<table>
<thead>
<tr>
<th>Company Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Founded in 2009</td>
</tr>
<tr>
<td>Headquartered in Sunnyvale, CA</td>
</tr>
<tr>
<td>Seven offices worldwide across the US, China, India, and Taiwan</td>
</tr>
<tr>
<td>250+ employees and growing</td>
</tr>
<tr>
<td>Expertise in targeting heterogeneous, multi-core architectures: GPGPUs, DSPs, SoCs, FPGAs, Mobile GPUs, ARM, x86</td>
</tr>
<tr>
<td>Products and Services:</td>
</tr>
<tr>
<td>1. Machine Learning and Neural Networks</td>
</tr>
<tr>
<td>2. Video/Image processing and OpenCV</td>
</tr>
<tr>
<td>3. Video Codecs (x265)</td>
</tr>
<tr>
<td>4. Compiler Technology for OpenCL, Renderscript, CUDA, C++ AMP</td>
</tr>
<tr>
<td>5. Performance Optimization Services</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Global Customers and Partners</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telesstream</td>
</tr>
<tr>
<td>AMD</td>
</tr>
<tr>
<td>Xilinx</td>
</tr>
<tr>
<td>Synopsys</td>
</tr>
<tr>
<td>Intel</td>
</tr>
<tr>
<td>Imagination</td>
</tr>
<tr>
<td>ARM</td>
</tr>
<tr>
<td>NVIDIA</td>
</tr>
</tbody>
</table>
Deep Learning Team

- Advisor & CTO - Dr. Wen-Mei Hwu
  - Walter J. Sanders Professor of Electrical Engineering at U of I, Urbana-Champaign

- Academic Advisor - Dr. Thomas S. Huang
  - Swanlund Endowed Professor Emeritus of Electrical and Computer Engineering at U of I, Urbana-Champaign

- Team of ~30 for DNN research and development

- Create optimized libraries for multi-GPU training
  - Torch7
  - Caffe

- Develop CNNs classifiers for:
  - DSPs
  - FPGAs
  - GPUs
  - Custom embedded designs
**Complete NN Workflow**

**Training Data Collection/Labelling**
- Customer can provide data or data may be collected/generated
- Labelling is performed confidentially, in-house

**Neural Network Design and Training**
- NN architecture and frameworks are designed to fit into target HW
- Explore large parameter space (learning rates, momentum, dropout, conv. layer dimensions, depth, etc.)
- Trained using our optimized versions of Torch7, Caffe, and CUDA convnet

**Application Deployment**
- Optimize for target hardware (intrinsics, assembly, RTL, OpenCL, etc.)
- Build an application around an NN classifier

**Updates and Improvements**
- Incrementally update NN with new training data
- Revise based on customer feedback
Application Domains

Advanced Driver Assistance (ADAS)
- Traffic Sign Recognition
- Lane Marker Detection
- Pedestrian Detection
- Free Space Detection

Security & Surveillance
- Face Recognition
- Voice Recognition
- Suspect Tracking
- Crowd/Riot Detection
- Threat Detection

Image & Video Classification
- Video Analytics: extract content keywords
- Image Analytics: defense, medical

Retail & Manufacturing
- Manufacturing QC
- Loss Prevention
- Inventory Tracking

Traffic Monitoring
- Vehicle Counting
- Accident Detection
- Traffic Pattern Detection
- Type Classification, e.g. truck, car, motorcycle

Video Quality Control
- Audio/Video synchronization
- Artifact Detection
- Banding Detection
- Macro Block Detection

Object Tracking & Recognition
- Faces
- Vehicles
- Gestures
- Retail Goods

Defense
- UAV Image Data Mining
- Threat Detection
- Weapon Identification
- Vehicle Identification, e.g. planes, tanks

Robotics
- Terrain Traversal
- Speech Processing
- Object Identification
- Behavior Recognition
- Adaptive Learning

MulticoreWare
Automotive OEMs and Tier 1’s exploring multiple platforms
- GPUs
- FPGAs
- DSPs
- Custom Embedded

NVIDIA Drive PX 2
- 4 GPUs
- 250W
- Liquid-Cooled
Vehicle Detection
Vehicle Detection
Pedestrian Detection
Pedestrian + Vehicle + Sign Detection
Pedestrian + Vehicle + Sign Detection
Challenges for Embedded Applications

- Need fast Detection (not just Classification)
  - Localization is expensive
  - Generating proposals is a bottleneck

- Need less complex network architectures
  - Limited memory, compute, power
  - ImageNet Benchmark: large class counts, large networks

- Fixed-Point for prediction path
  - Accuracy vs. speed
- Need fast Detection (not just Classification)
  - Localization is expensive
  - Generating proposals is a bottleneck

- Need less complex network architectures
  - Limited memory, compute, power
  - ImageNet Benchmark: large class counts, large networks

- Fixed-Point for prediction path
  - Accuracy vs. speed
Typical Detection Pipeline

- Read frame
- **Create object proposals** (e.g. Selective Search, EdgeBoxes, etc.)
- **For each** proposal:
  - Crop from frame
  - Pre-process (e.g. color, warp)
  - Run CNN classifier
- Post-process (NMS)
Where are the cycles?

- CNN Classifier: 53%
- Generate Proposals: 39%
- Crop Proposals: 8%
Object Proposals via Selective Search

- Meets 3 primary criteria for proposal generation
  - Captures multiple scales
  - Robust to scene conditions
  - Reasonably fast
- Required porting/optimization for DSP
  - Implemented in C/C++
  - Reduced cycles by ~30X
- Still significant portion of total compute time
- EdgeBoxes instead?
Fast R-CNN

- Change the pipeline
  - No need to run classifier on each proposal
  - Re-use convolution feature map across proposals

- Requires RoI Pooling layer
  - Extracts proposal features from full frame map
  - Requires custom implementation

- Still need fast Selective Search (SS)
  - Now even more limited by SS speed

*Image from Girshick, “Fast R-CNN”*
- Read frame
- Create object proposals (Selective Search)
  - For each proposal:
    - Crop from frame
    - Pre-process
    - Run CNN classifier
- Run CNN classifier on entire frame
  - Take as input a list of proposals
- Post-process (NMS)
Where are the cycles with Fast R-CNN?

- **Generate Proposals**: 91%
- **CNN Classifier**: 8%
Faster R-CNN

- State-of-the-Art Localization + Classification
  - 14+ implementations by 2015 ImageNet competitors
  - Used by winners 12/13 categories
- Use a CNN to create proposals
  - RPN (region proposal network)
  - Re-use convolution feature map for localization & classification
- Eliminates the need for prior object proposals
  - No more Selective Search or EdgeBoxes

Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

*Image from Ren et al., “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”*
- Read frame
- Create object proposals
  - Run CNN classifier on entire frame
    - Run convolution layers
    - Generate proposals using feature map and region proposal network
    - Re-use feature map for classification
- Post-process (NMS)
Challenges for Embedded Applications

- Need fast Detection (not just Classification)
  - Localization is expensive
  - Generating proposals is a bottleneck

- Need less complex network architectures
  - Limited memory, compute, power
  - ImageNet Benchmark: large class counts, large networks

- Fixed-Point for prediction path
  - Accuracy vs. speed
Network Architecture Choices

- Layer weights vs. internal memory size
- Design networks with much lower memory & MAC counts
  - AlexNet 10-layer /w 7CL+3FC /w ~1 GMACs, 400MB weight-space
  - 5-Layer network /w 3CL+2FC /w 5 MMACs, 512KB weight-space
- New techniques: Pruning (Han et al., “Deep Compression”)
Network Architecture Choices

- MSRA Deep Residual Learning
  - New insights allow deeper networks
  - ImageNet 2015 Winner – 152 layers!
  - Won by significant margin in both localization and classification
  - Lower FLOPs and parameters than VGG-19!
Challenges for Embedded Applications

- Need fast Detection (not just Classification)
  - Localization is expensive
  - Generating proposals is a bottleneck

- Need less complex network architectures
  - Limited memory, compute, power
  - ImageNet Benchmark: large class counts, large networks

- Fixed-Point for prediction path
  - Accuracy vs. speed
Fixed-Point Considerations

- For prediction/testing path, CNNs appear robust
  - <1% loss in accuracy when using 8-bit fixed point!

- Training may need some special care
  - Stochastic rounding (Gupta et al., IBM Research)
  - Dynamic fixed point (Courbariaux et al.)

- Deeper networks
  - MSRA ImageNet 2015 Winner – 152 layers
  - Implications for error?
Current Work

- Implementation of Faster R-CNN

- Investigating deep residual learning networks
  - Less computation
  - Fewer parameters

- Investigating fixed-point accuracy for residual networks
Contact Me

- Anshu Arya
  - anshu@multicorewareinc.com