Lab Course Cognitive Systems: Depth & Bounding Box Prediction

Final Presentation

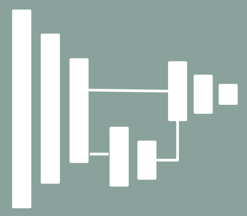
WS 17/18

S. Aklanoglu, J. Schuck, Y. El himer, F. Retkowski

Motivation

1st STAGE

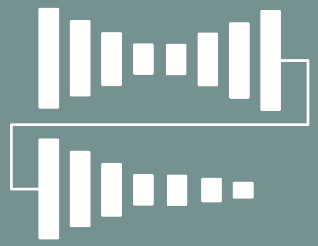
Goal
Background



DeepTLR

2nd STAGE

Caffe
FZI



FCRN-SSD

3rd STAGE

ResNet50-Deconv +
VGG16-SSD

FCRN-DSSD

4th STAGE

ResNet50-Deconv-SSD-
Deconv

Evaluation

5th STAGE

Results
Challenges
Solutions

Future Work

6th STAGE

Improvements

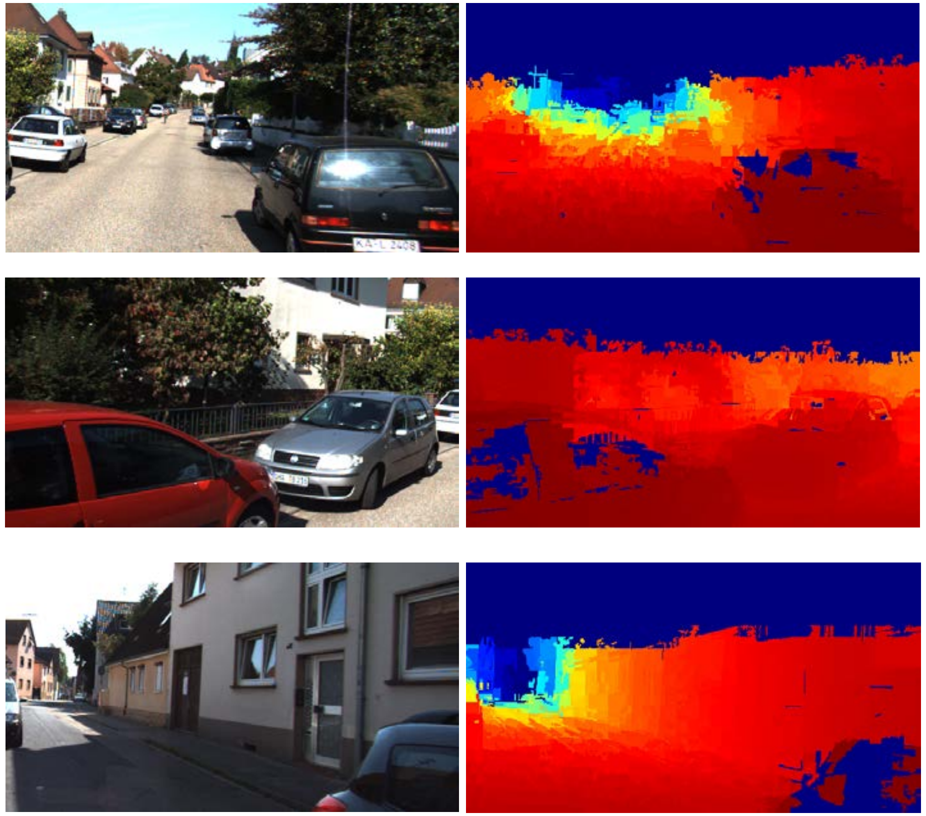
1. Motivation

Goal: Improve vehicle detections with CNN in real-time

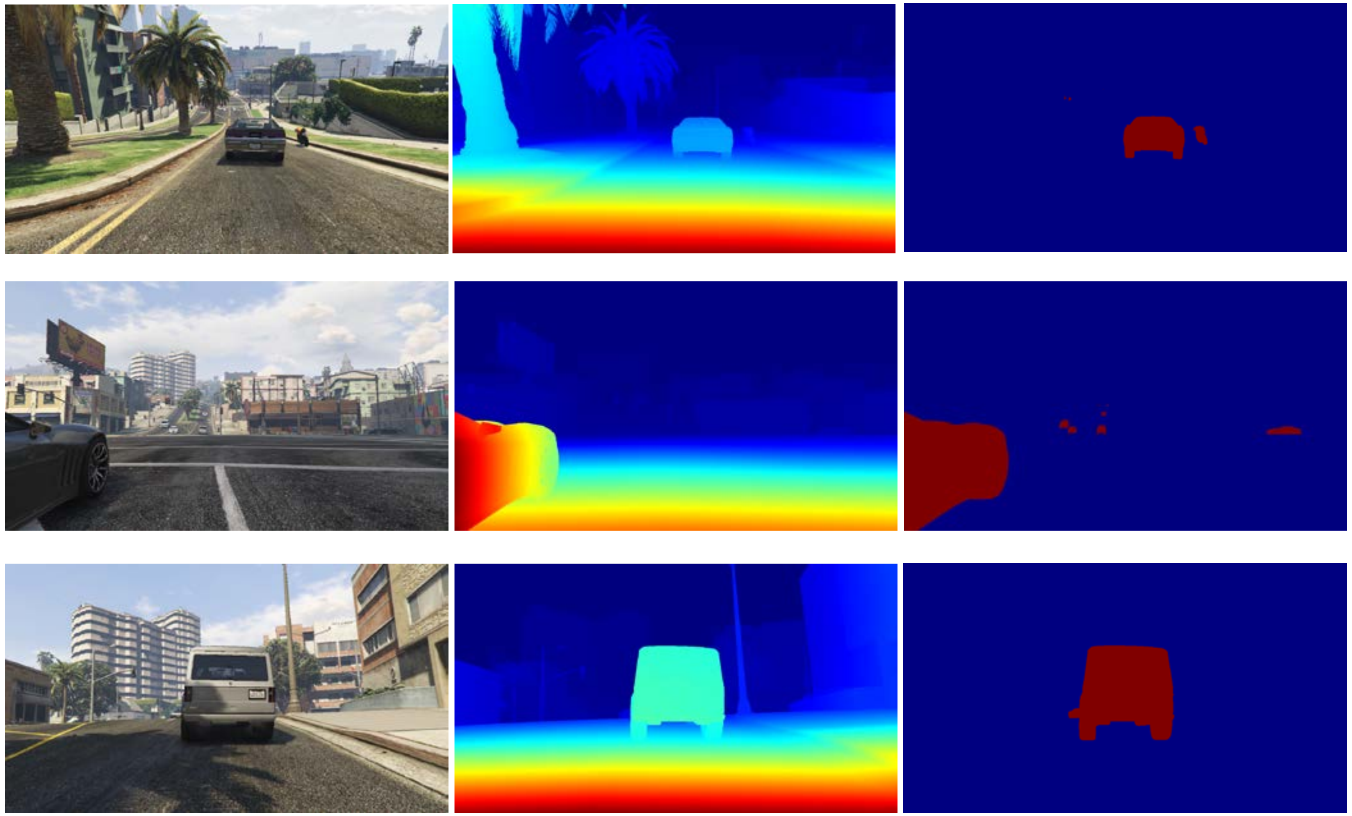
- Issues: poor performance for largely overlapping vehicles
- Possible solution: using depth information to distinguish between overlapping vehicles
- Evaluation of synthetic GTA V training data

1. Motivation

Used datasets



KITTI 15k, 640 x 352, uint8/16 depth map
54205 boxes



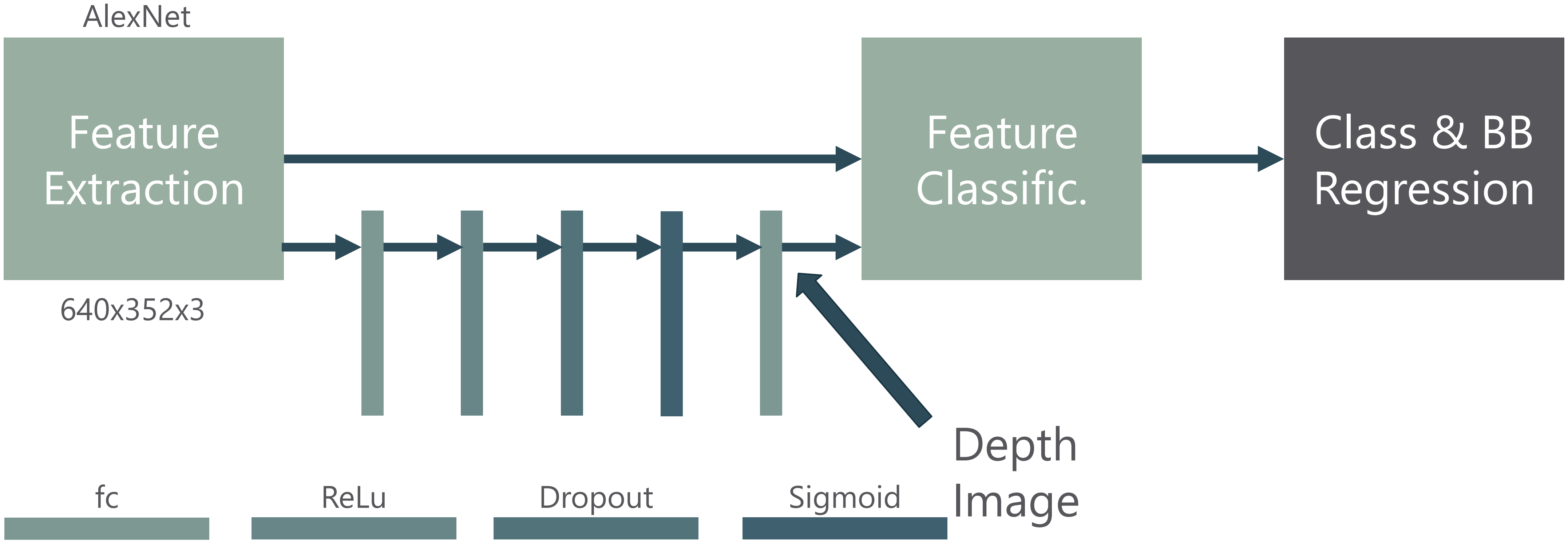
GTA V 16k, 1280 x 720, uint8 depth map
80364 boxes

2. DeepTLR+Conv

Learning from depth images for monocular object detection with convolutional neural networks, MA A. Lesi

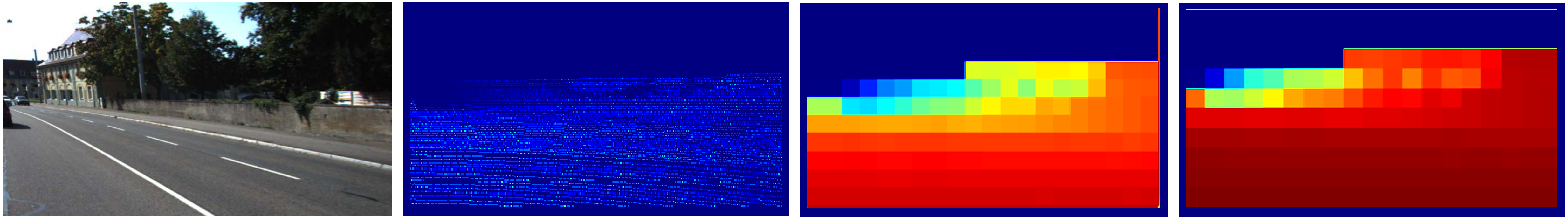
Joint convolution depth prediction and object detection:

- Training with KITTI & GTA V images



2. DeepTLR+Conv

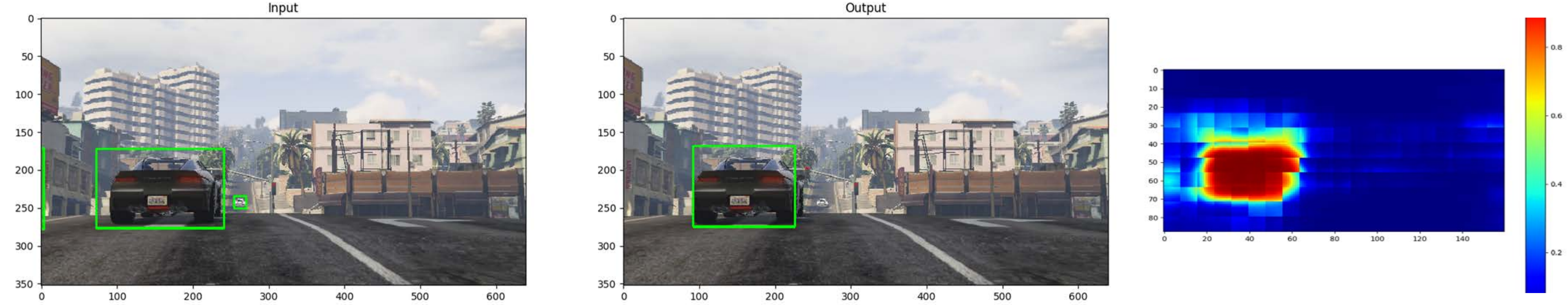
Current results: Depth map prediction



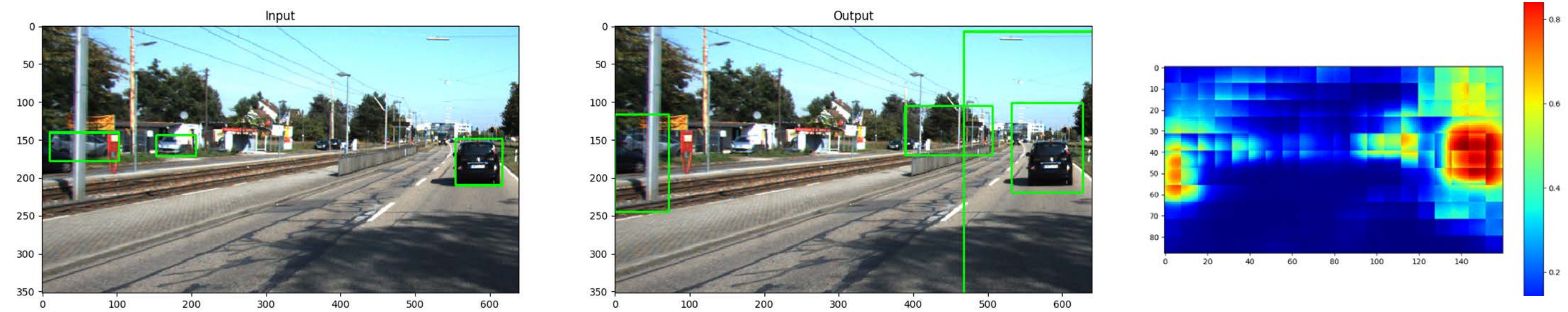
2. DeepTLR+Conv

Current results: Model trained on GTA V

Inference on GTA V instance

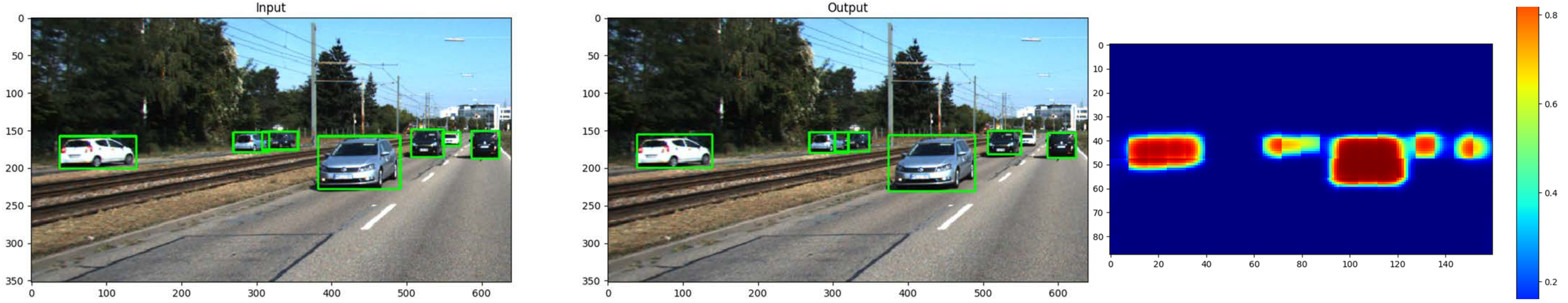


Inference on KITTI instance

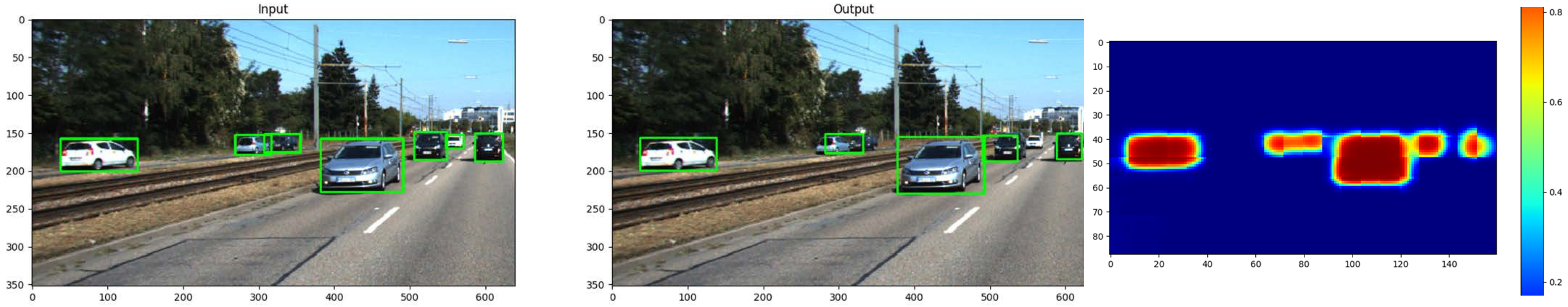


2. DeepTLR+Conv

Current results: Model trained on KITTI + GTA V



Compared to the Model trained only on KITTI



2. DeepTLR+Conv

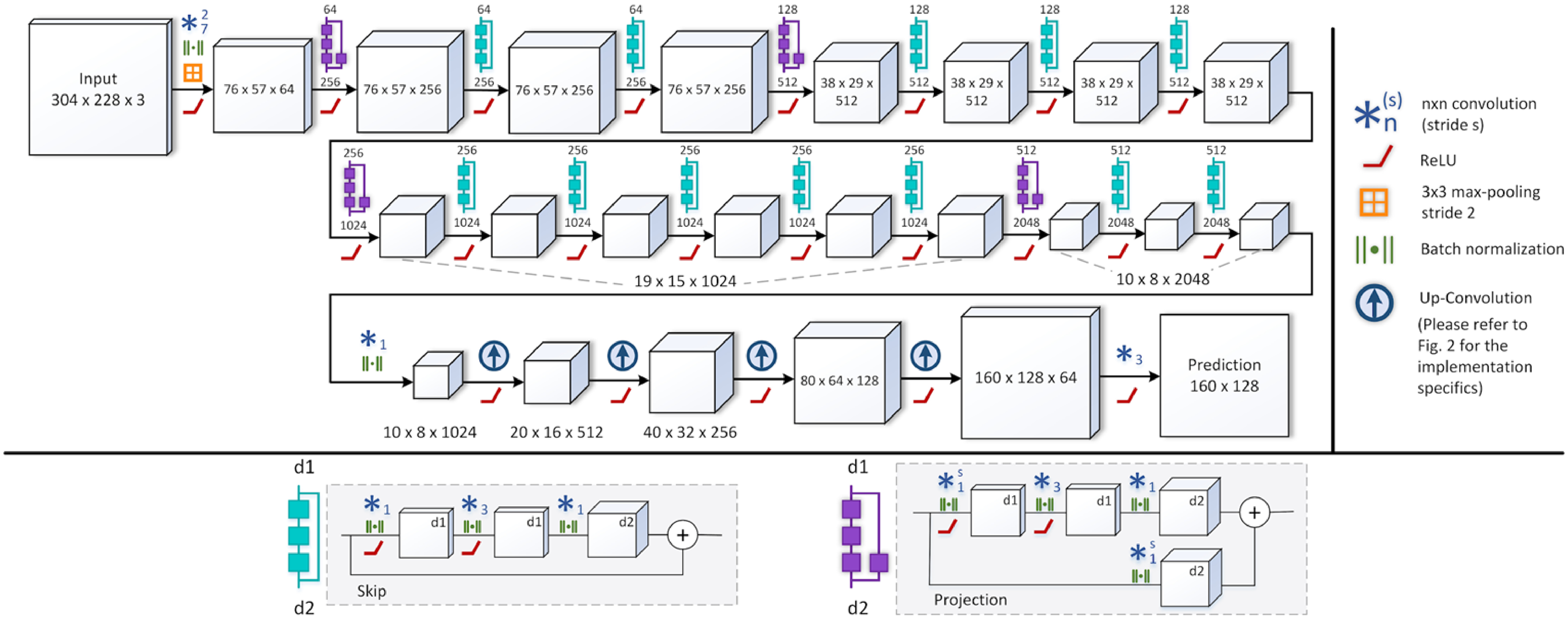
Evaluation:

	Trained on KITTI + GTA V		Trained on GTA V	
	Evaluated on KITTI	Evaluated on GTA V		Evaluated on KITTI
REL	**0.318767**	**-**	**0.317722**	**-**
RMSE	**109.07**	**-**	**108.68**	**189.00**
LOG10	**0.100764**	**-**	**0.0999771**	**-**
d1	**0.622377**	**-**	**0.627336**	**0.0165936**
d2	**0.92577**	**-**	**0.927262**	**0.057907**
d3	**0.958095**	**-**	**0.95856**	**0.170823**

3. **FCRN**-SSD

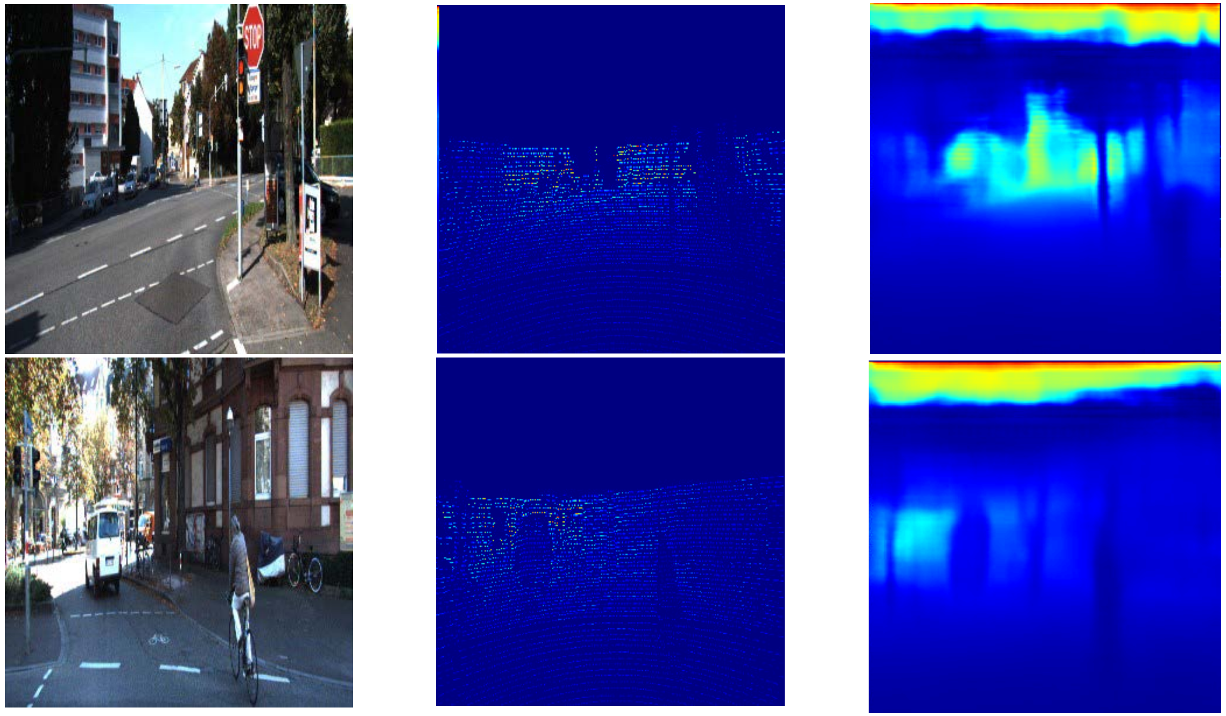
Deeper Depth Prediction with Fully Convolutional Residual Networks, Laina et al.
\+ SSD: Single Shot MultiBox Detector, Liu et al.

ResNet50 (fully convolution mode) and deconvolution layers:



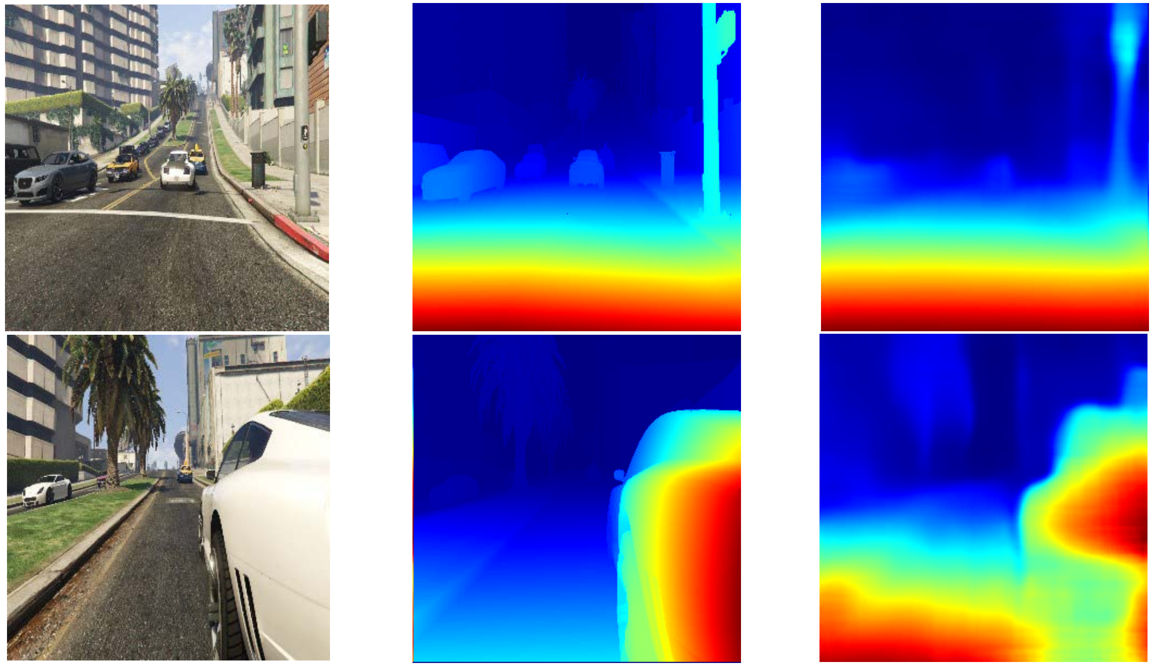
3. **FCRN**-SSD

Current results: Model trained on KITTI, inferenced on KITTI testset



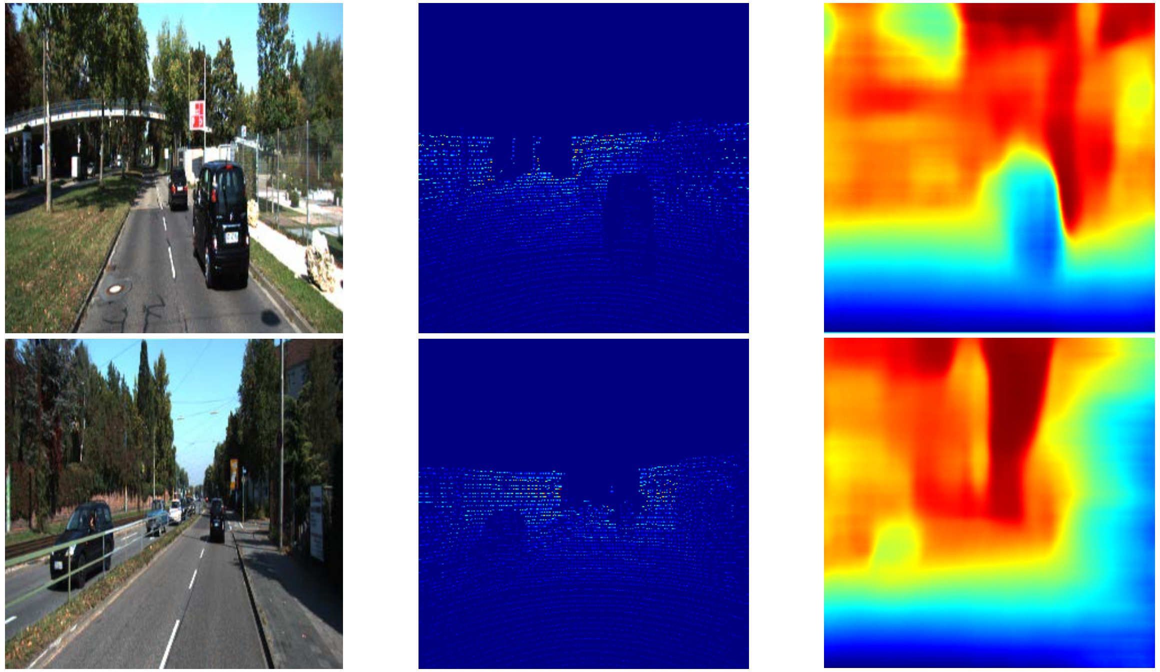
3. **FCRN**-SSD

Current results: Model trained on GTA V, inferenced on GTA V testset



3. **FCRN**-SSD

Current results: Model trained on GTA V, inferenced on KITTI testset



3. **FCRN**-SSD

Evaluation:

	Trained on KITTI		Trained on GTA V	
	Evaluated on KITTI testset	Evaluated on GTA V testset		Evaluated on KITTI testset
REL	**-**	**-**	**-**	**-**
RMSE	**13.78**	**75.86**	**15.60**	**156.47**
LOG10	**2.02**	**2.03**	**0.48**	**4.27**
d1	**0.04**	**0.01**	**0.65**	**0.002**
d2	**0.09**	**0.02**	**0.80**	**0.004**
d3	**0.17**	**0.03**	**0.88**	**0.007**

3. **FCRN-SSD**

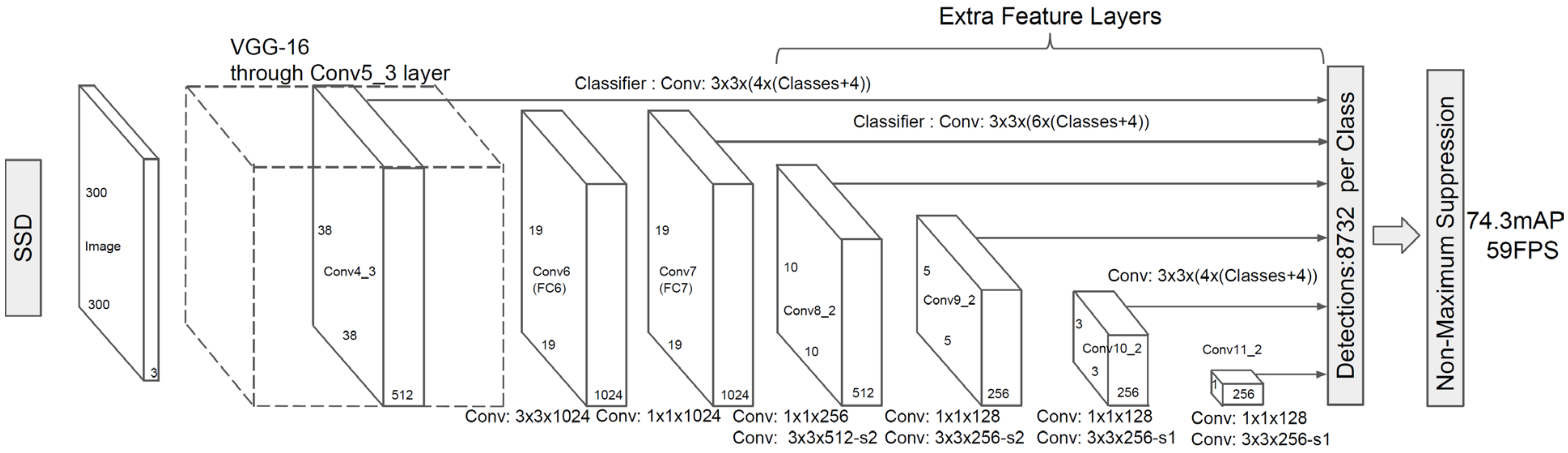
- We chose the FCRN approach because of superior depth prediction performance
- Deconvolutional network originally used for semantic segmentation did not train (with RMSE or Huber loss) and remained noisy
- Next step: combining with bounding box prediction

- Advantages of two isolated neural nets:
  - Stable depth prediction
  - Stable bounding box detection
  - In multitask learning often hard to get loss function right
  - Easier/better evaluation of GTA V data with two distinct networks

3. FCRN-**SSD**

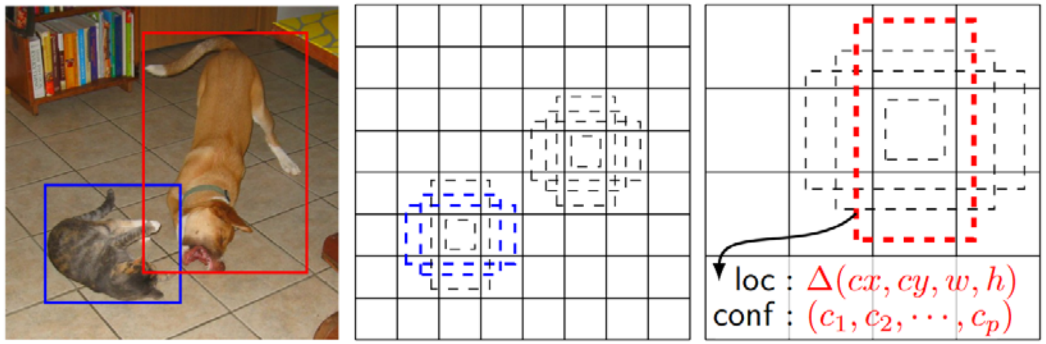
Deeper Depth Prediction with Fully Convolutional Residual Networks, Laina et al.
\+ SSD: Single Shot MultiBox Detector, Liu et al.

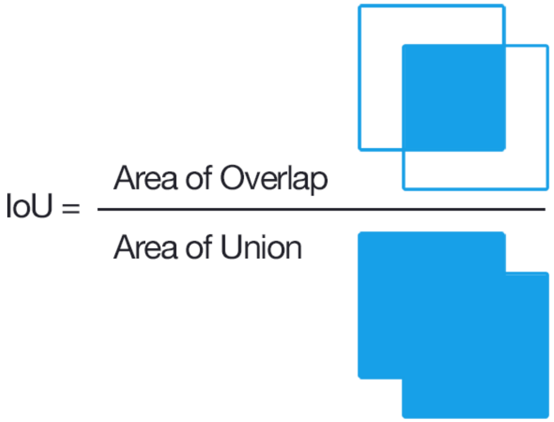
VGG16 (without fully connected layers) with auxillary SSD layers:

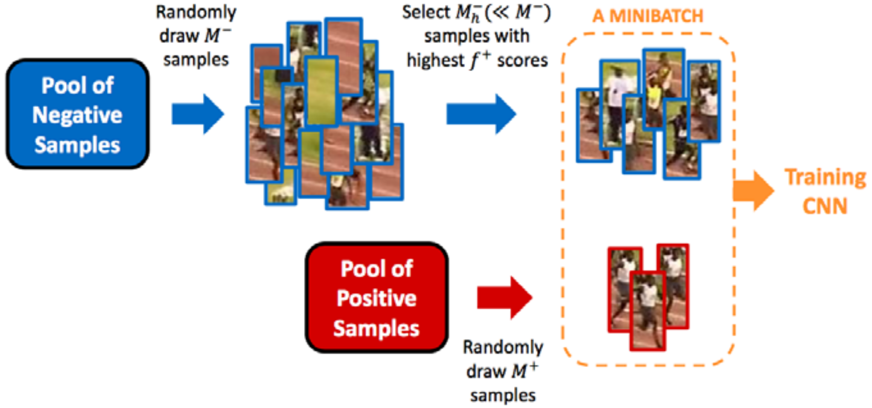


3. FCRN-**SSD**

Multibox approach:

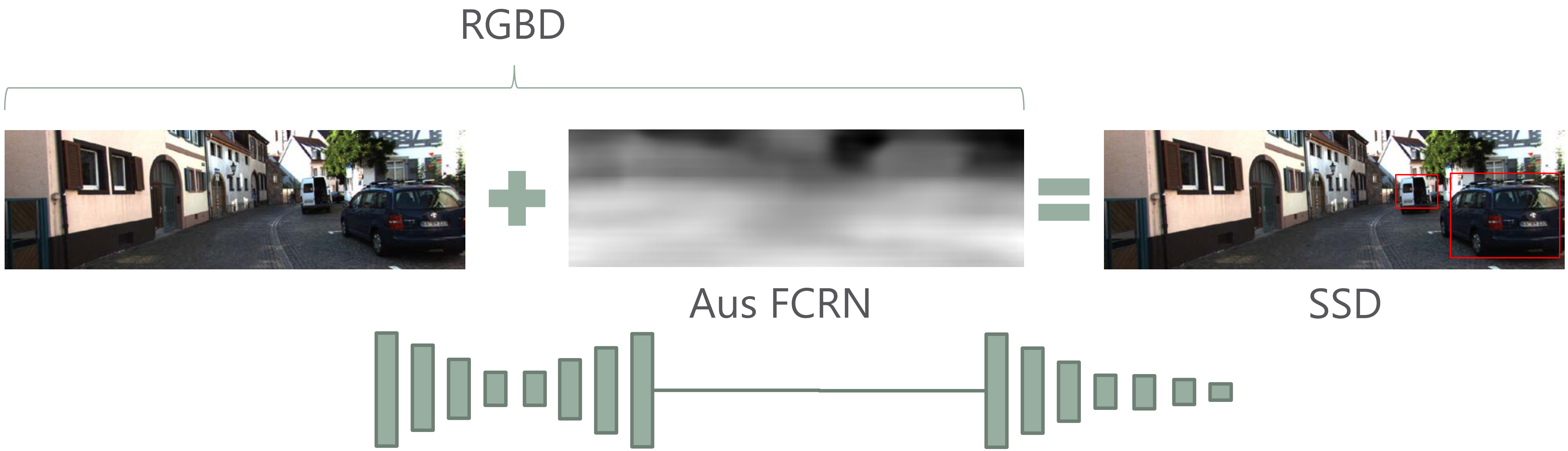






- Boxes will be estimated on multiple feature layers with different feature map sizes and default boxes
- Hard negative mining with ratio of negative to positive examples of around 3:1

3. FCRN-**SSD**



3. FCRN-**SSD**

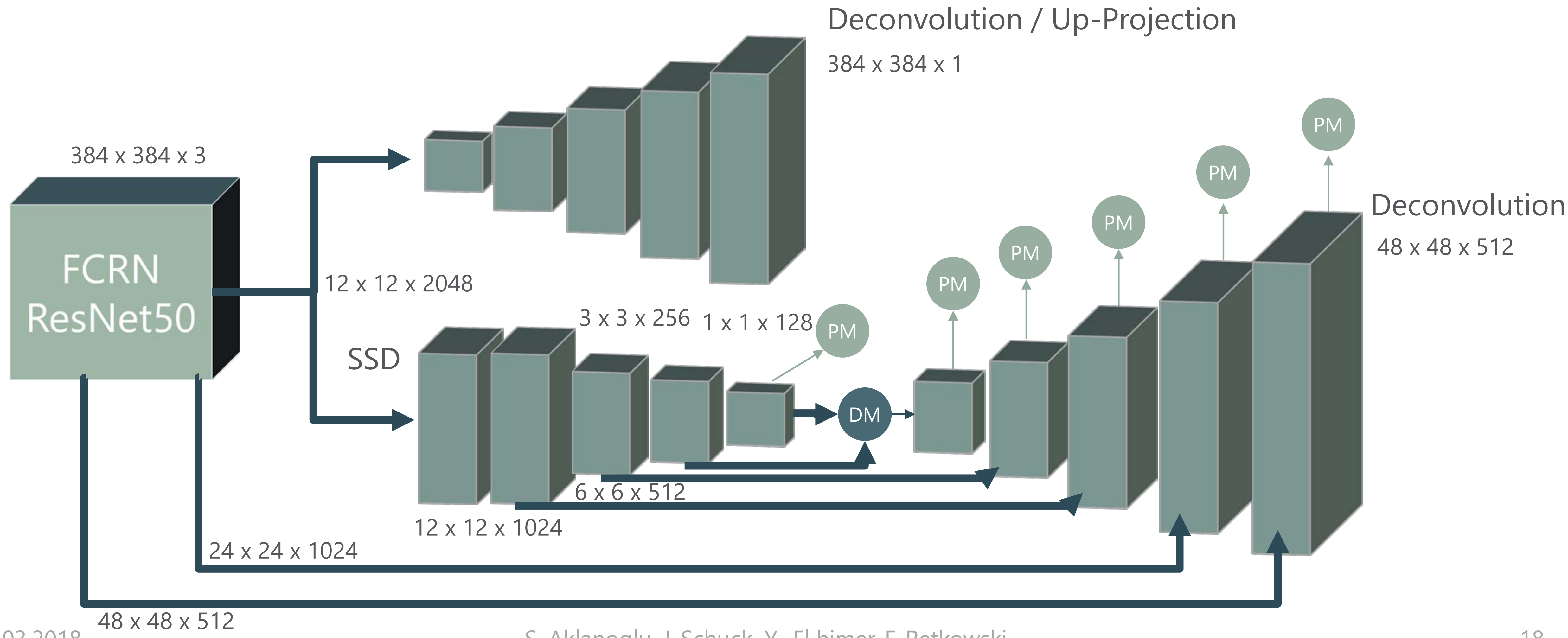
Evaluation: Model trained on KITTI, inferenced on KITTI testset

	Trained on KITTI with GTA V Depth Channels
	Evaluated on KITTI testset
mAP	**0.25**

4. FCRN-DSSD

Deeper Depth Prediction with Fully Convolutional Residual Networks, Laina et al.
+ SSD: Single Shot MultiBox Detector, Liu et al.
+ DSSD : Deconvolutional Single Shot Detector Fu et al.

End-to-end depth and bounding box prediction:



4. FCRN-DSSD

Image Preprocessing:

- Online image augmentation in Tensorflow
  - Bounding box adjustments
  - Random horizontal flip
  - Random color distortion
  - Patch sampling

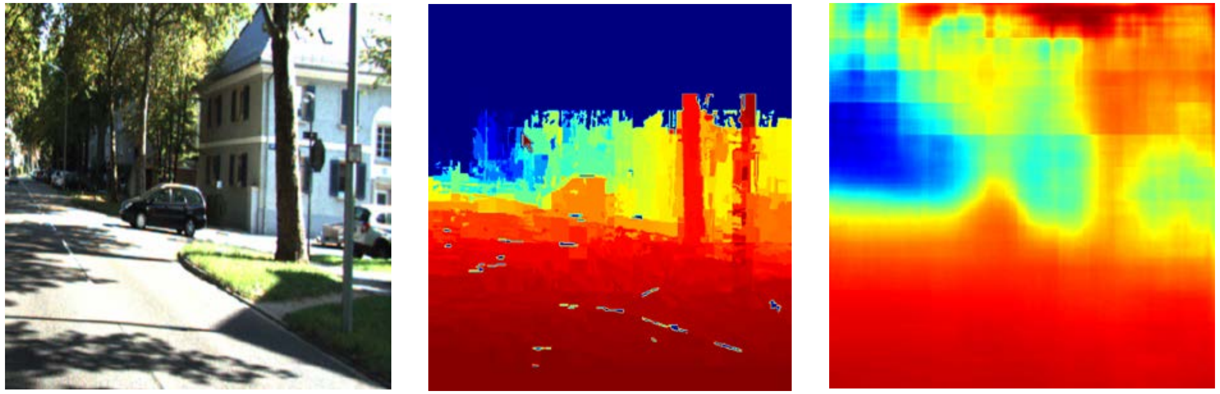
1280 x 720 x 3

384 x 384 x 3

4. FCRN-DSSD

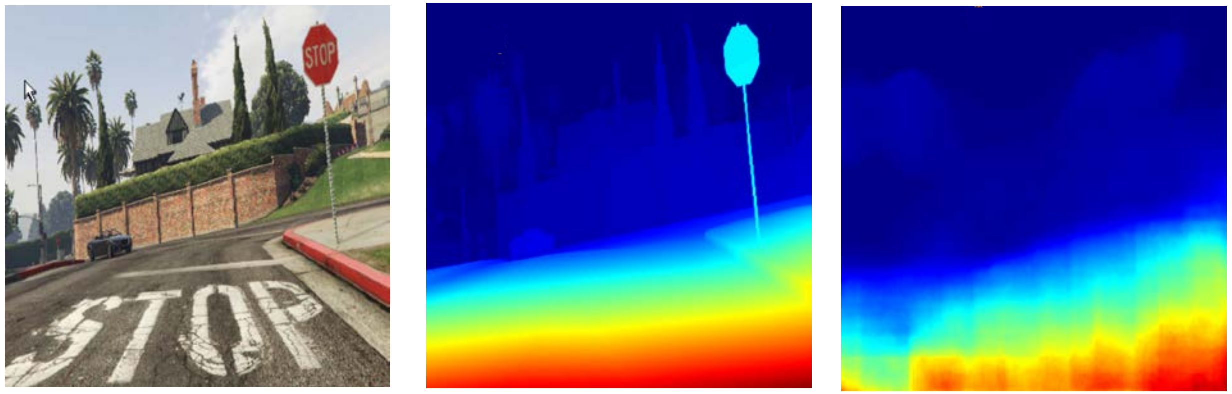
Current results:

KITTI



Model trained on KITTI, inferenced on KITTI testset

GTA V

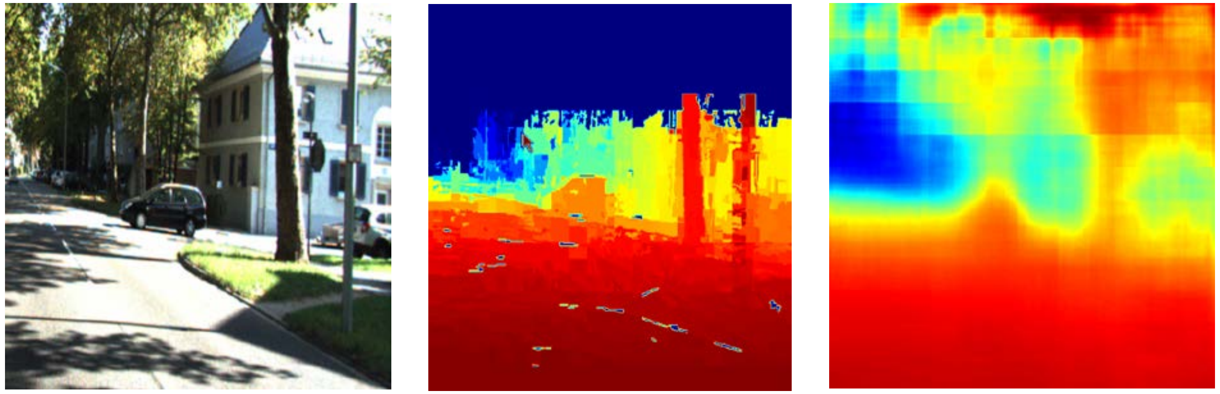


Model trained on GTA V, inferenced on GTA V testset

4. FCRN-DSSD

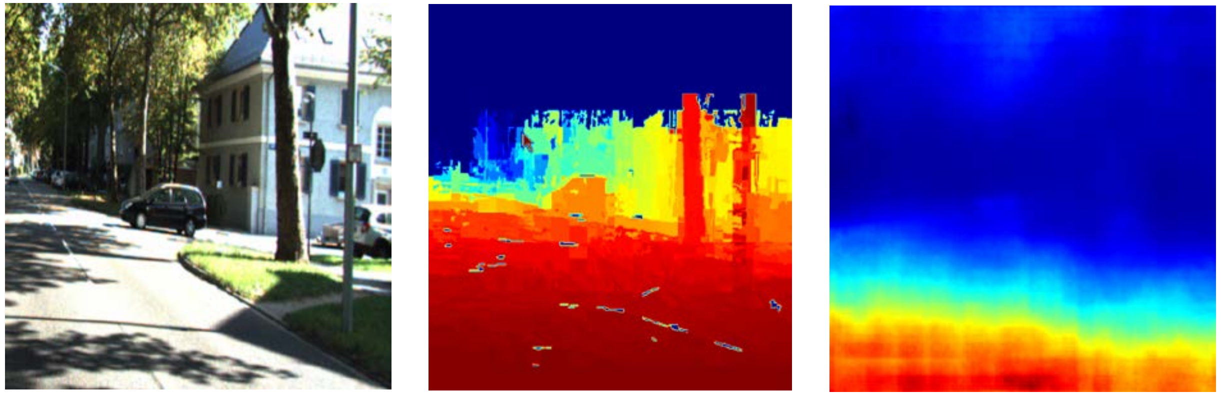
Current results:

KITTI



Model trained on KITTI, inferenced on KITTI testset

KITTI



Model trained on GTA V, inferenced on KITTI testset

4. FCRN-DSSD

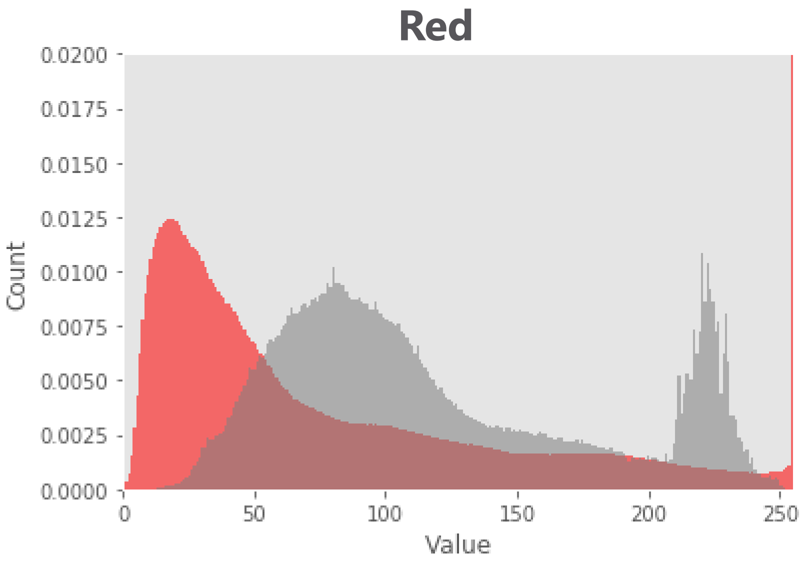
Evaluation:

	Trained on KITTI		Trained on GTA V	
	Evaluated on KITTI testset	Evaluated on GTA V testset		Evaluated on KITTI testset
REL	**199.94**	**188.51**	**65.82**	**62.57**
RMSE	**205.92**	**195.99**	**92.71**	**81.22**
LOG10	**-**	**-**	**5.32**	**-**
d1	**-**	**-**	**-**	**-**
d2	**-**	**-**	**-**	**-**
d3	**-**	**-**	**-**	**-**
mAP	**0.37**	**0.07**	**0.13**	**0.17**

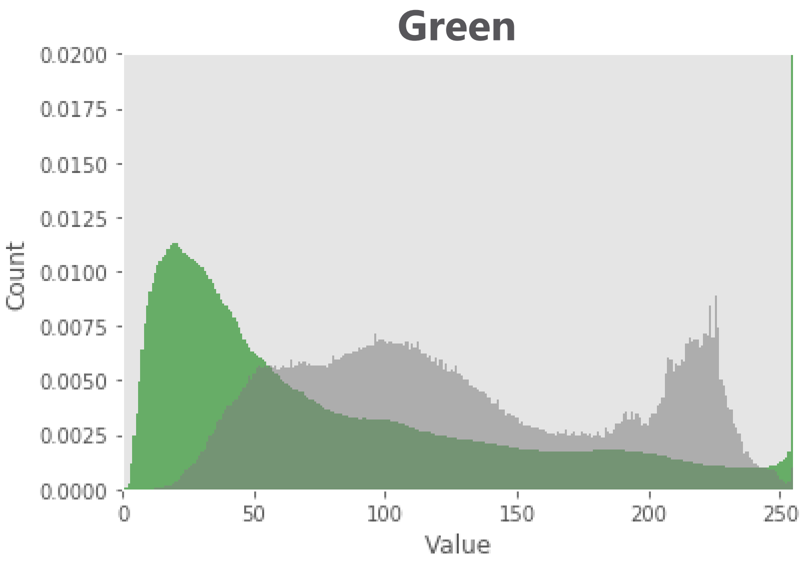
- training is slow ~ 6 days, inference ~1s per batch → 62,5 ms – 16 Hz

5. Evaluation

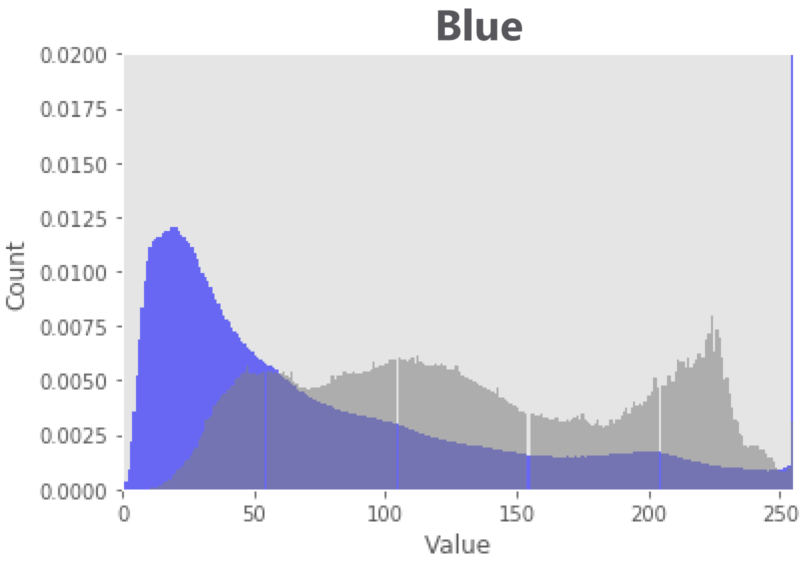
RGB ranges:



Red: KITTI



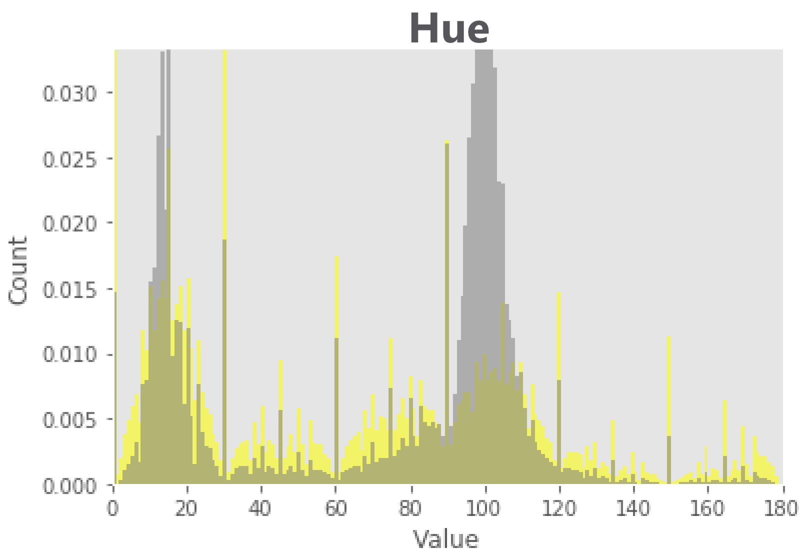
Green: KITTI



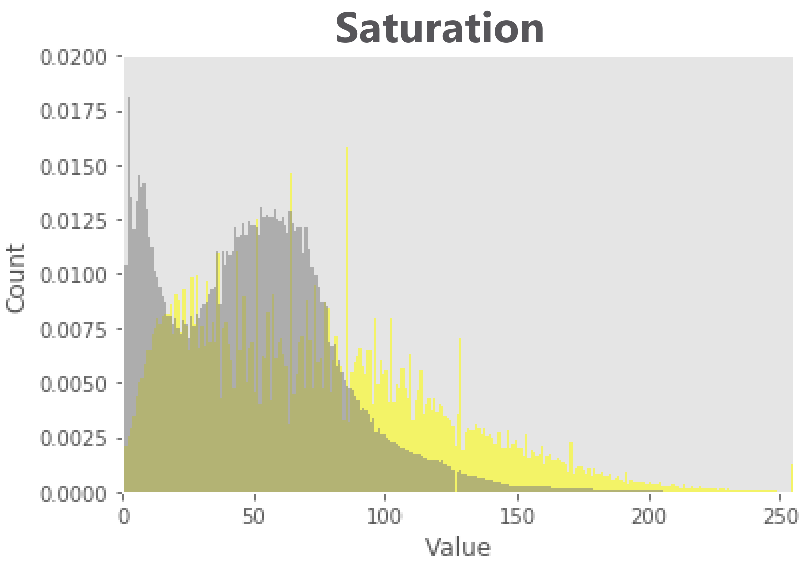
Blue: KITTI

5. Evaluation

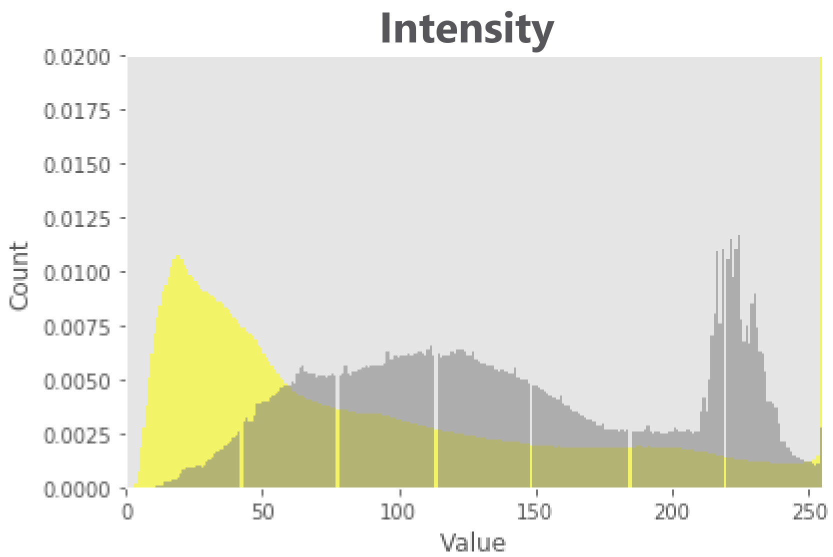
HSI ranges:



Yellow: KITTI



Yellow: KITTI

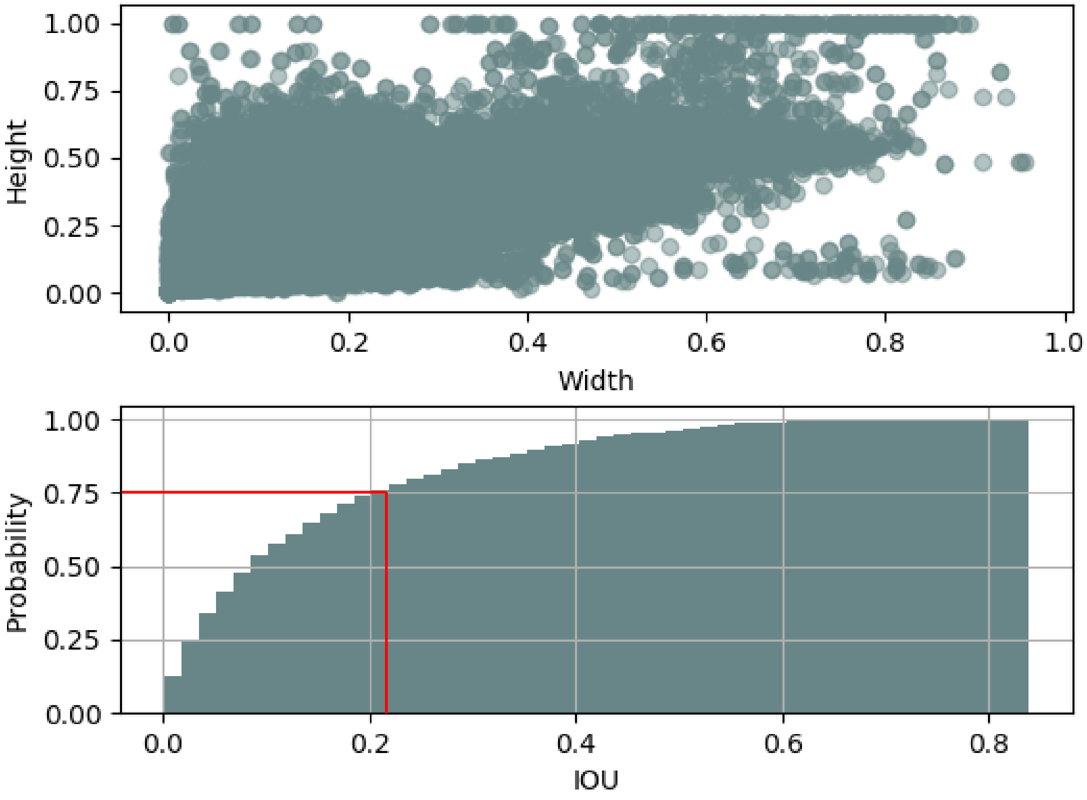


Yellow: KITTI

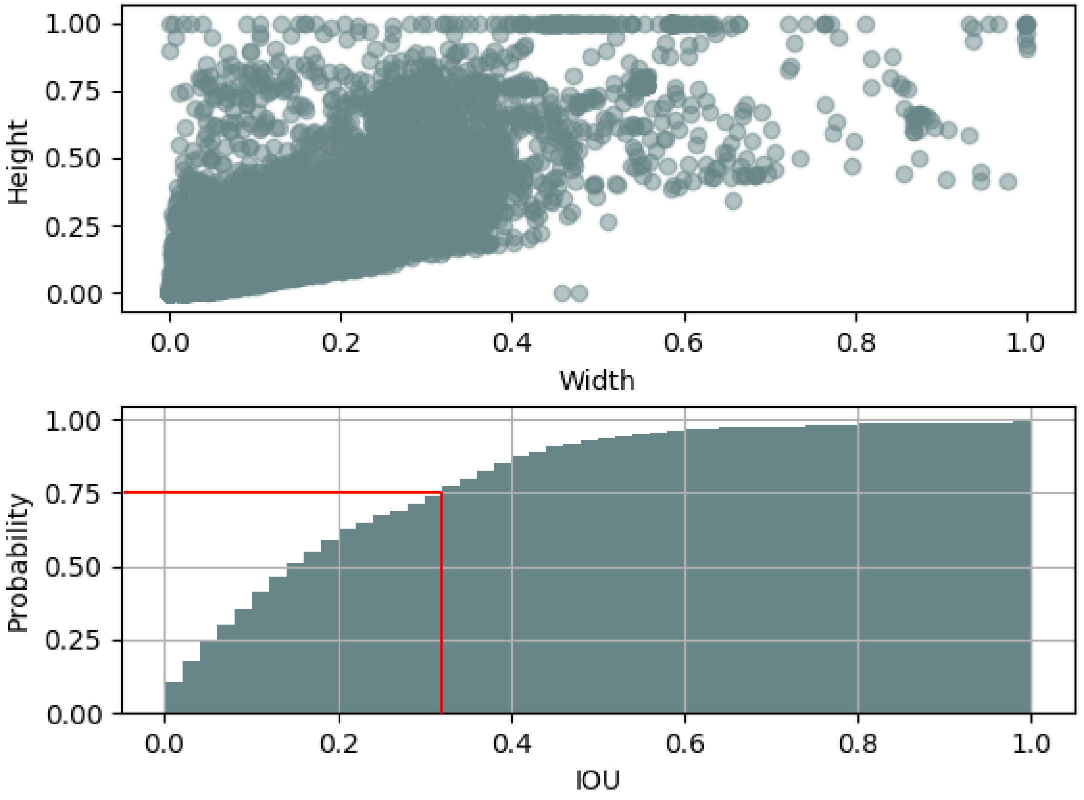
5. Evaluation

Bounding box distributions:

KITTI



GTA V

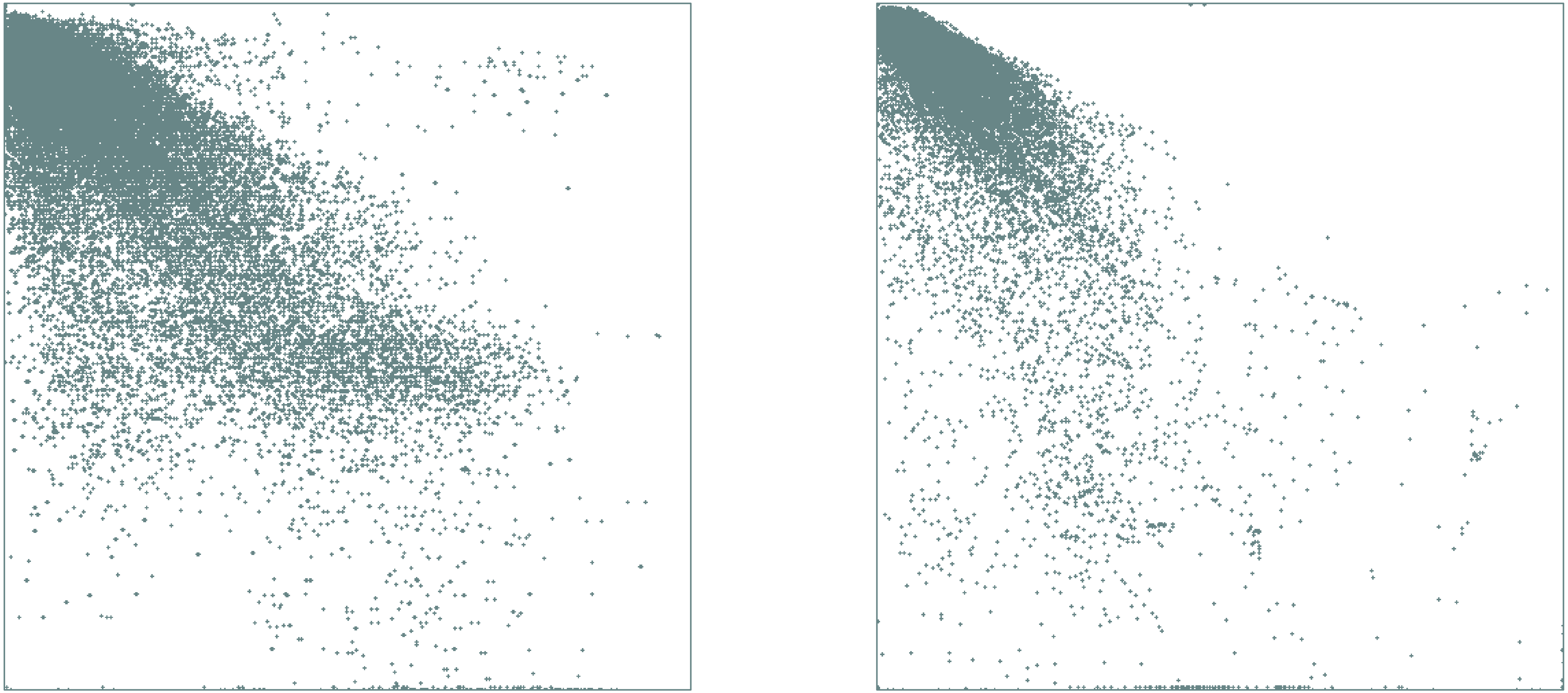


5. Evaluation

Bounding box positions:

KITTI

GTA V



5. Evaluation

Summary:

- Depth prediction trained on GTA V data works good, due to “perfect” data with first 2 approaches, multitask learning doesn’t perform good
- Bounding box prediction with GTA V data is difficult and needs careful parameter choosing/tuning for all approaches
- Using loss weights with 4:1 or 8:1 (DSSD:FCRN) when multitask training on GTA V or 1:2 when training on KITTI
- GTA V trained networks perform poor on KITTI datasets

6. Future Work

Datasets:

→ Do offline preprocessing to handle ground truth bounding boxes out of range

→ Need to use more GTA V images ~ 200.000 images, since difference in images is very low

→ Adjust color ranges of GTA V data according KITTI and better in-game traffic flow control

→ Eliminate wrong bounding boxes due to insufficient occlusion handling in game – using stencil map for bounding box creation and matching with data needed

FCRN:

→ Using berHu loss instead of l2 norm (better results in paper), we already implemented it, but didn‘t train with

→ Combine KITTI and GTA datasets for training

SSD:

→ loss has problems to converge, especially for positive bounding box examples and localization; lots of small bounding boxes in GTA V data, better default box sizes, ratios and tiling needed

6. Future Work

FCRN-DSSD:

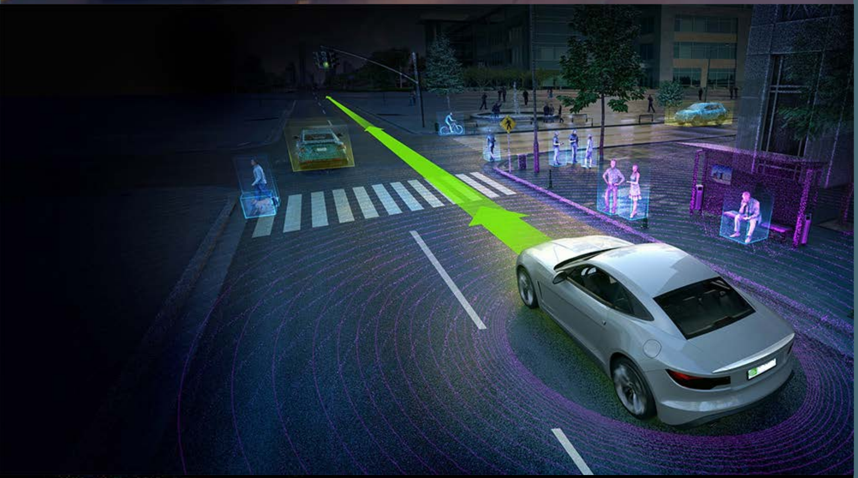
→Networks are in modular architecture, it’s easy to change certain parts/networks – trying other detectors

→Extend code to support multiple GPUs for faster training, since a lot of trial and error is needed to get optimal loss weights, default box sizes, ratios, tiling etc.

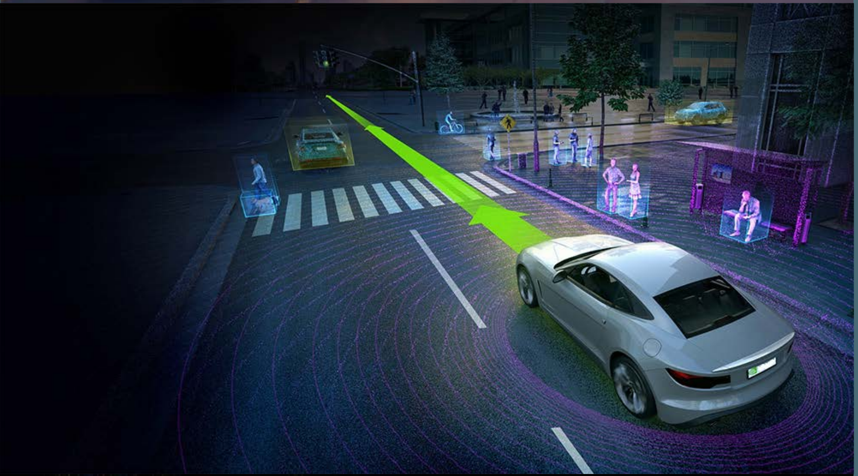
→If learning rate too high, SSD loss will oscillate, depth loss will converge fast

→If learning rate too low, depth loss will not converge, SSD loss will oscillate less

→Finding optimal weight for both loss function - Adaptive Loss Balancing



Questions?


Backup

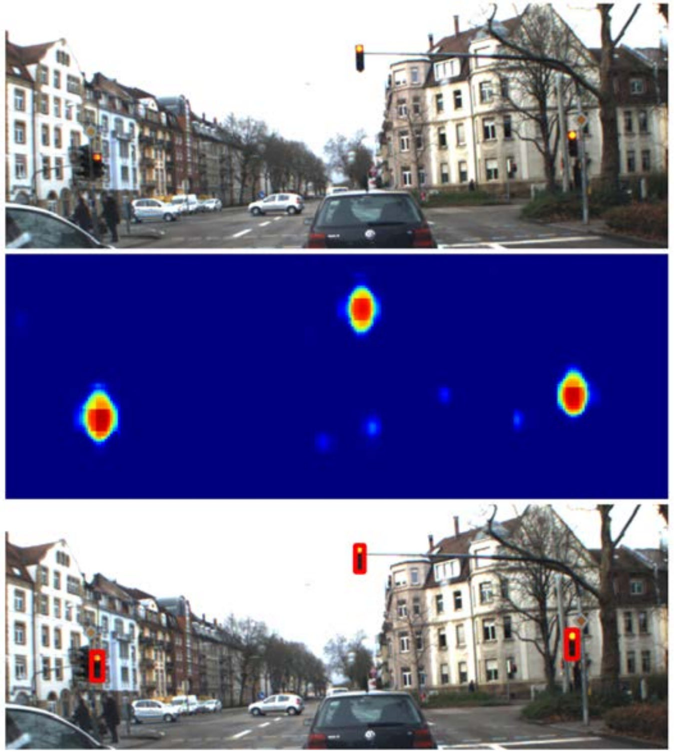
2. DeepTLR+Conv

Accomplished tasks:

- ✓ Caffe running
- ✓ Data processing & augmentation
- ✓ LMDB creation
- ✓ Training on KITTI
- ✓ Training on GTA V
- ✓ Training on mixed KITTI and GTA V
- ✓ Evaluation routines

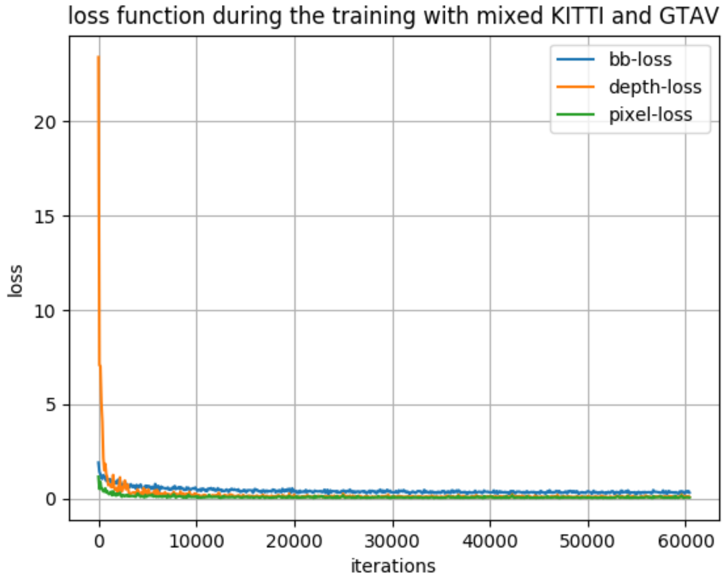
- → Using code from M. Weber and A. Lesi
- → No modifications on framework planned
- → Inference Speed: 20 – 45 ms

Caffe



2. DeepTLR+Conv

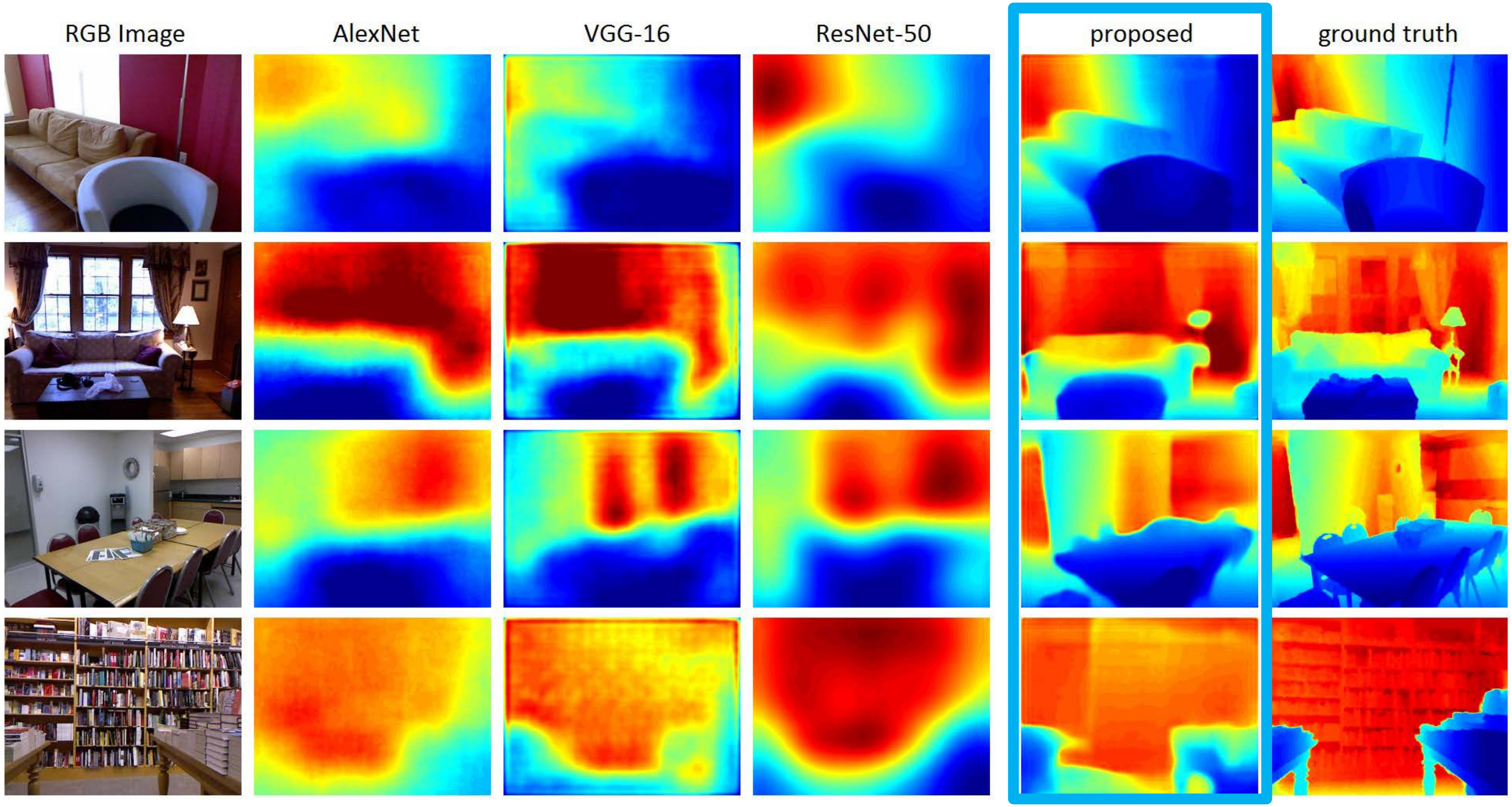
Loss during training:



3. FCRN-SSD

Deeper Depth Prediction with Fully Convolutional Residual Networks, Laina et al.
\+ SSD: Single Shot MultiBox Detector, Liu et al.

Pretrained weights trained on NYU v2 dataset:



3. **FCRN**-SSD

What was available with FCRN?

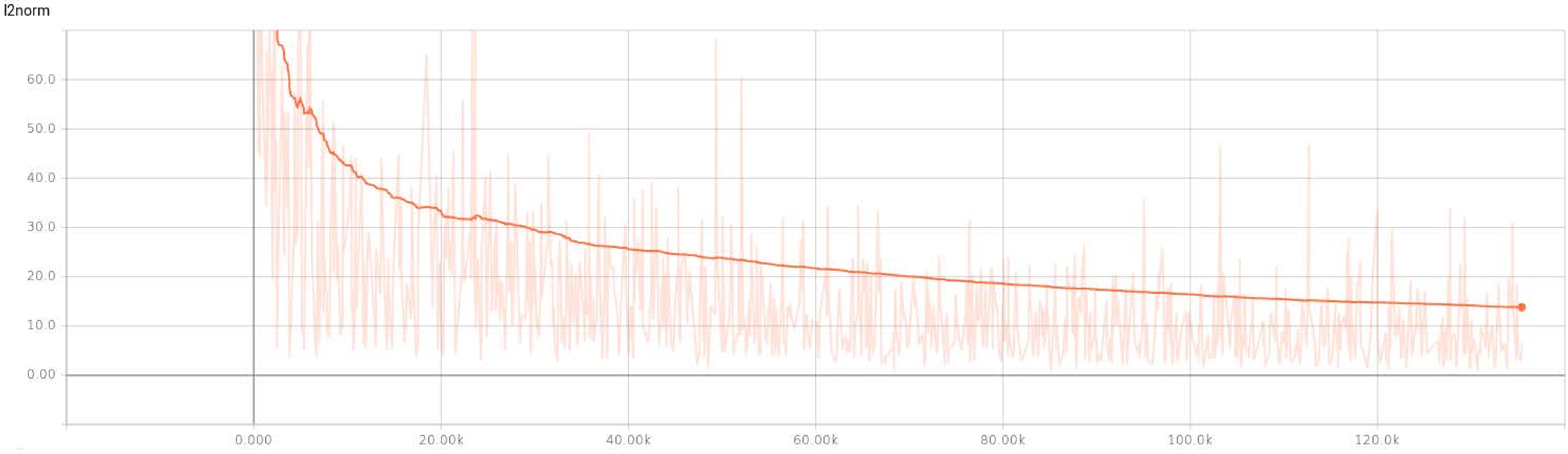
- Architecture in TensorFlow
- Evaluation Routines in Matlab

What have we done with FCRN?

- Architecture customized
- Data Processing
- Trainings Routines
- Training on KITTI data
- Training on GTA V data
- Evaluation Routines

3. **FCRN**-SSD

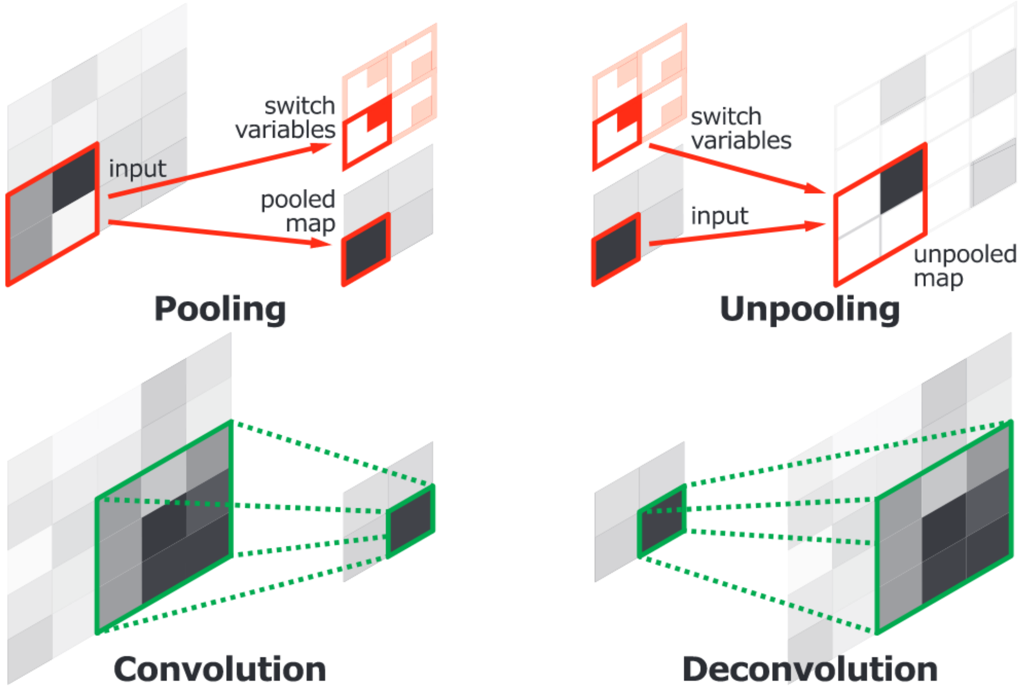
L2 training loss:

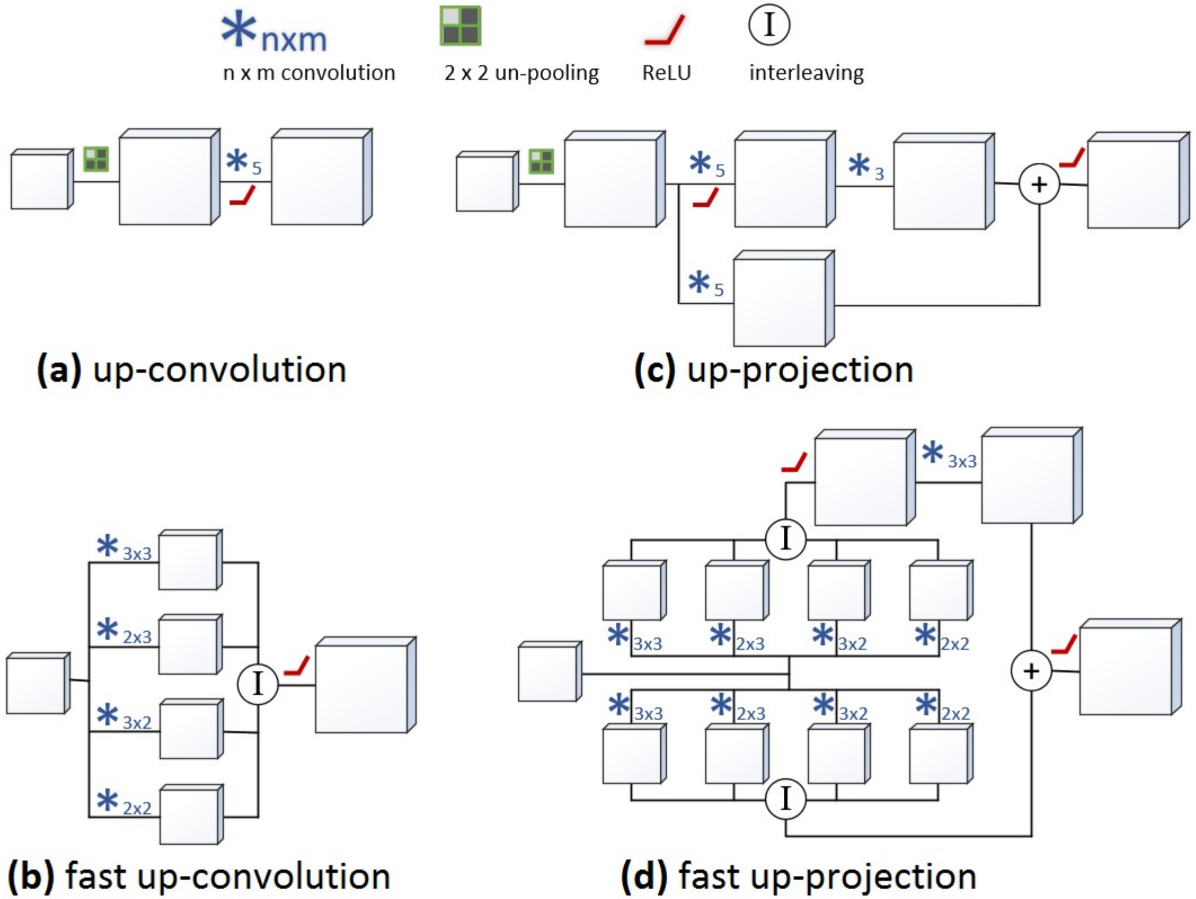


3. **FCRN**-SSD

Deeper Depth Prediction with Fully Convolutional Residual Networks, Laina et al.
\+ SSD: Single Shot MultiBox Detector, Liu et al.

Up-Projection Module:





3. FCRN-**SSD**

Evaluation:

+

=

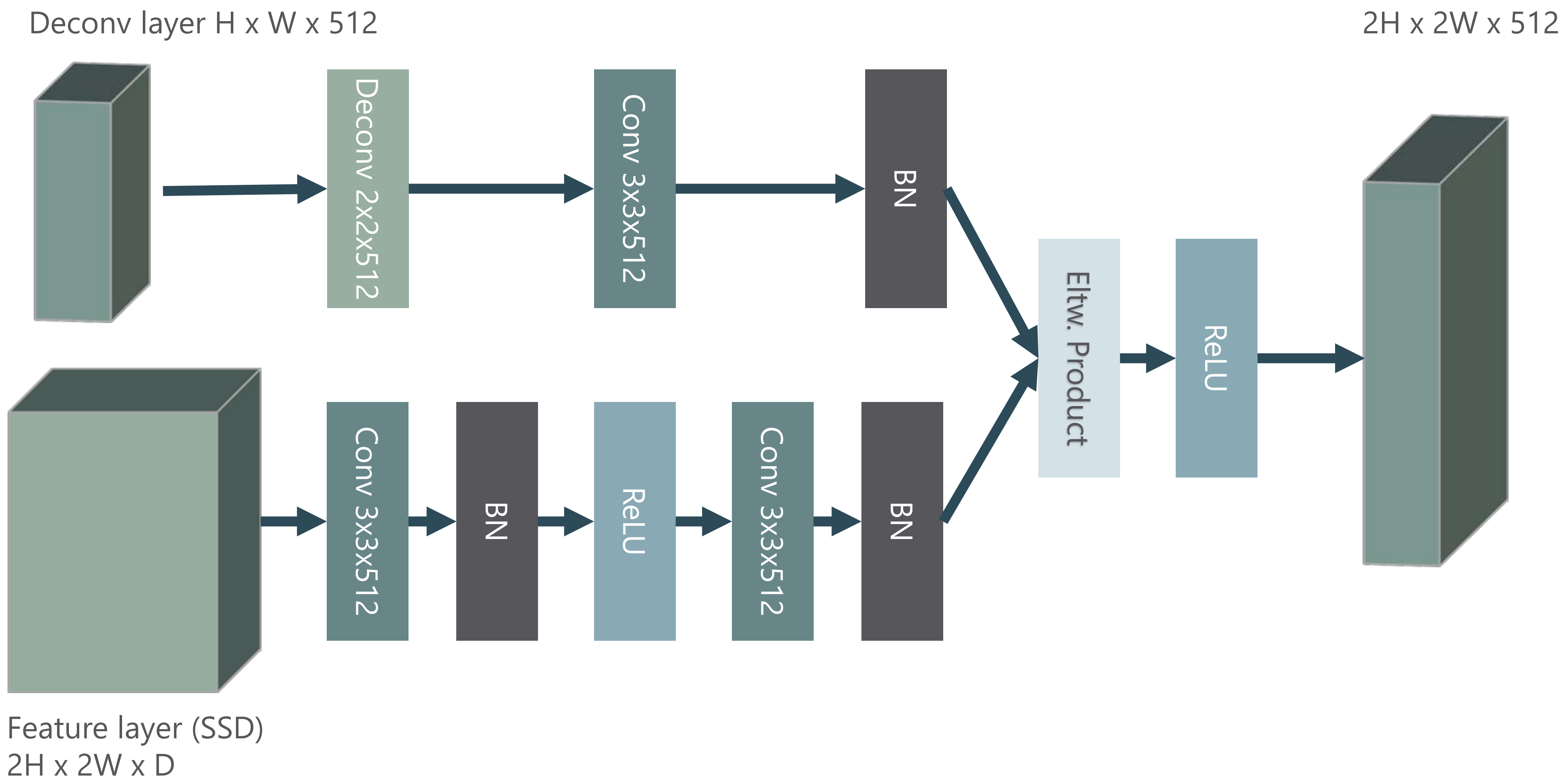
- Depth Images are generated by FCRN (epoch no. 67 at time of writing)
- Trained and Tested together with KITTI Object Challenge training dataset (60/40 split)
- Optimizer: RMSProp
- 8 images / batch, lr: 0.001, weight_decay: 0.0005
- Loss Function: $$L(x,c,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g))$$

*Training cancelled due to deadline of students in pool room

4. FCRN-DSSD

Deeper Depth Prediction with Fully Convolutional Residual Networks, Laina et al.
+ SSD: Single Shot MultiBox Detector, Liu et al.
+ DSSD : Deconvolutional Single Shot Detector Fu et al.

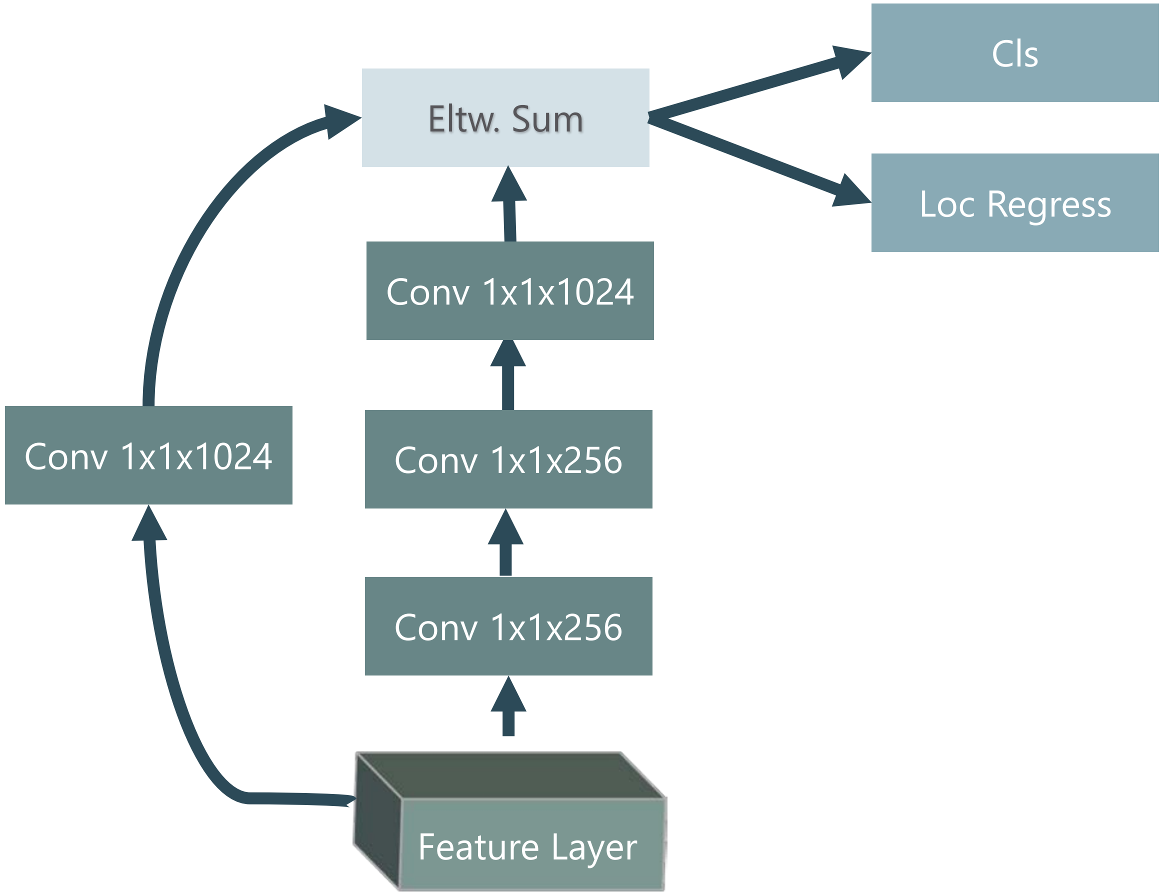
SSD Deconvolution Module:



4. FCRN-DSSD

Deeper Depth Prediction with Fully Convolutional Residual Networks, Laina et al.
+ SSD: Single Shot MultiBox Detector, Liu et al.
+ DSSD : Deconvolutional Single Shot Detector Fu et al.

SSD Prediction Module:



4. FCRN-DSSD

Deeper Depth Prediction with Fully Convolutional Residual Networks, Laina et al.
+ SSD: Single Shot MultiBox Detector, Liu et al.
+ DSSD : Deconvolutional Single Shot Detector Fu et al.

Loss functions:

- Depth: **berHu**

$$c = \frac{1}{5}\max_i(|\tilde{y}_i - y_i|), \qquad \mathrm{B}(x) = \begin{cases} |x|, & |x| \le c \\ \frac{x^2 + c^2}{2c}, & |x| > c \end{cases}$$

- SSD:

$$L(x,c,p,g) = \frac{1}{N}\left(L_{conf}(x,c) + \alpha L_{loc}(x,p,g)\right), \alpha = 1, N = \#\, matched\ default\ bb$$

$$L_{loc}(x,p,g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx,cy,w,h\}} x_{ij}^{k} smooth_{L1}(p_i^m - \widehat{g_j}^m)$$

$$L_{conf}(x,c) = -\sum_{i \in Pos}^{N} x_{ij}^{p} \log(\widehat{c_i}^p) - \sum_{i \in Neg} \log(\widehat{c_i}^0)$$