SVNet

A Deep-Learning-Based Perception for ADAS and Autonomous Driving
Robustness
SVNet@NVIDIA Jetson TX2

No In-Path object
SVNet

Algorithm Flow

SVNet has a unique 'Proposal layer'.

Conv layer: deep convolutional neural networks

Proposal layer: multi-scale region proposal

ROI pooling

Feature map

Feature vectors

FC layer: Fully Connected networks

Detect Results (Bounding Box, label)

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

✓ optimal parameters of network (size of kernels, # of layers, depth of channels) for the target platform

✓ optimal parameters of network (# of layers, weight connections) for the target platform

- Candidate Region is smaller than full size (ex 30x70 pixels)

- Full Size image (ex 1080p, 720p.)
SVNET AI TRAINING

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

- Detection Success: Pedestrian: 94%, Vehicle: 95%
- Detection Failure: Pedestrian: 6%, Vehicle: 5%
- False Detection

- Manual Correction

Input Image → Automatic Labeling → Detection Success → Ground Truth

~1 in 5 min video

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.
Vision-based processing

- Object Detection
- Road Sign/Traffic Signal Detection (WIP)
- Lane Detection
- Free Space Detection
- Place Recognition (TBD) (*1)
- Visual odometry
- SLAM
- Global HD Map

Device

- Camera (720p)
- GPS + INS
- Ego-motion

Information flow

Server

- Global HD Map update
- Information Fusion
- Local Map
- Map merging

Information Fusion

Local Map generated from other vehicles

(*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 \((\pm 5.96 \times 10^{-8} \sim 6.55 \times 10^{4})\), 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: 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 \textbf{FP32}
  - true-FP16: load/store FP16, calculate \textbf{FP16}

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

- Full Size image (ex 1080p, 720p.)

Feature map

Proposal Layer
FP32

Candidate Regions

Feature vectors

ROI pooling
ture-FP16

Detection Results (Bounding Box, label)

FC Layer
pseudo-FP16 + true-FP16

Conv Layer
pseudo-FP16 + true-FP16

Image
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”.

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

**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

**Production Roadmap**

- **Level 4**
- **Level 2**
- **Level 2**
- **Aftermarket**

- **2016**
- **2017**
- **2018**
- **2019**
- **2020**
- **2021**

**Product**
- **ADAS Device**
- **NAVIGATION**
- **Korean Tier-1**
- **KR Tier-1**
- **CN Tier-1**
- **Europe Tier 1**
- **US OEM (POC)**
- **European OEM**
- **CN OEM**
- **CN Tier 1 -> CN OEM**

- **Free Space Detection**
- **Object Detection**
- **Driver Monitoring**
- **Text Recognition**
- **Lane Detection**
Thanks for listening!

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