AUTONOMOUS DRONE NAVIGATION WITH DEEP LEARNING

Nikolai Smolyanskiy, Alexey Kamenev, Jeffrey Smith

NVIDIA

Project Redtail

May 8, 2017

100% AUTONOMOUS FLIGHT OVER 1 KM FOREST TRAIL AT 3 M/S

AGENDA

Why autonomous path navigation?

Our deep learning approach to navigation

System overview

Our deep neural network for trail navigation

SLAM and obstacle avoidance

WHY PATH NAVIGATION?

Drone / MAV Scenarios

Industrial inspection

Search and rescue

Video and photography

Delivery

Drone racing

WHY PATH NAVIGATION?

Land Robotics Scenarios

Delivery

Security

Robots for hotels, hospitals, warehouses

Home robots

Self-driven cars

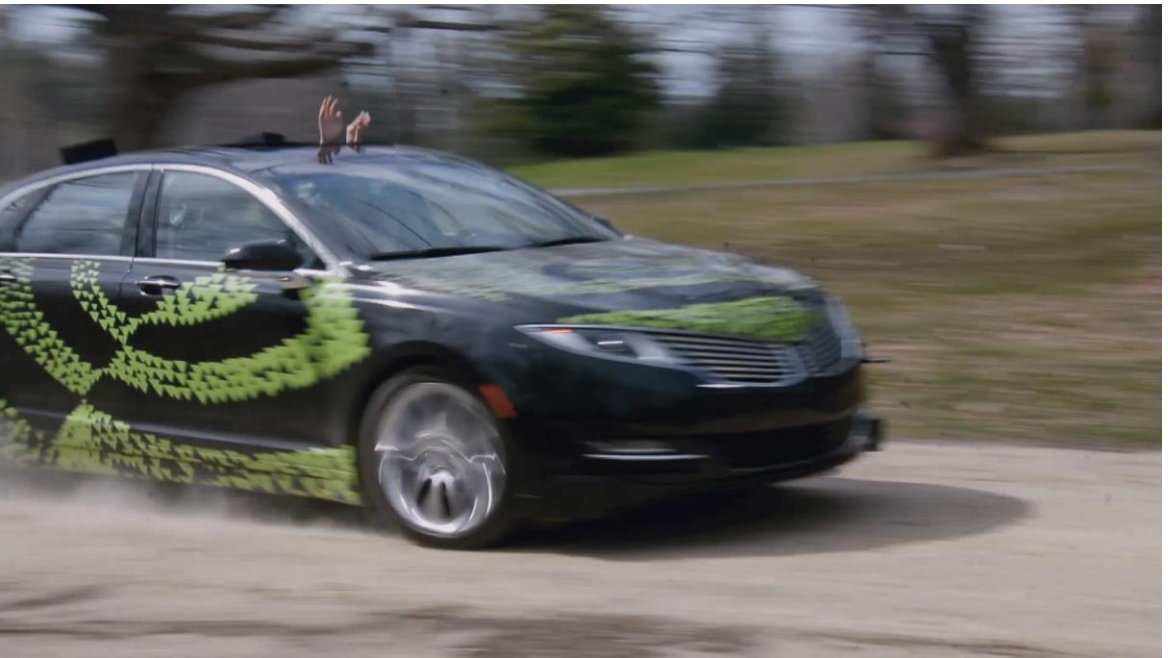
DEEP LEARNING APPROACH

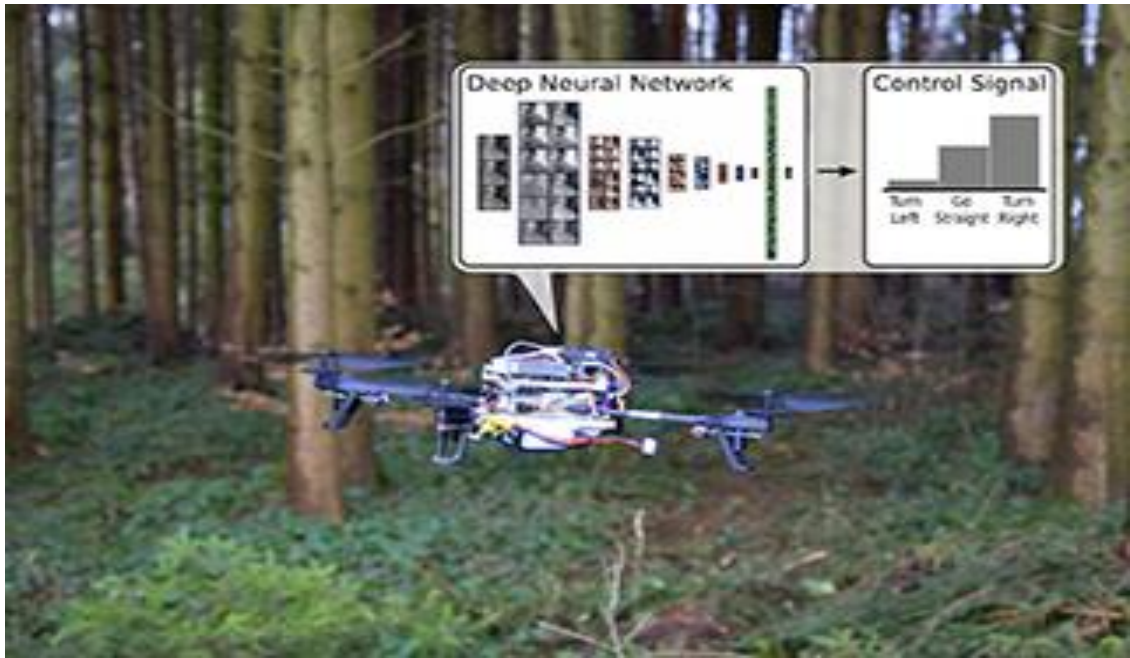
Can we use vision only navigation?

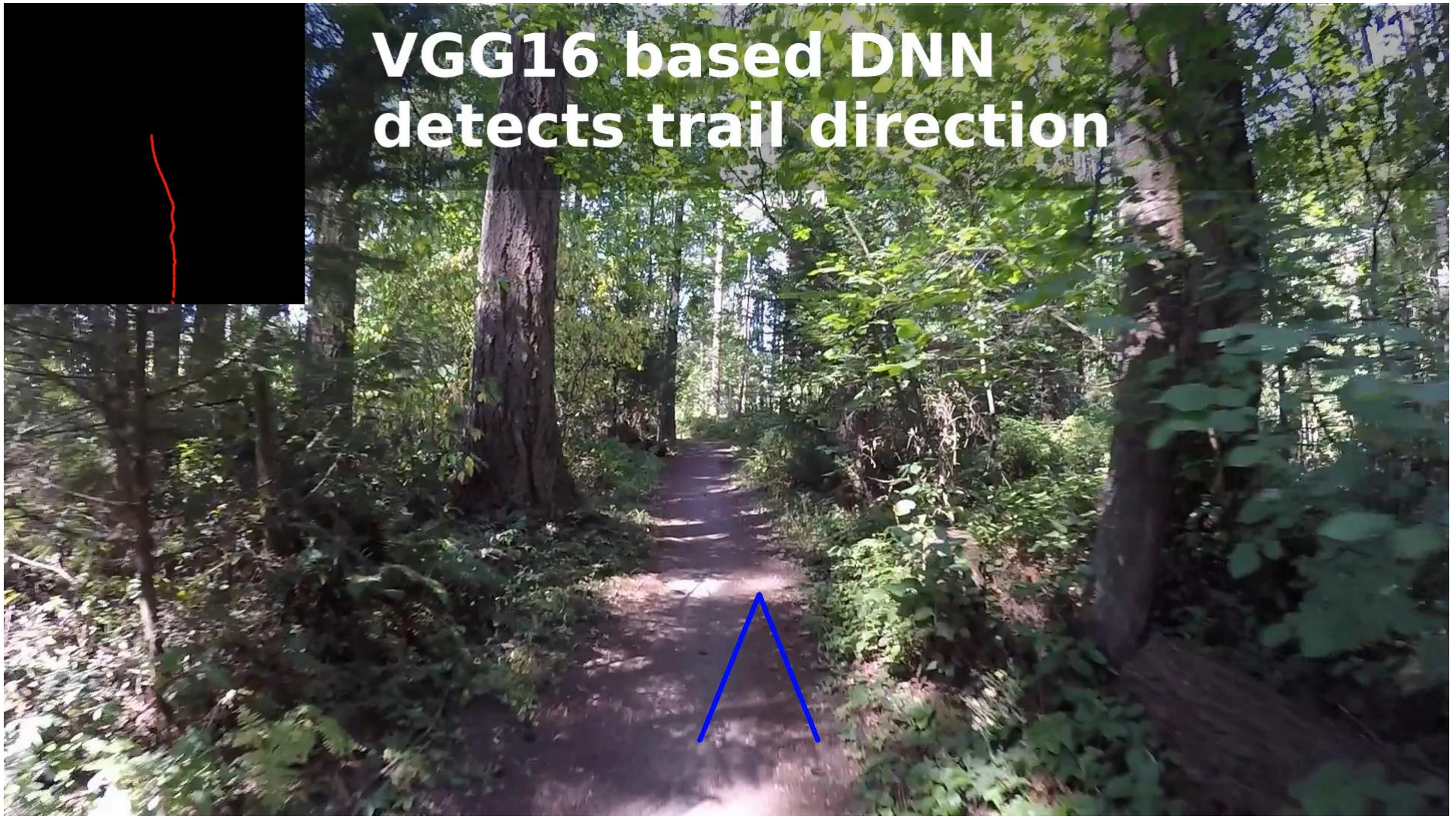
NVIDIA's end-to-end self-driving car

Giusti et al. 2016, IDSIA / University of Zurich

Several research projects used DL and ML for navigation



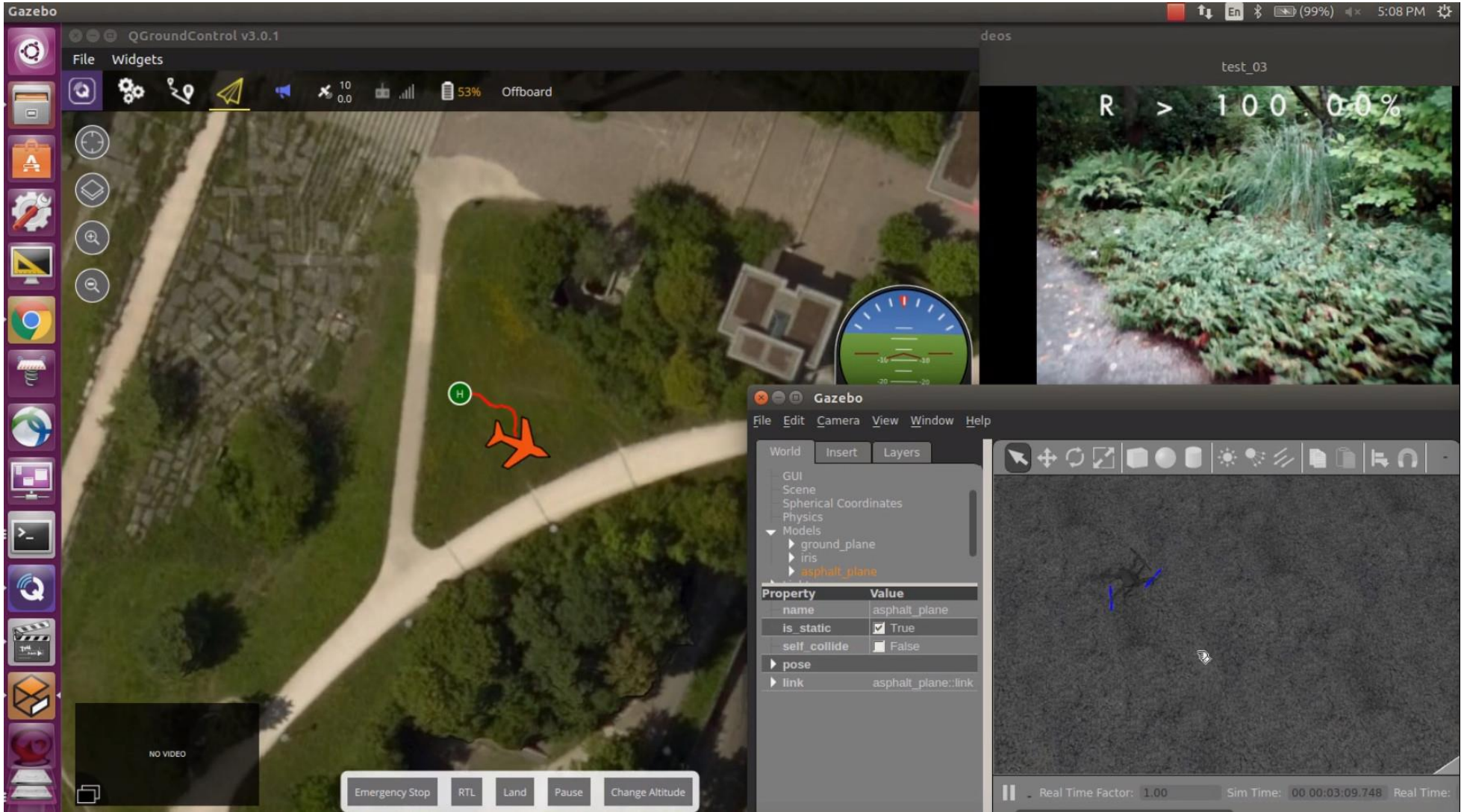




OUR PROTOTYPE FOR TRAIL NAVIGATION WITH DNN

SIMULATION

We used software in the loop simulator (Gazebo based)



PROJECT PROGRESS

PROJECT TIMELINE

Autonomous Drone Navigation With Deep Learning

Flight Over 250m Trail

Nikolai Smolyanskiy, Alexey Kamenev, Jeffrey Smith

NVIDIA Corporation

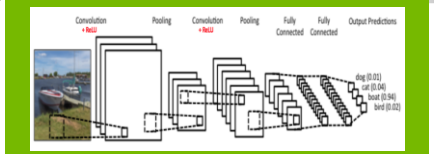
GPU Technology Conference
8-11 May 2017, Session #S7172

100% AUTONOMOUS FLIGHT OVER 250 METER TRAIL AT 3 M/S

DATA FLOW

Camera

TrailNet DNN



Steering Controller

$$f(x)$$

Pixhawk Autopilot

Image Frame:
640x360

Probabilities of:
3 views: Left, Center, Right
3 positions: Left, Middle, Right

Next waypoint:
Position
Orientation

TRAINING DATASETS

Automatic labelling from left, center, right camera views

IDSIA, Swiss Alps dataset: 3 classes, 7km of trails, 45K/15K train/test sets




Our own Pacific NW dataset: 9 classes, 6km of trails, 10K/2.5K train/test sets




Giusti et al. 2016

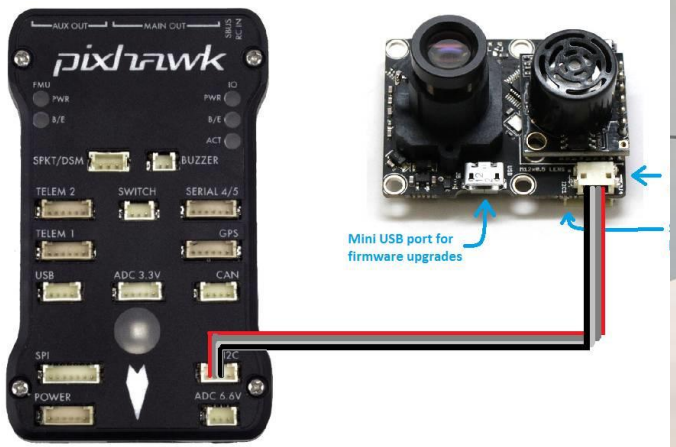
HARDWARE SETUP

Customized 3DR Iris+ with Jetson TX1/TX2

We use a simple 720p front facing webcam as input to our DNNs

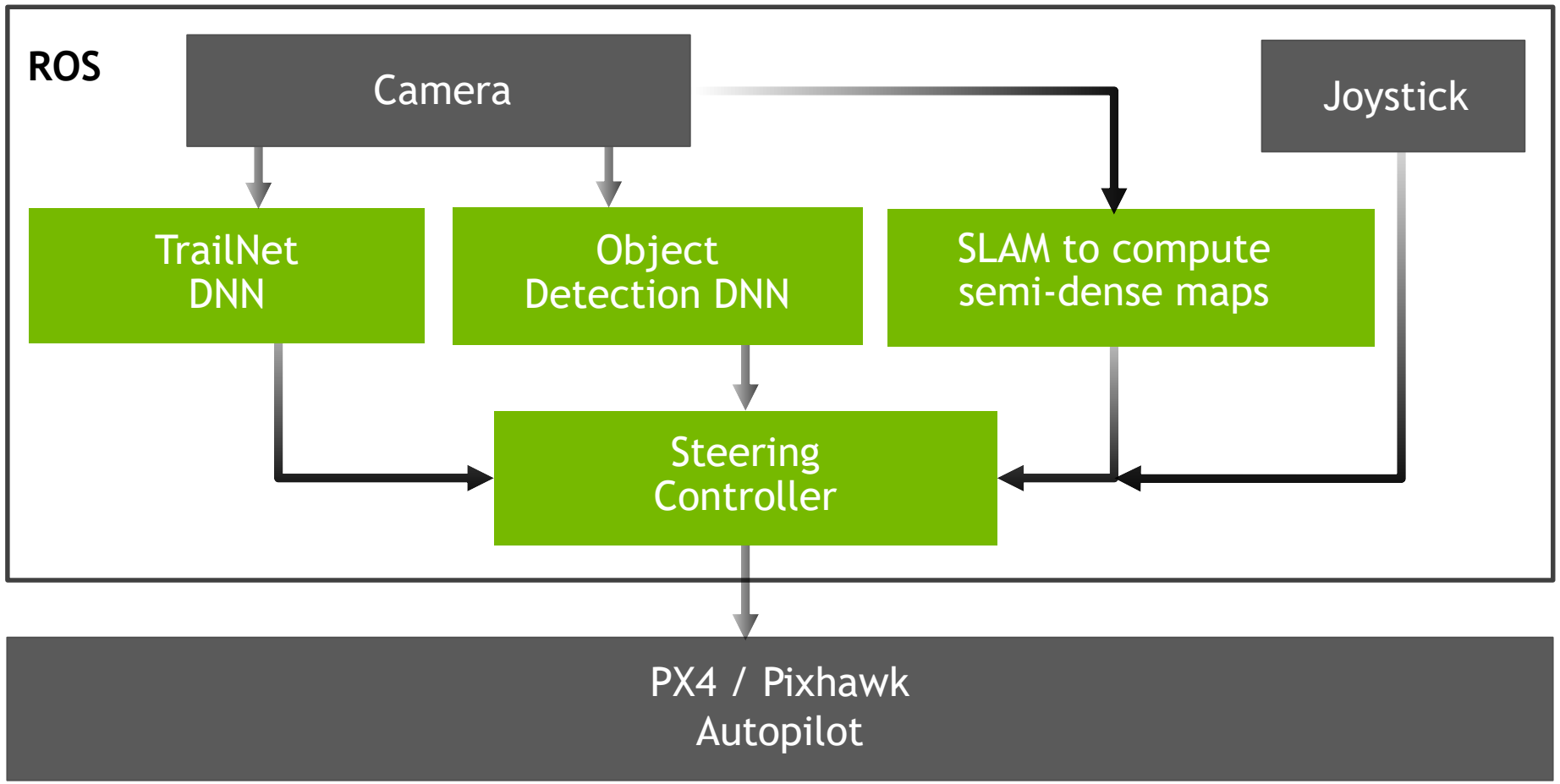
Pixhawk and PX4 flight stack are used as a low level autopilot

PX4FLOW with downfacing camera and Lidar are used for visual-inertial stabilization



SOFTWARE ARCHITECTURE

Our runtime is a set of ROS nodes



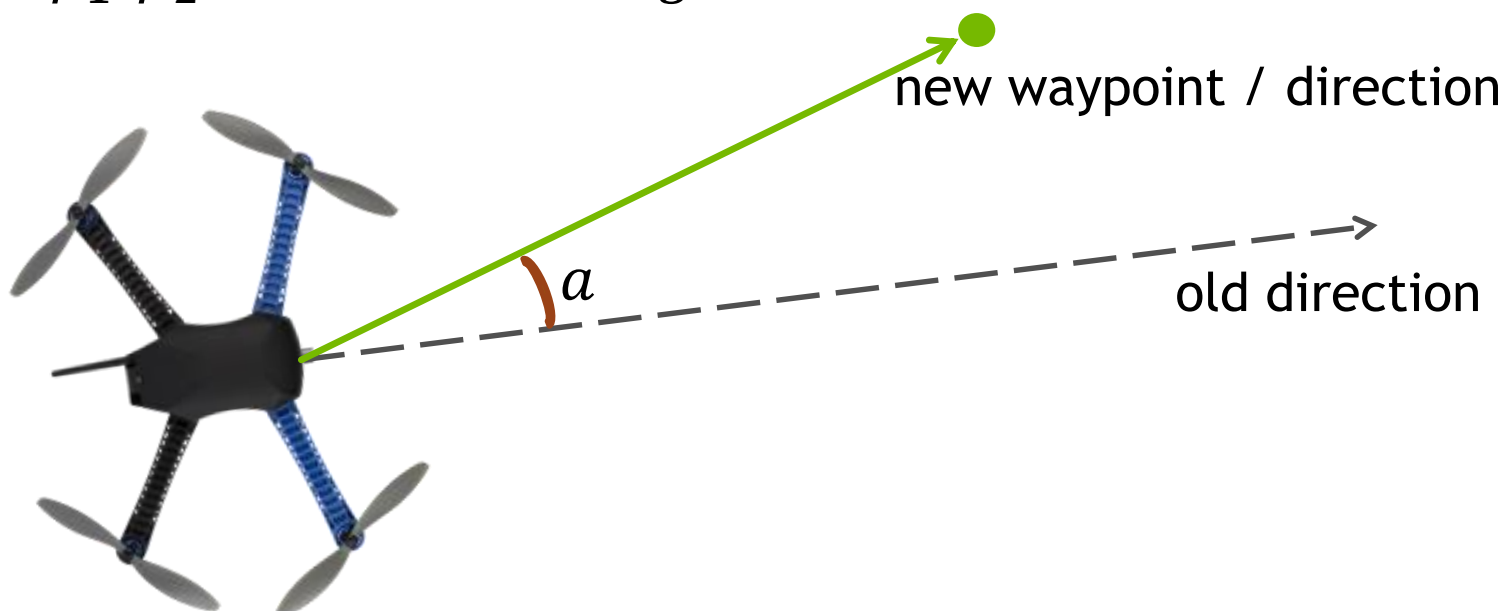
CONTROL

Our control is based on waypoint setting

$$a = \beta_1 \left(Pr(view_{right}|image) - Pr(view_{left}|image) \right) + \beta_2 \left(Pr(side_{right}|image) - Pr(side_{left}|image) \right)$$

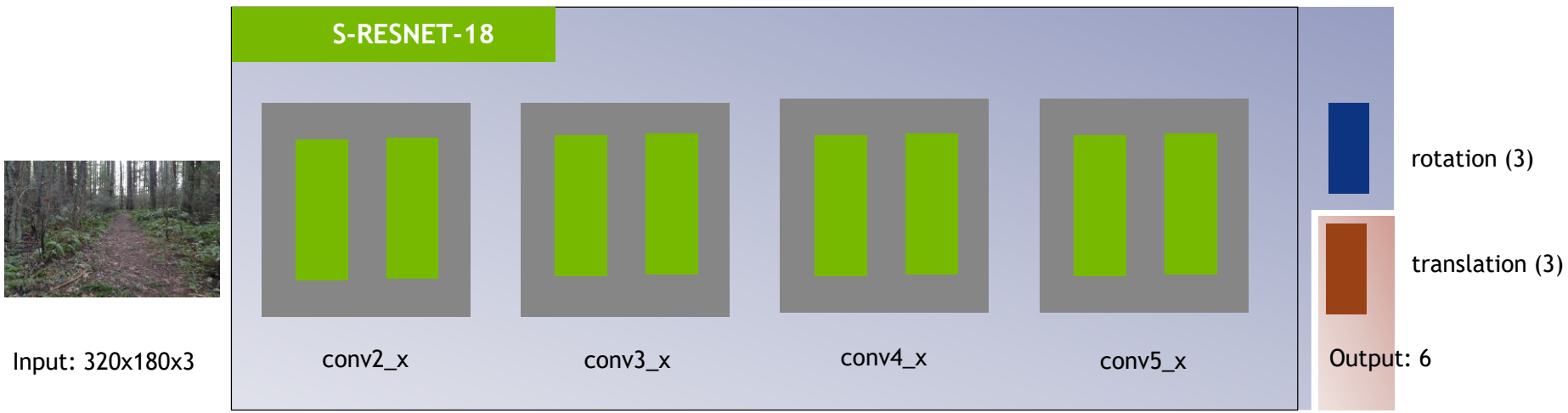
a – "steering" angle; β_1, β_2 – "reaction" angles

$a > 0$ turns left,
$a < 0$ turns right



TRAILNET DNN

1. Train ResNet-18-based network (**rotation** only) using large Swiss Alps dataset



2. Train **translation** only using small PNW dataset

K. He et al. 2015

TRAILNET DNN

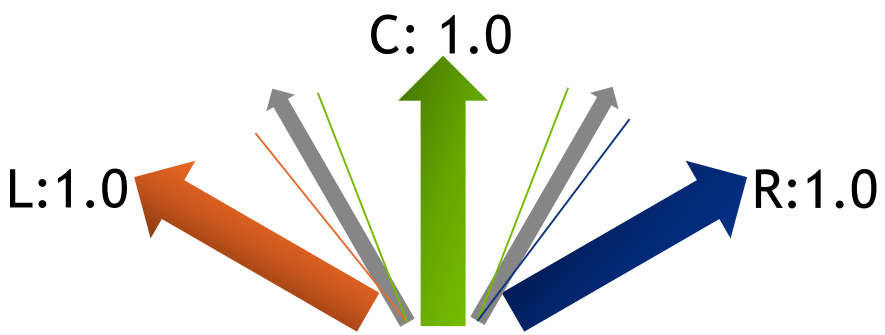
Training with custom loss

Classification instead of regression

Ordinary cross-entropy is not enough:

1. Images may look similar and contain label noise

2. Network should not be over-confident



TRAILNET DNN

Training with custom loss

Loss:

$$L = -\sum_i p_i \ln(y_i) - \alpha(-\sum_i y_i \ln(y_i)) + \beta\theta$$

$\boldsymbol{y}: softmax\ output$

$\boldsymbol{p}: smoothed\ labels$

$\alpha, \beta: scalars$

where

Softmax cross entropy with label smoothing (smoothing deals with noise)

Model entropy (helps to avoid model over-confidence)

Cross-side penalty (improves trail side predictions)

$t = argmax(\boldsymbol{p})$

$$\theta = \begin{cases} y_{2-t}, & t = 0, 2 \\ 0, & t = 1 \end{cases}$$

V. Mnih et al. 2016

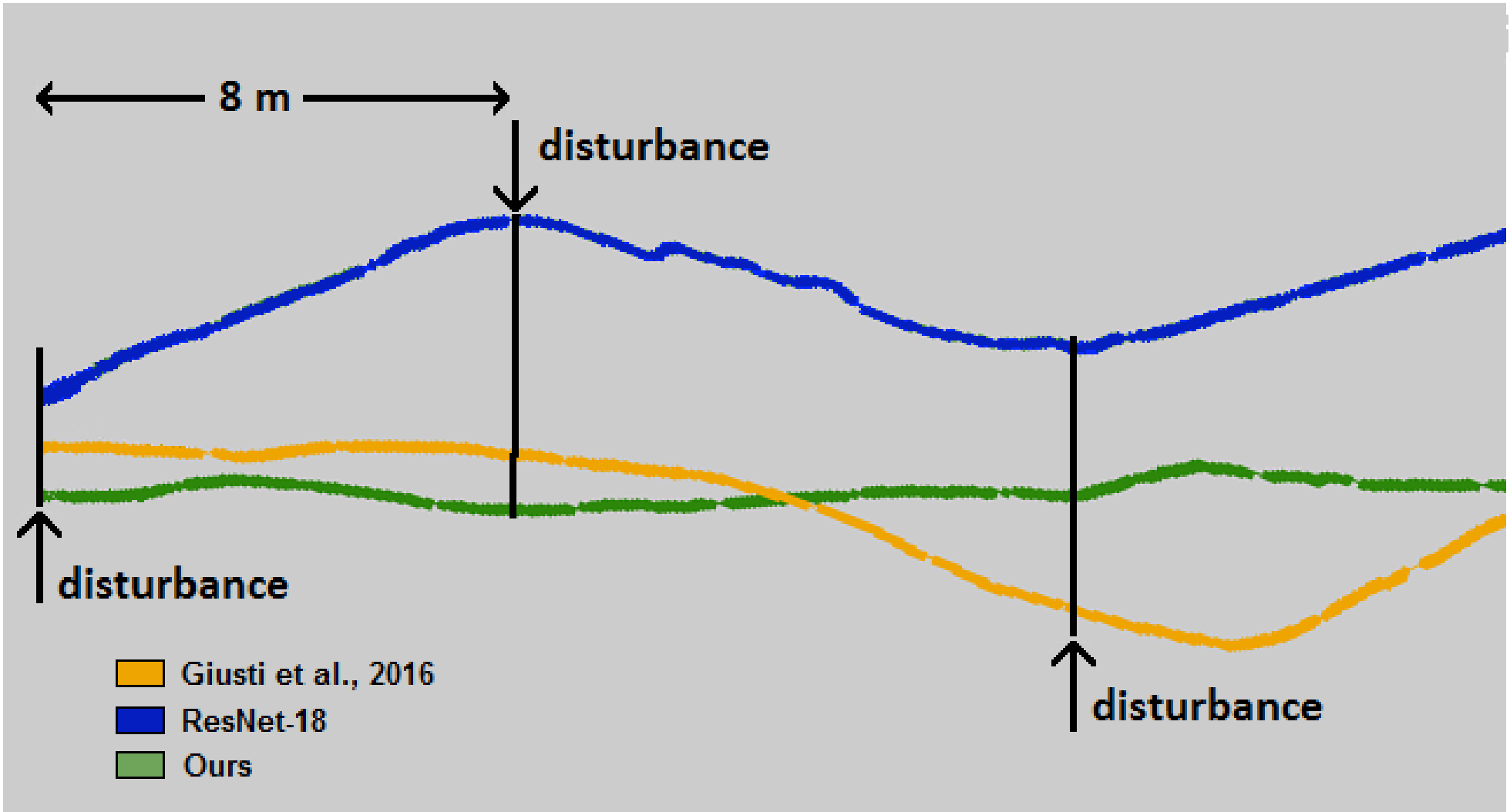
DNN ISSUES

DNN EXPERIMENTS

NETWORK	AUTONOMY	ACCURACY (ROTATION)	LAYERS	PARAMETERS (MILLIONS)	TRAIN TIME (HOURS)
S-ResNet-18	100%	84%	18	10	13
SqueezeNet	98%	86%	19	1.2	8
Mini AlexNet	97%	81%	7	28	4
ResNet-18 CE	88%	92%	18	10	10
Giusti et al.	80%	79%	6	0.6	2

[K. He et al. 2015]; [F. Iandola et al. 2016]; [A. Krizhevsky et al. 2012]; [A. Giusti et al. 2016];

DISTURBANCE TEST

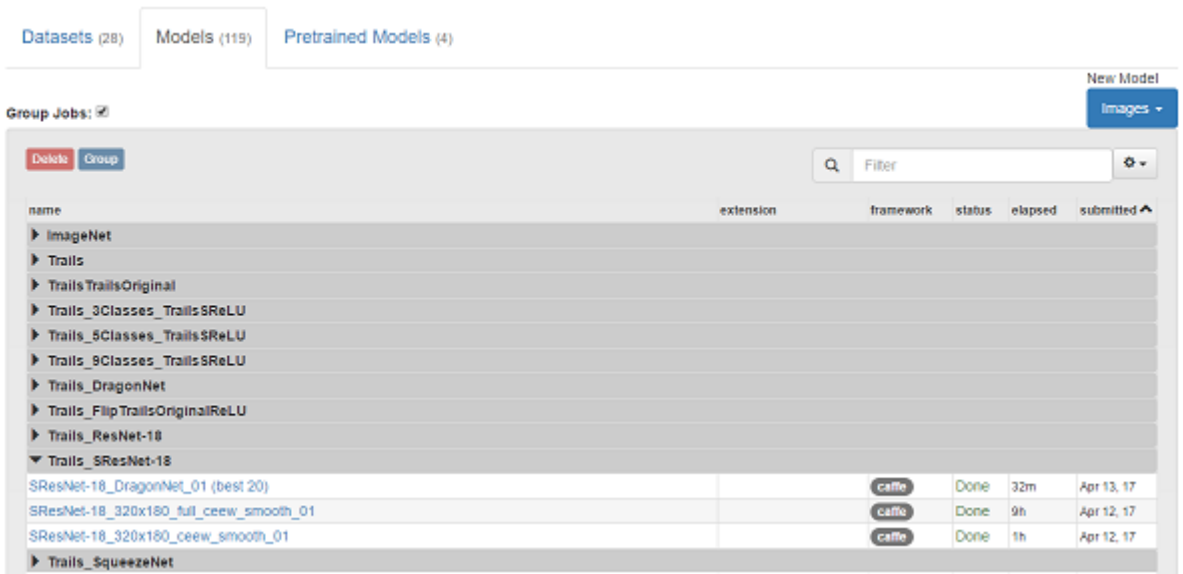


MORE TRAINING DETAILS

Data augmentation is important: flips, scale, contrast, brightness, rotation etc

Undersampling for small nets, oversampling for large nets

Training : Caffe + DIGITS



Inference: Jetson TX-1/TX-2 with TensorRT

RUNNING ON JETSON

NETWORK	FP PRECISION	TX-1 TIME (MSEC)	TX-2 TIME (MSEC)
ResNet-18	32	19.0	11.1
S-ResNet-18	32	21.7	14.0
	16	11.0	7.0
SqueezeNet	32	8.1	6.0
	16	3.1	2.5
Mini AlexNet	32	17.0	9.0
	16	7.5	4.5
YOLO Tiny	32	19.1	11.4
	16	12.0	5.2
YOLO	32	115.2	63.0
	16	50.4	27.0

OBJECT DETECTION DNN

Modified version YOLO (You Only Look Once) DNN

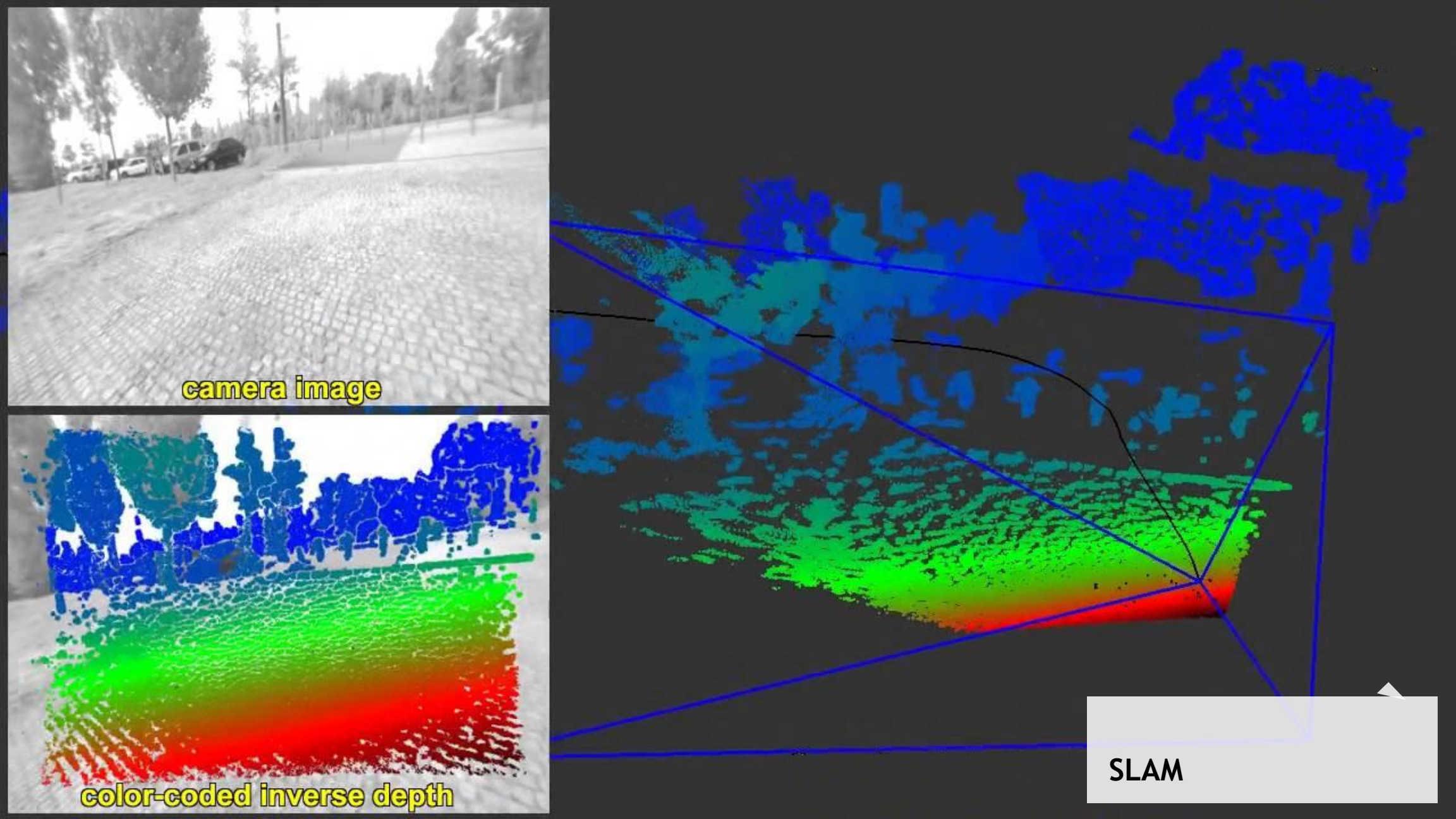
Replaced Leaky ReLU with ReLU

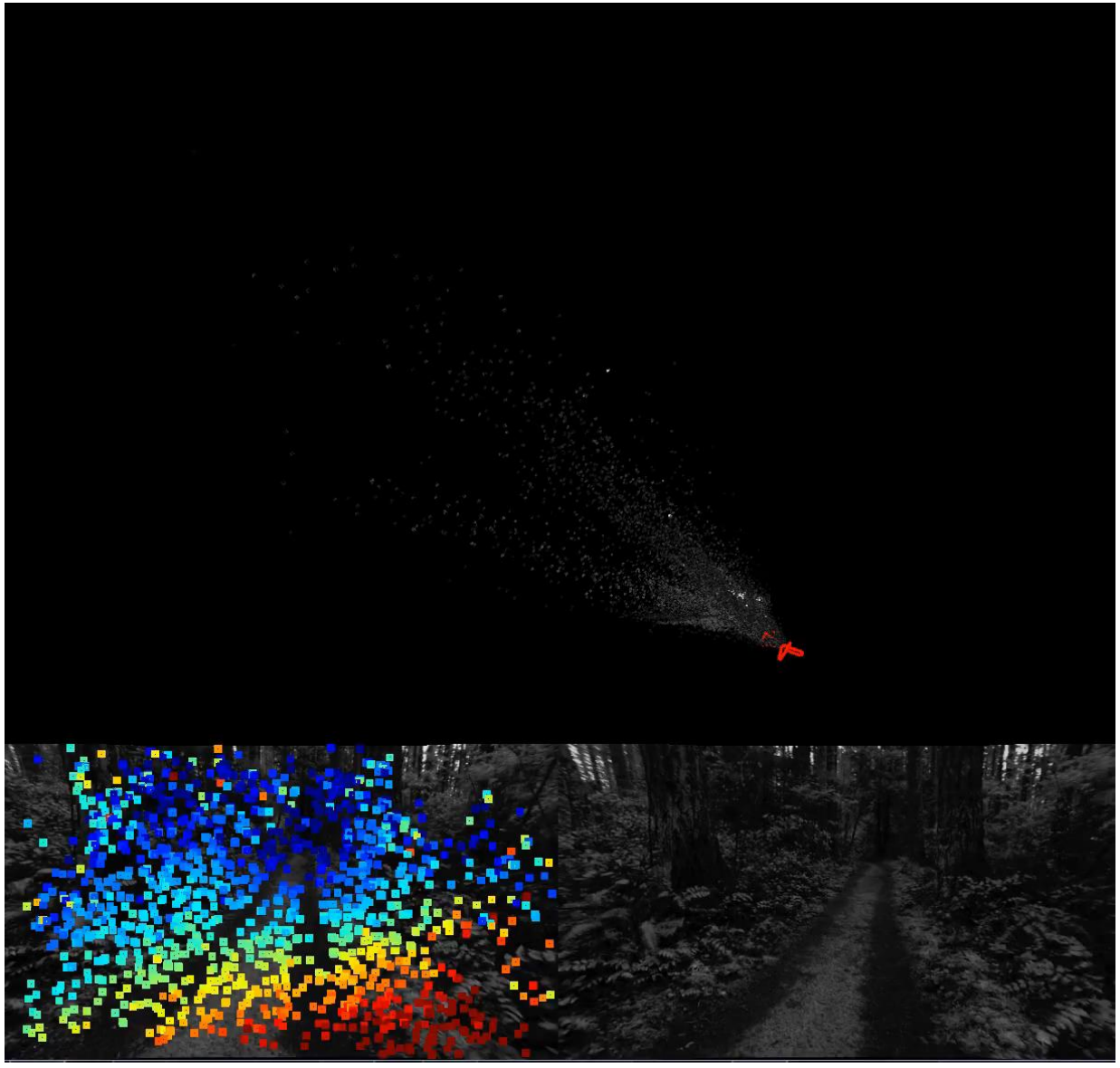
Trained using darknet then converted to Caffe model

TrailNet and YOLO are running simultaneously in real time on Jetson

J. Redmon et al. 2016

THE NEED FOR OBSTACLE AVOIDANCE


camera image
color-coded inverse depth
SLAM



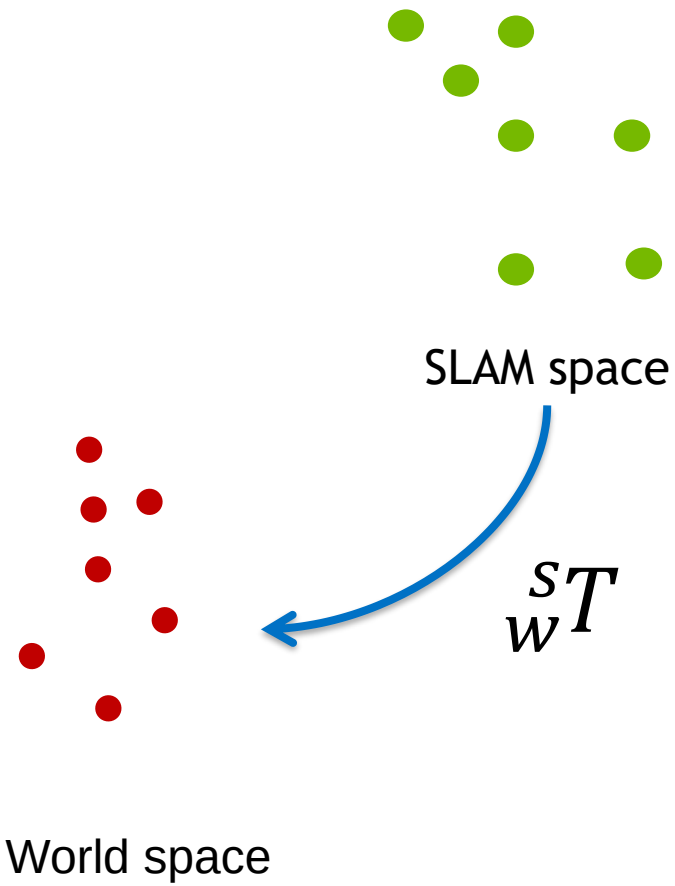
SLAM RESULTS

PROCRUSTES ALGORITHM

Find the transform

Aligns two correlated point clouds

Gives us real-world scale SLAM data



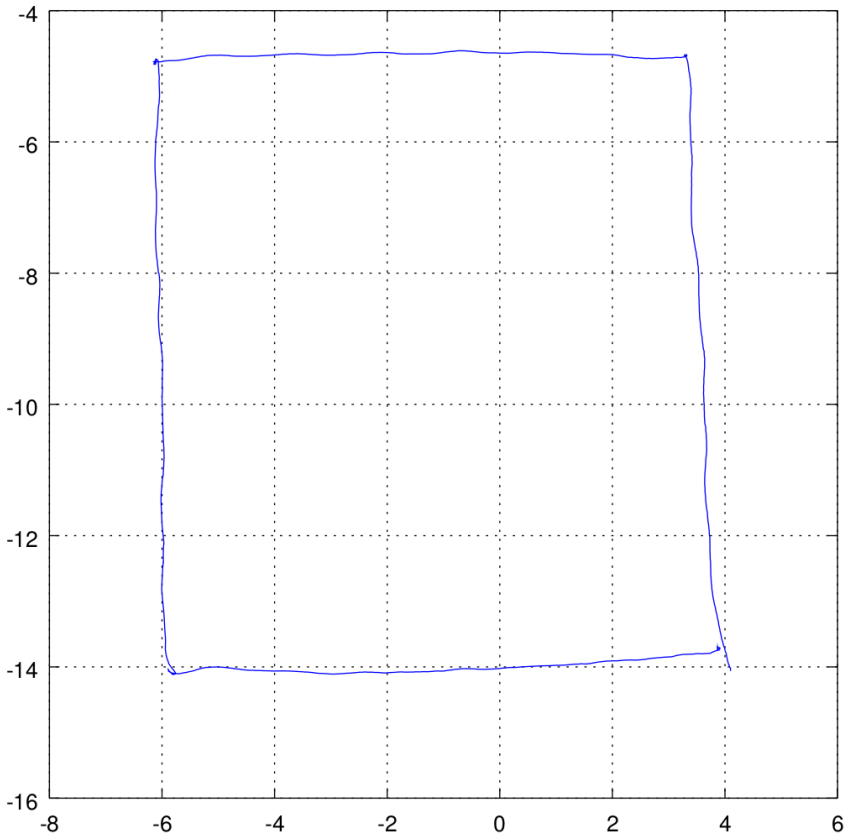
PIXHAWK VISUAL ODOMETRY

Estimating error

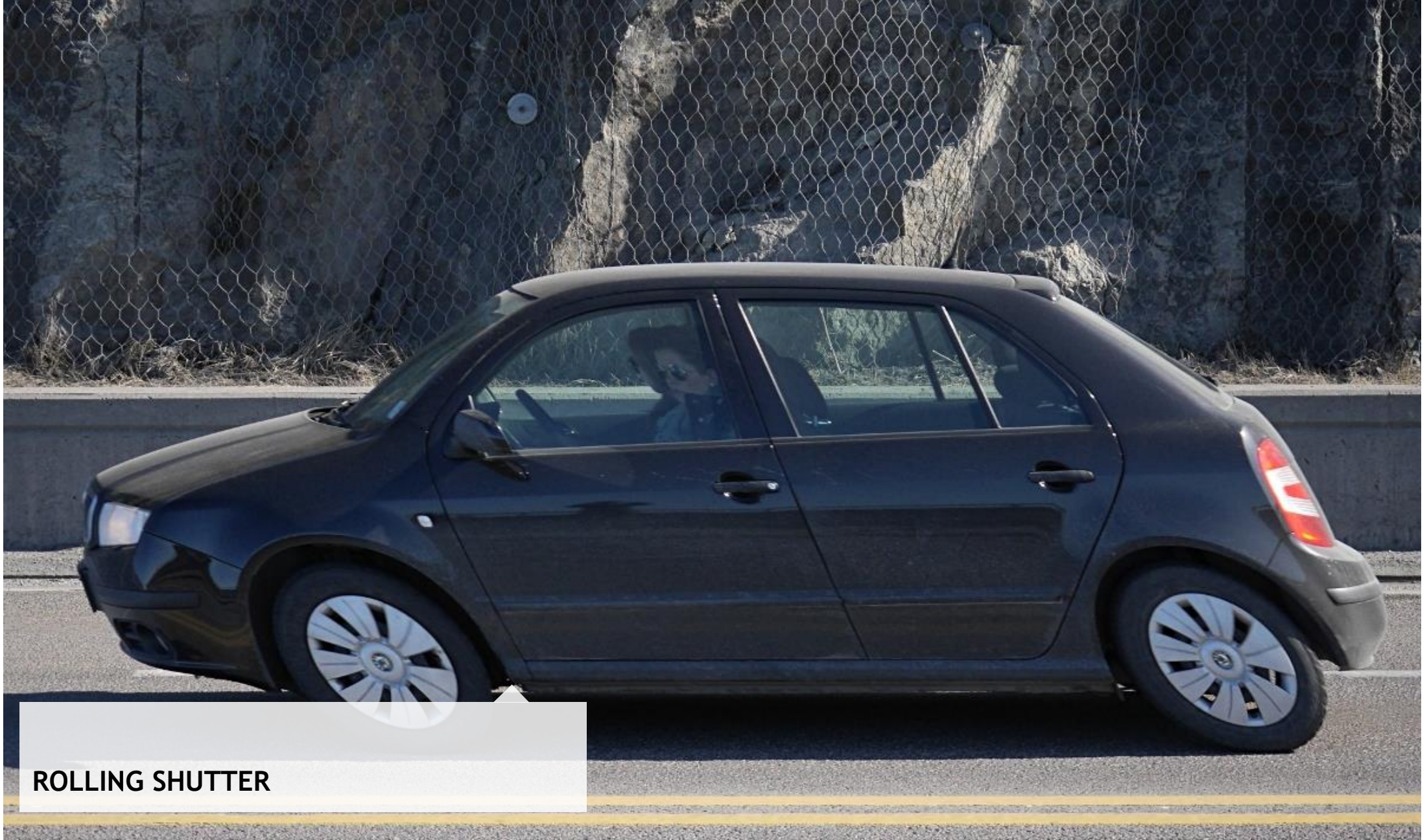
Optical flow sensor PX4FLOW

Single-point LIDAR for height

Gives 10-20% error in pose estimation



PX4 pose from flight in 10m square



ROLLING SHUTTER

SLAM FOR ROLLING SHUTTER CAMERAS

Solve for camera pose for **each scanline**

Run time is an issue

2x - 4x slower than competing algorithms

Direct Semi-dense SLAM for Rolling Shutter Cameras (J.H. Kim, C. Cadena, I. Reid)
In IEEE International Conference on Robotics and Automation, ICRA 2016

SEMI-DENSE MAP COMPUTE TIMES ON JETSON

	TX1 CPU USAGE	TX1 FPS	TX2 CPU	TX2 FPS
DSO	3 cores @ ~60%	1.9	3 cores @ ~65%	4.1
RRD-SLAM	3 cores @ ~80%	0.2	3 cores @ ~80%	0.35

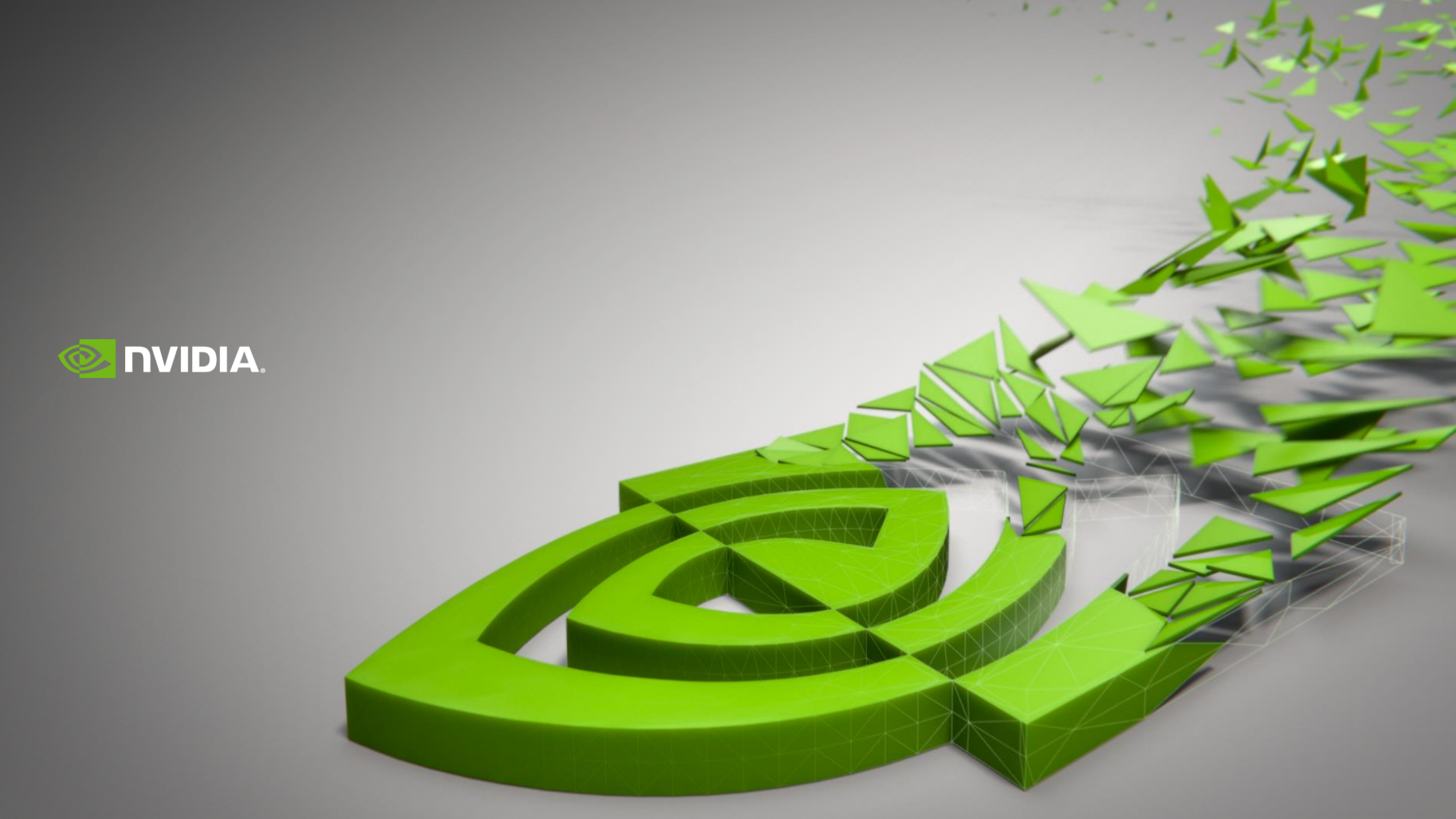
CONCLUSIONS. FUTURE WORK

We achieved 1 km forest flights with semantic DNN

Accurate depth maps are needed to avoid unexpected obstacles

Visual SLAM can replace optical flow in visual-inertial stabilization

Safe reinforcement learning can be used for optimal control


NVIDIA