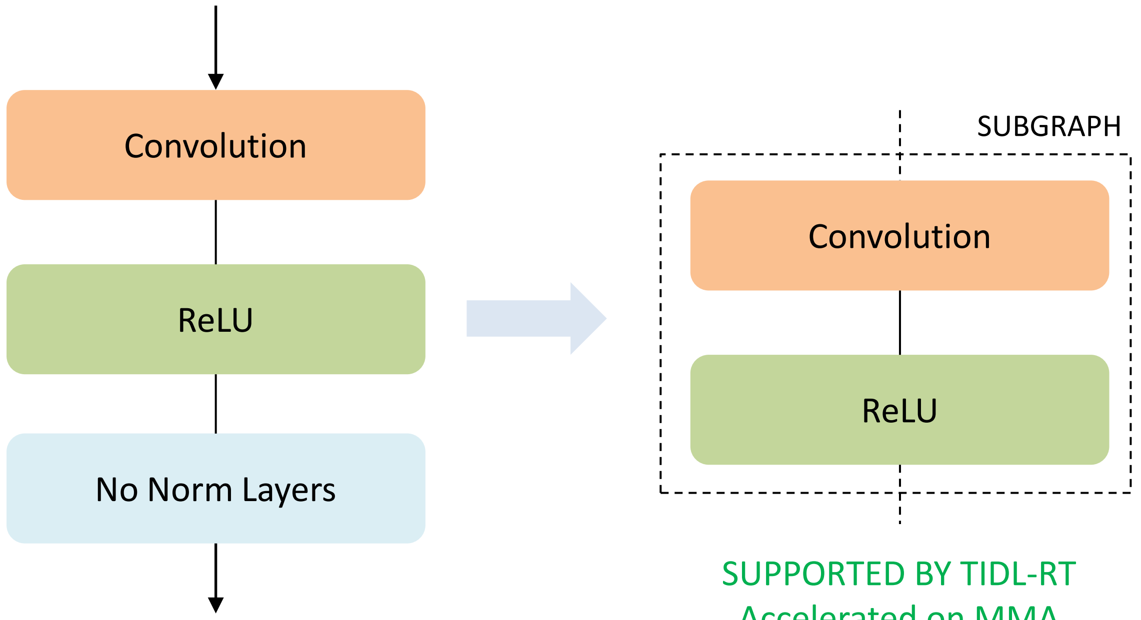
Results

ResNet-18

Instance Norm Layers

47 Subgraphs
Causes CPU execution only

~1s on CPU only



SUPPORTED BY TIDL-RT
Accelerated on MMA

HOWEVER: Do not want to change algorithms to suit the hardware

Results

ResNet-18

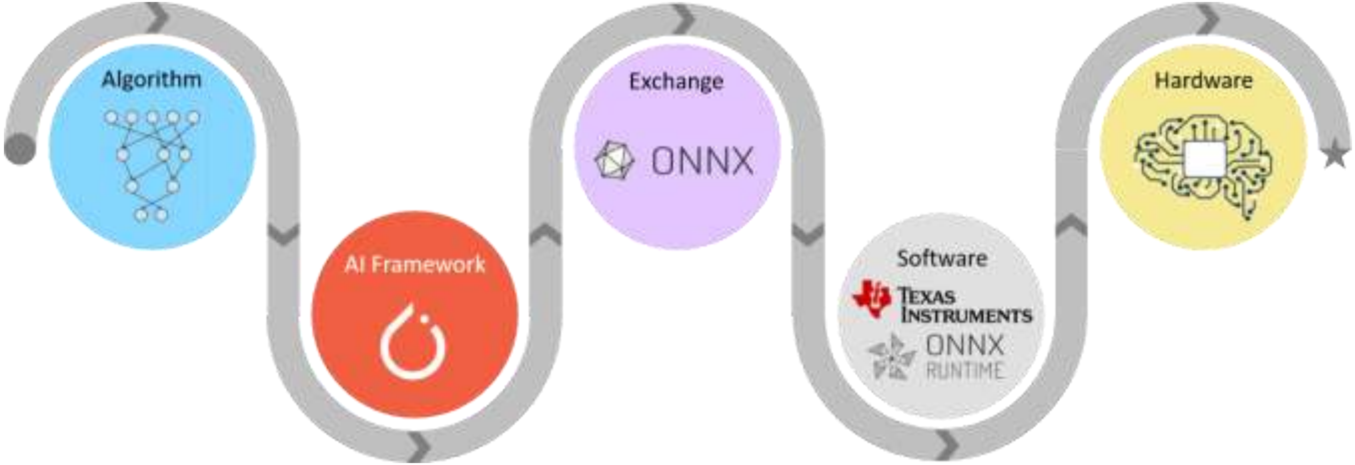
No Norm Layers

Gradient clipping and weight normalisation at training time only

1 Subgraph

572 Hz

Overview of challenges



Accelerating custom AI architectures is difficult

Exploitation onto AI silicon requires closer interaction between algorithm and software engineers than embedding solely onto CPU

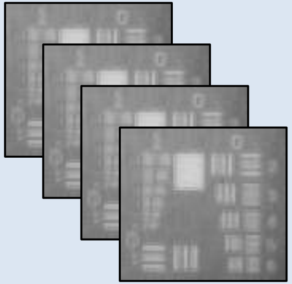
Need to execute custom layers on any dedicated AI silicon

This is only experience with the TI Jacinto, are other suitable options the same?

Challenges of AI Industrialisation

Nikki Easton
Image Processing Department

Challenges of Multi-Camera Imaging



Dean Goff
Electro-Optics Department

- **Multiple cameras/sensors are now ubiquitous in smartphones**
- **In smartphones, they have several use cases:**
  - Optical zoom can be achieved by employing multiple fixed focal length cameras – maintaining a compact form factor and without moving parts
    - With the addition of a Time of Flight sensor, depth information can be used to improve the image fusion process, as well as introduce synthetic bokeh
  - The sharpness and contrast of colour images can be improved by fusing colour and monochrome images together – monochrome sensors are more sensitive
- **From MBDA's perspective, multiple sensors could have several use cases:**
  - Improving operator situational awareness by fusing infrared (IR) and visible images
  - Improve target detection and defeat counter measures by sensing in multiple wavebands
  - Maximise design flexibility by employing multiple compact sensors, with the potential to:
    - Remove gimbals
    - Free up space for other sensors, e.g. RF
    - Improve signal-to-noise ratio and angular resolution using frame averaging and super resolution

- **To investigate these techniques MBDA has created a multi-camera rig**
  - 4x Teledyne FLIR Boson® LWIR cameras (640x512 resolution)
  - 1x Basler ace 2 monochrome visible band camera (1920x1200 resolution)
  - HP Z4 workstation (NVIDIA® Quadro® P4000 8GB GPU)

- **Configurations**
  - Visible & IR fusion (1x IR plus monochrome)
  - Multi-camera IR computational super resolution (2x, 3x or 4x IR cameras)

- **Algorithms**
  - Two-scale image fusion (TIF)

Ref 1 D. P. Bavirisetti, D. Prasad and R. Dhuli, "Two-scale image fusion of visible and infrared images using saliency detection.," *Infrared Physics & Technology* , vol. 76, pp. 52-64, 2016

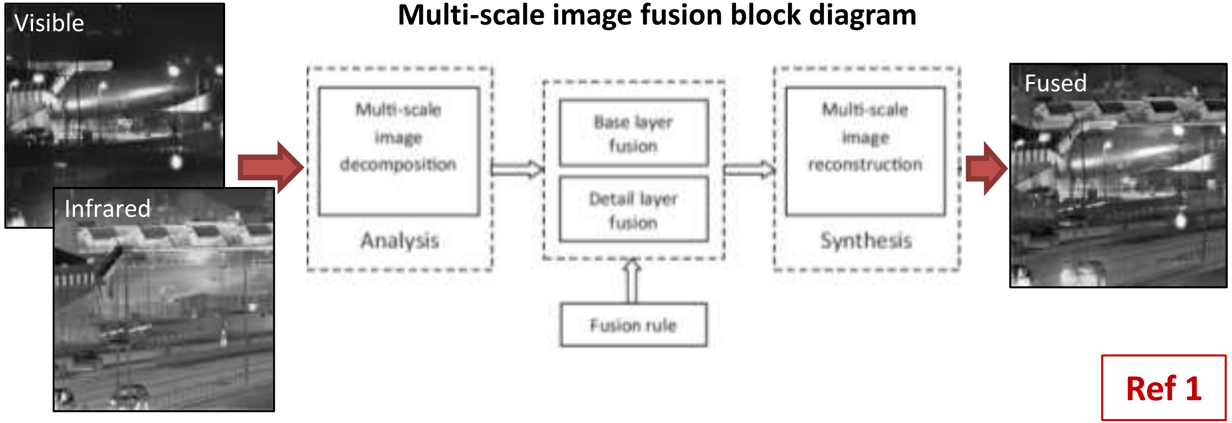
  - Super Resolution (SR) achieved by Maximum Likelihood Estimation (MLE)

Ref 2 G. Carles, J. Downing and A. R. Harvey, "Super-resolution imaging using a camera array," *Optics letters,* vol. 39, no. 7, pp. 1889-1892, 2014

- **Implementation (MATLAB)**
  - Multi-threading and GPU acceleration used to increase runtime performance:
    - Separate threads for TIF and MLE algorithms
    - Both MLE and TIF algorithms utilise the GPU
  - Runtime performance: TIF and SR algorithms running together at ~10Hz with GPU acceleration
    - Unsurprisingly, decreasing the number of cameras involved in SR improves runtime performance

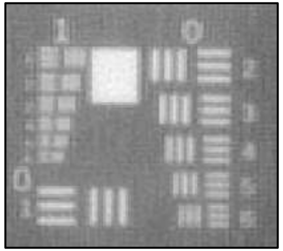
- **TIF Algorithm** Ref 1
  - TIF is a fast and efficient method of fusing images based on saliency detection and two-scale image decomposition

Multi-scale image fusion block diagram



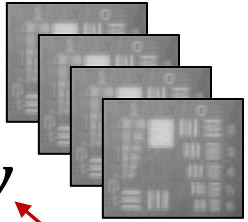
Ref 1

- **MLE Super Resolution (Carles *et al.*)** Ref 2
  - The MLE SR algorithm is a homography (or projection) based technique that formulates the SR problem as a classic restoration model: $\boldsymbol{y} = \boldsymbol{M}\boldsymbol{x} + \boldsymbol{e}$
    - Where $\boldsymbol{y}$ is an ordered vector of acquired image data, $\boldsymbol{M}$ represents a combination of a decimation matrix and warping matrix and $\boldsymbol{x}$ is the high resolution image to be estimated
  - MLE attempts to estimate $\boldsymbol{x}$ iteratively using Bayesian estimation without any prior:



$\boldsymbol{x} = 1280 \times 1024\ vector$

$$\boldsymbol{x}_{n+1} = diag(\boldsymbol{x}_n)\boldsymbol{M}^T\left(diag(\boldsymbol{M}\boldsymbol{x}_n)\right)^{-1}\boldsymbol{y}$$



$\boldsymbol{y} = 640 \times 512 \times 4\ vector$

$\boldsymbol{M} = 1310720 \times 1310720\ sparse\ matrix^* \ (nz = 8180541)$

* MATLAB® only supports double precision sparse matrices

- **Computational Super Resolution is our main focus**
  - This allows us to design a system with both a wide field of view and a high angular resolution

- **Two main challenges**
  - Increase the frame rate – the cameras' maximum frame rate is 60Hz (initial aim)
  - Port the algorithm to representative hardware

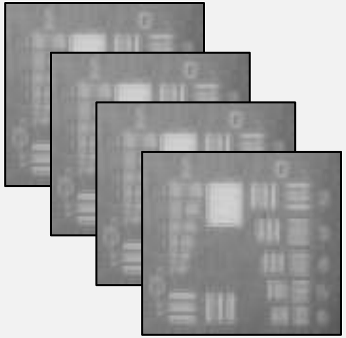
- **Potential hardware: FPGA or TI Jacinto™** (or something else?)
  - Given the large memory requirements, an FPGA doesn't feel right (I'm happy to be corrected)

- **Other options to increase runtime and decrease memory footprint**
  - Modify existing algorithm – only super resolve a region of interest
  - Adopt a different algorithm
  - Develop a Machine Learning (ML) solution – stereo SR networks already exist

Thank you for listening

Challenges of AI Industrialisation

Nikki Easton – Image Processing Department



Challenges of Multi-Camera Imaging

Dean Goff – Electro-Optics Department

Any questions?