STRADVISION

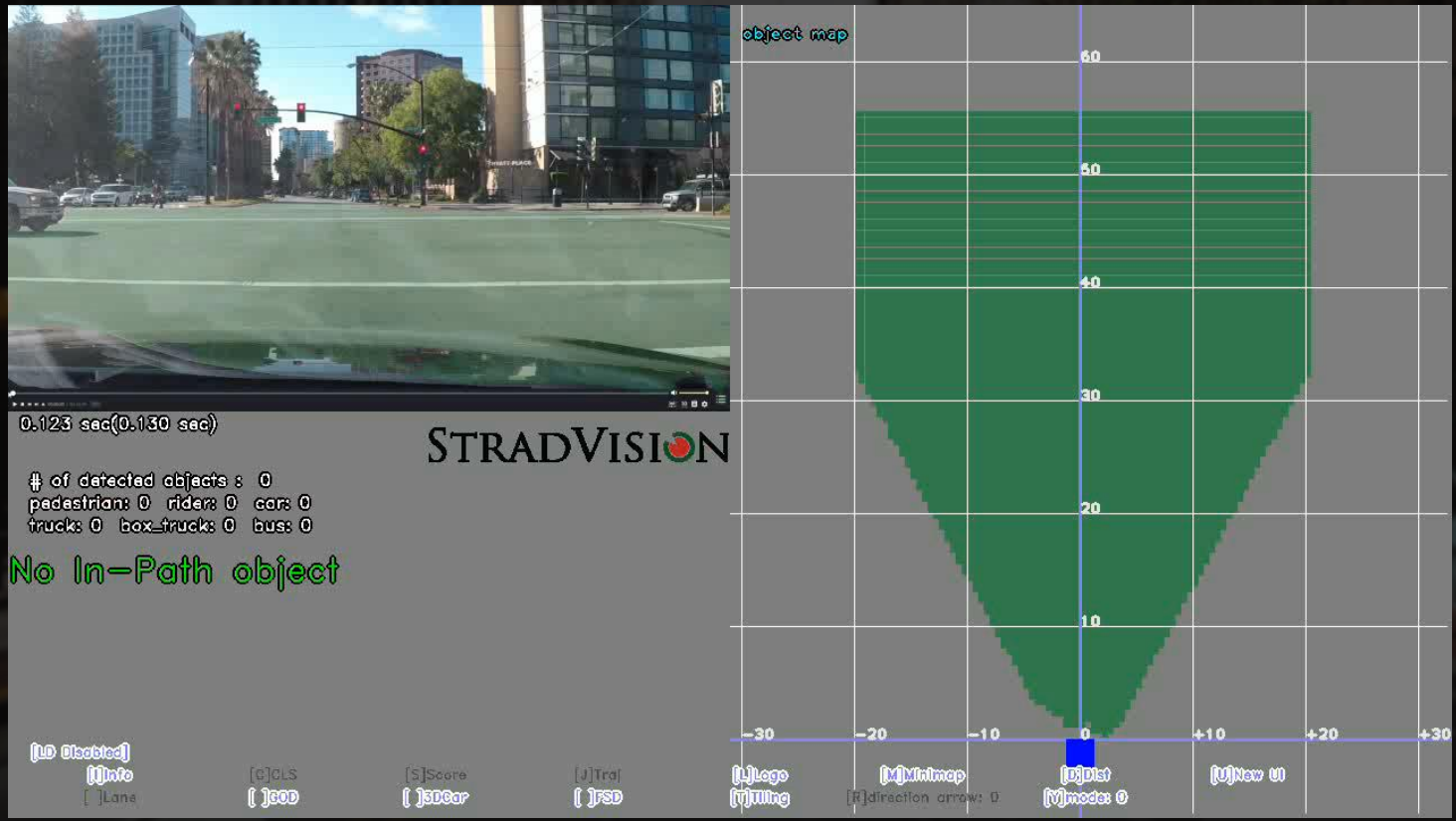
Autonomous Driving AI

SVNet

A Deep-Learning-Based Perception for ADAS and Autonomous Driving

Robustness

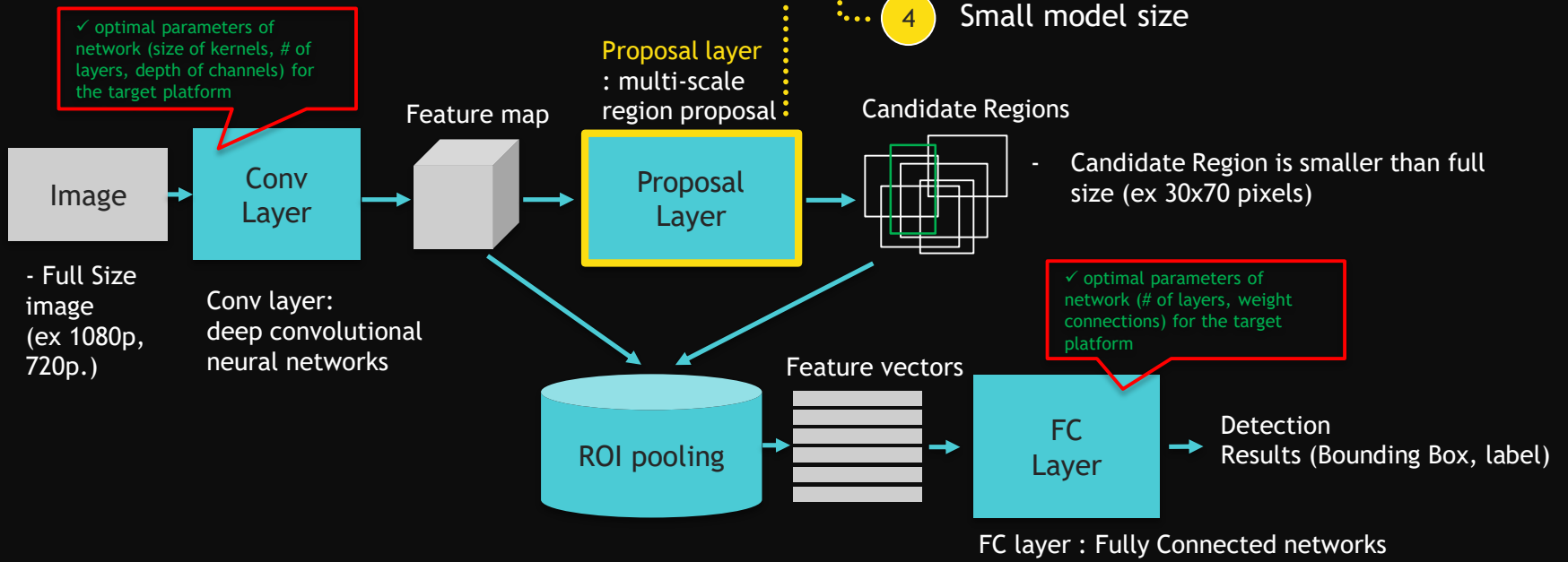
SVNet@NVIDIA Jetson TX2



SVNet
Algorithm Flow

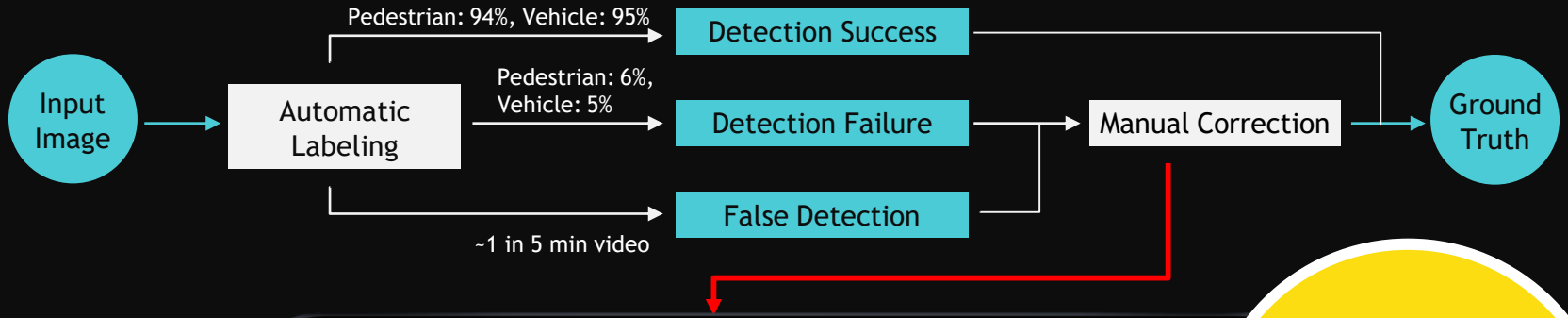
SVNet has a unique 'Proposal layer'.

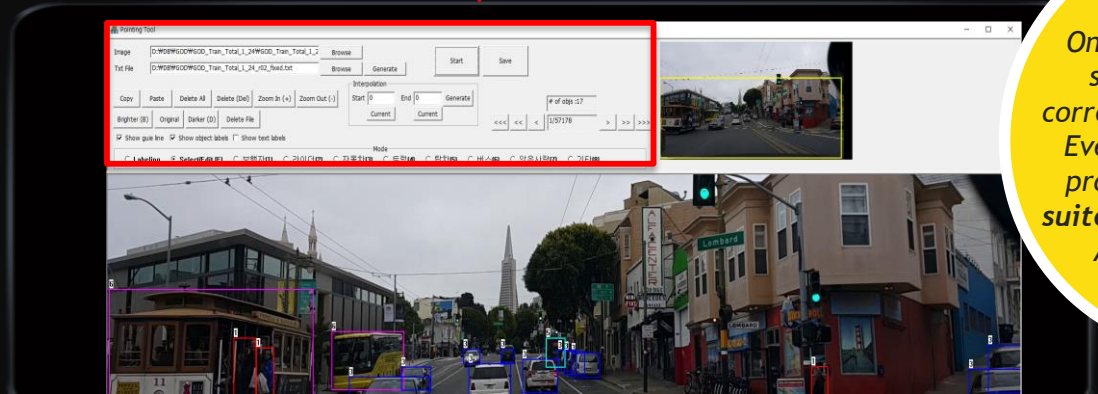
1. Robust for bad situations
2. Small object sizes
3. Robust for occlusion
4. Small model size



SVNET AI TRAINING

Customer may let 'SVNet' recognize a new object in fast and easy way.

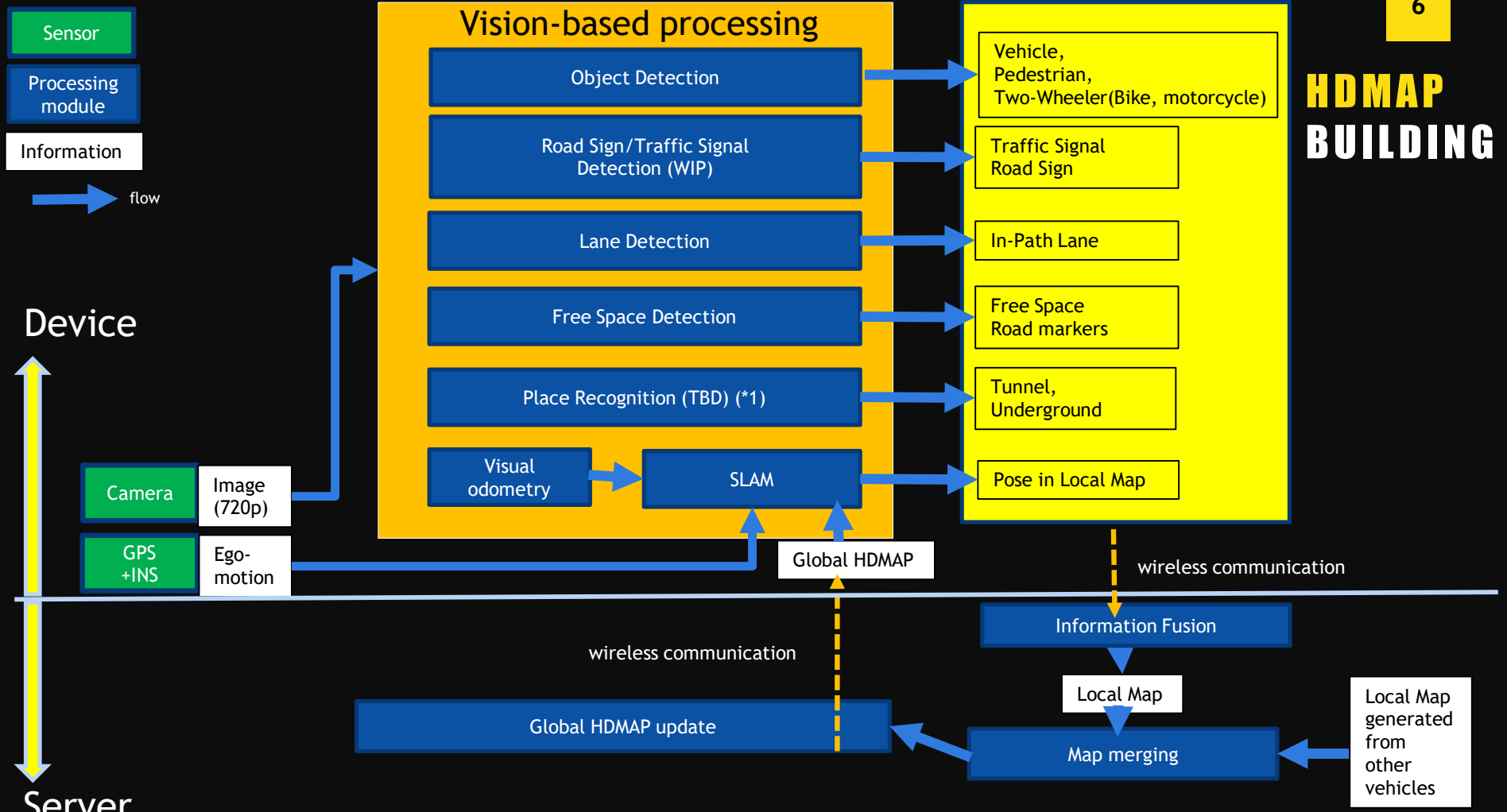




SVN Training Suite

*Only **5%** of the objects should be manually corrected in input images. Even more, Stradvision provides **'SVN training suite'** application to make AI training process **easy**.*

HDMAP BUILDING

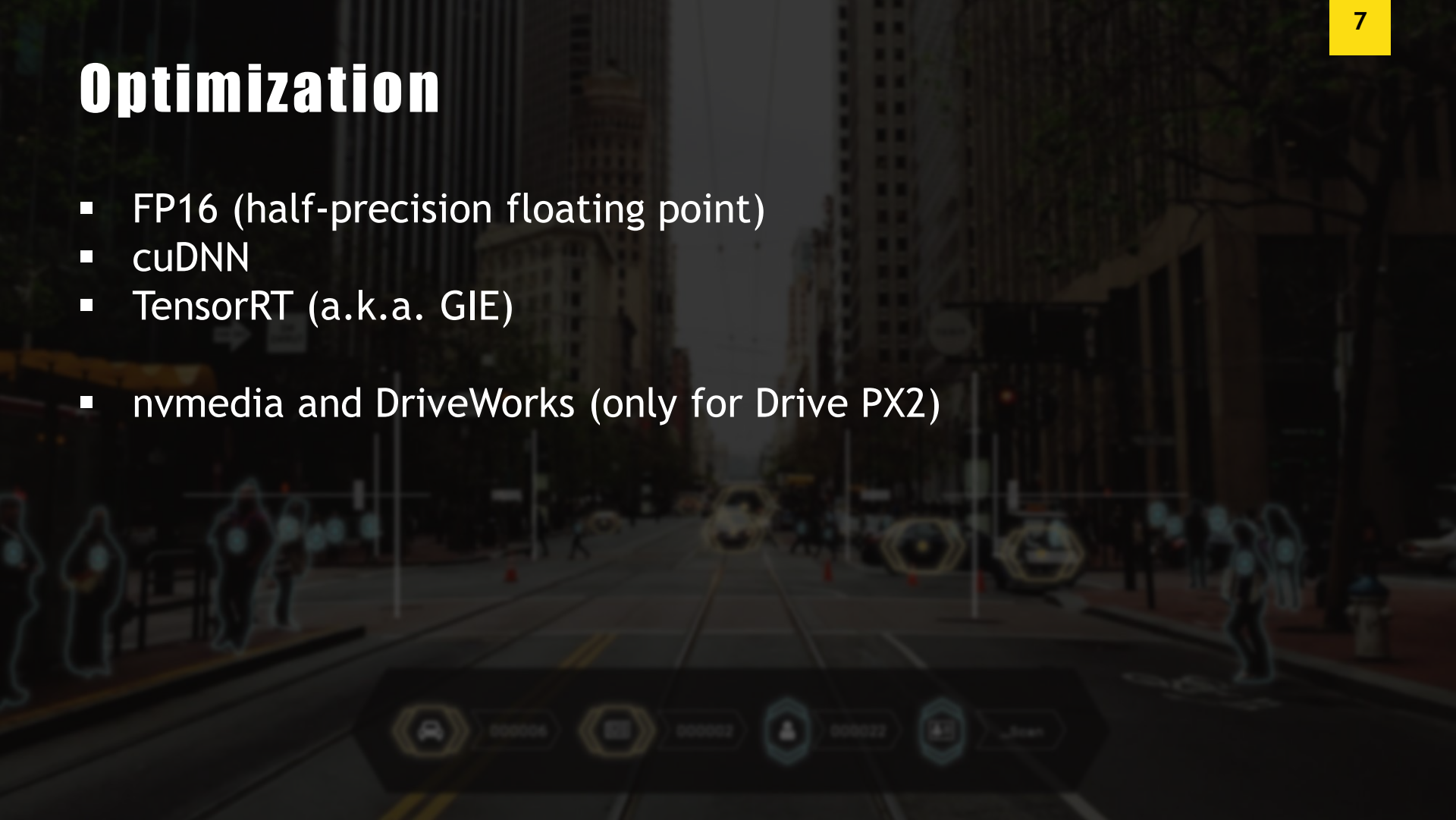


(*1) Can be developed upon customer's request

Optimization

- FP16 (half-precision floating point)
- cuDNN
- TensorRT (a.k.a. GIE)

- nvmedia and DriveWorks (only for Drive PX2)



FP16 and half2

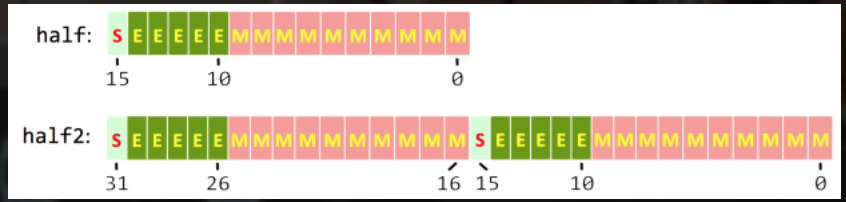
FP32: single-precision floating point (float type)
FP64: double-precision floating point (double type)
FP16: half-precision floating point (??? type)

FP16 IEEE754 standard can present ($\pm 5.96*10^{-8}$ ~ $6.55*10^{4}$), zero and infinity.

half2: two FP16 data is packed in one 32-bit space.

Some hardwares (e.g. Parker) support native FP16 types and intrinsics (half and half2)

half2-type instructions are SIMD with 2 data.
half2 is ***2x faster*** than half or float.



images from: https://devblogs.nvidia.com/parallelforall/new-features-cuda-7-5/

cuDNN

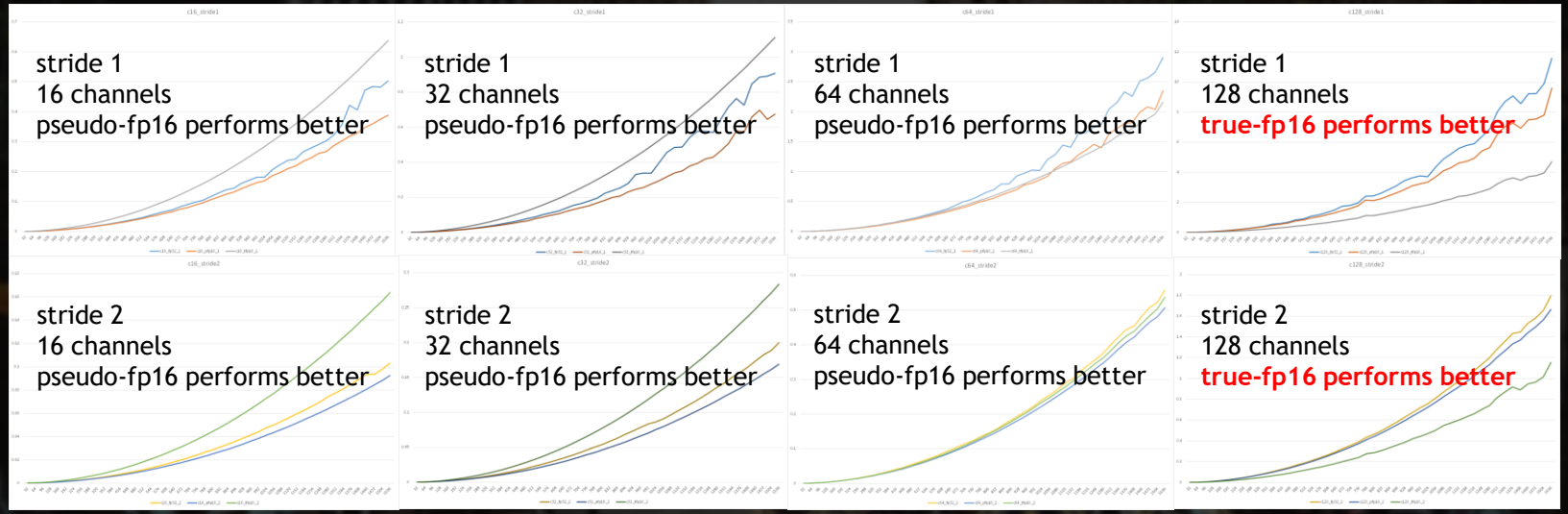
cuDNN: Deep Neural Network library for NVIDIA GPU

- cuDNN provides the fastest convolution method for SVNet on TX2 and PX2.
  - (We've tried OpenBLAS, CLBLAS, cuBLAS, MKL, TensorRT and so on)

- cuDNN supports FP16 types.
  - pseudo-FP16: load/store FP16, calculate **FP32**
  - true-FP16: load/store FP16, calculate **FP16**

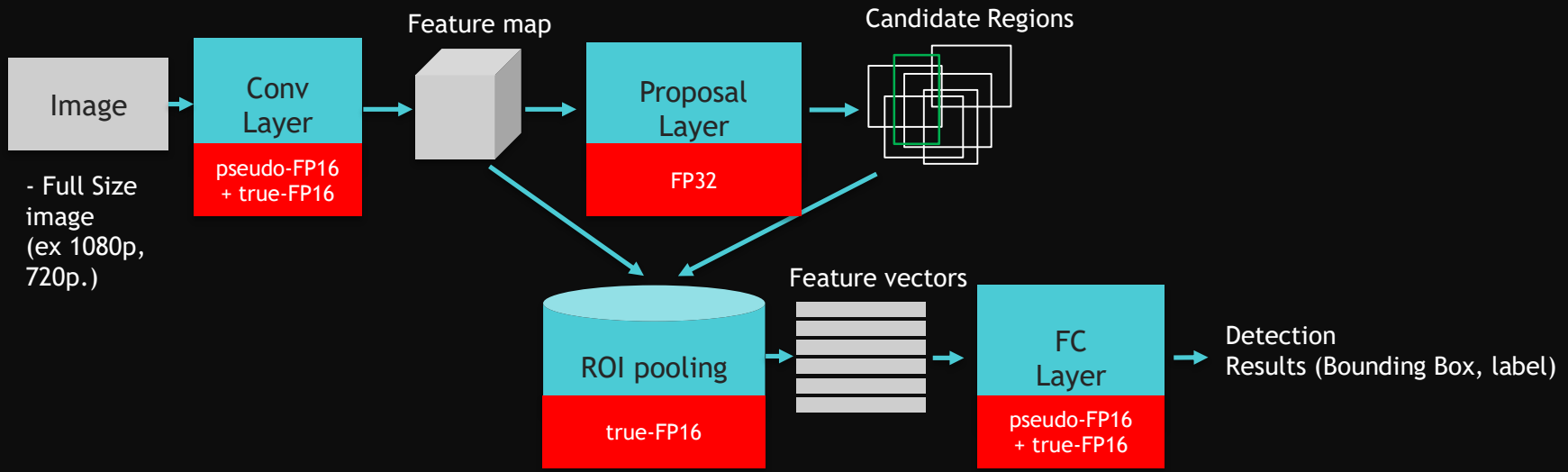
- To maximize performance, you have to find a specific configuration for each convolution.

cuDNN convolution performance



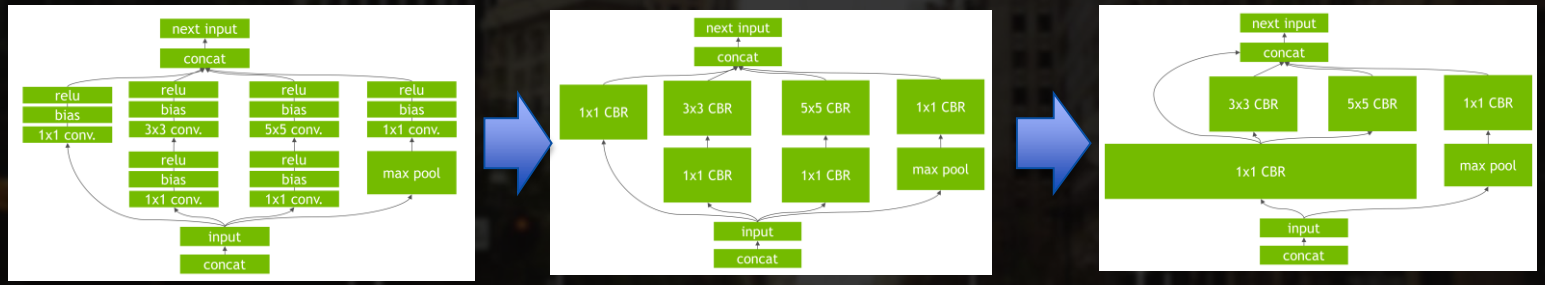
- In most cases, pseudo-FP16 performs better.
- In SVNet, a certain combination of pseudo-FP16 and true-FP16 is the fastest.

SVNet
The Fastest Combination



TensorRT (a.k.a. GIE)

TensorRT: Inference engine which optimizes network dynamically.



- It really runs hundreds of configurations of algorithms of layers with specified sizes and find the fastest configuration.
- If the network is made with only TensorRT-supported layers, TensorRT can be a good solution to optimize with less work.
- Extremely hard to debug.
- Sometimes slower than own implementation. (e. g. CReLU)

images from: https://devblogs.nvidia.com/parallelforall/deploying-deep-learning-nvidia-tensorrt/

TensorRT (a.k.a. GIE)

With TensorRT 2.1, CReLU is processed like “do Scale” → “do ReLU”.

```
layer {
  name: "conv2_2/conv" type: "Convolution"
  bottom: "conv2_1" top: "conv2_2_p"
  convolution_param {
    num_output: 16 kernel_size: 3 pad: 1
    stride: 1 group: 1
  }
}
```

```
---------- Timing conv2_2/conv(2)
Tactic 7 time 1.5872
Tactic 10 time 2.9521
Tactic 14 time 2.55718
Tactic 15 time 1.72042
Tactic 25 time 1.89606
Tactic 26 time 5.42506
Tactic 29 time 2.71546
             :
             :
Tactic 159 time 1.49738
Tactic 162 time 1.85568
Tactic 164 time 2.96736
```

```
layer {
  name: "conv2_2/invert" type: "Scale"
  bottom: "conv2_2_p" top: "conv2_2_m"
  scale_param { filler { type: "constant" value: -1.0 } bias_term: false }
}
layer {
  name: "conv2_2/concat" type: "Concat"
  bottom: "conv2_2_p" bottom: "conv2_2_m" top: "conv2_2/c_relu"
  concat_param { axis: 1 }
}
layer {
  name: "conv2_2/c_relu" type: "ReLU"
  bottom: "conv2_2/c_relu" top: "conv2_2/c_relu"
}
```

```
---------- Timing conv2_2/invert(10)
Tactic 0 is the only option, timing skipped
---------- Timing conv2_2/c_relu(0)
Tactic 0 is the only option, timing skipped
```

- There is no optimized tactic implemented.
- With our own CReLU implementation, performance got slightly better.

nvmedia and DriveWorks

GMSL camera interface is supported by Drive PX2.

nvmedia and DriveWorks help to capture with GMSL camera.

DriveWorks provides higher-level APIs than nvmedia. (Sensor, Display, etc.)

GoPro with HDMI2USB devices: at least 50ms delay.
GMSL camera: less than 20ms delay.

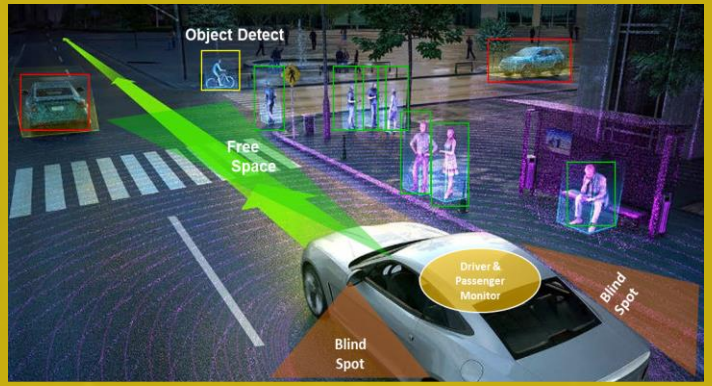
STRADVISION

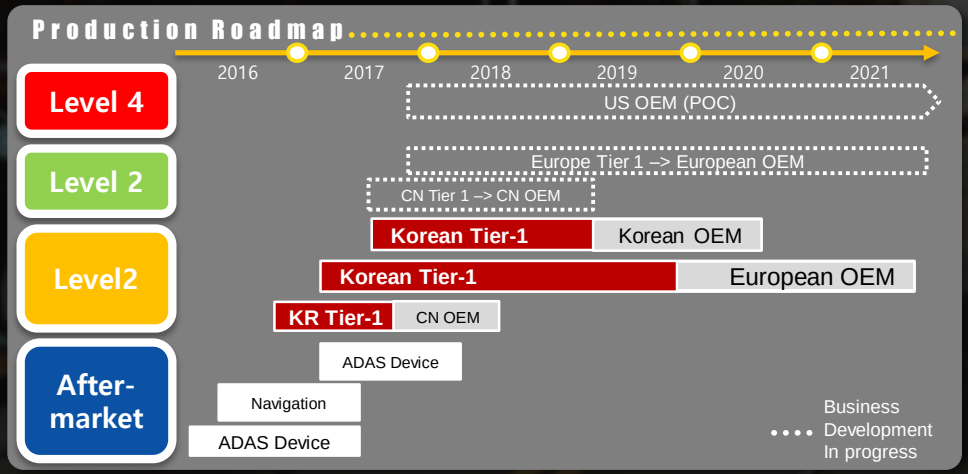
Strong Academic Background

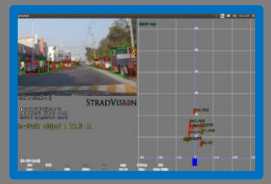
- 6 Ph.D & 17 MS, mostly from POSTECH
- 70 Employees
  (14 Algorithm engineers, 11 Optimization engineers, 3 Application engineer, 3 Data engineer, 1 Project manager, 2 Business Developer, 2 Operation Manager, 31 Data labeler)

Extensive Knowledge/Experience with Various Hardware Platforms

- 9 members from Intel, worked on various hardware platforms at Intel
- 6 members from Olaworks, worked with many smartphone OEMs (e.g. LGE, HTC, Samsung)
- 3 members from automotive industry, e.g. Yazaki, Denso, Mando-Hella, and TI

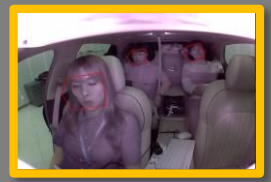




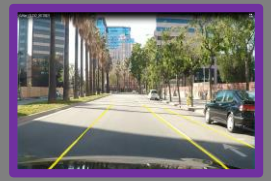

Object Detection

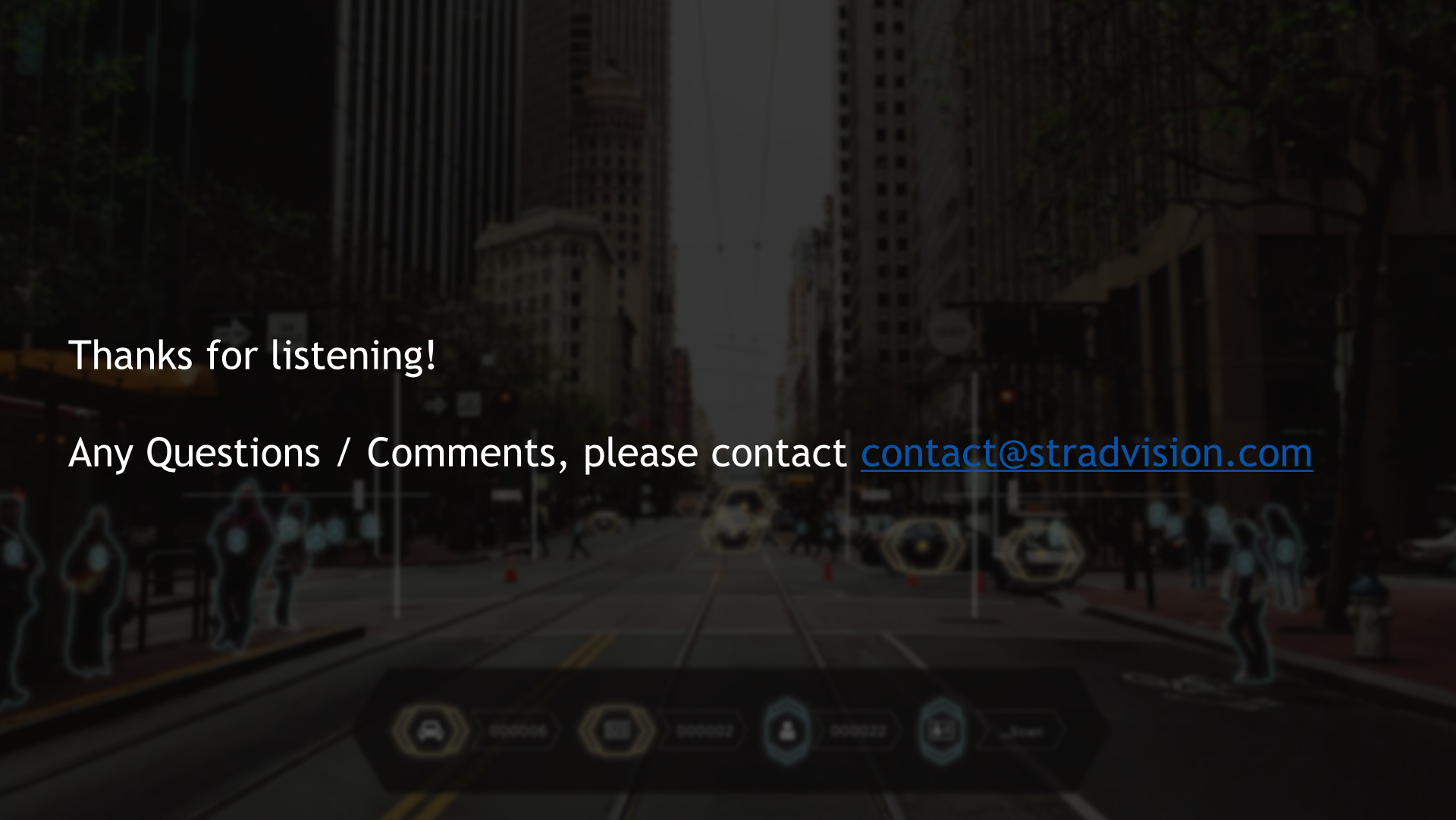

Free Space Detection


Blind Spot Detection


Driver Monitoring


Text Recognition


Lane Detection



Thanks for listening!

Any Questions / Comments, please contact contact@stradvision.com