Automotive 3D Object Detection Without Target Domain Annotations

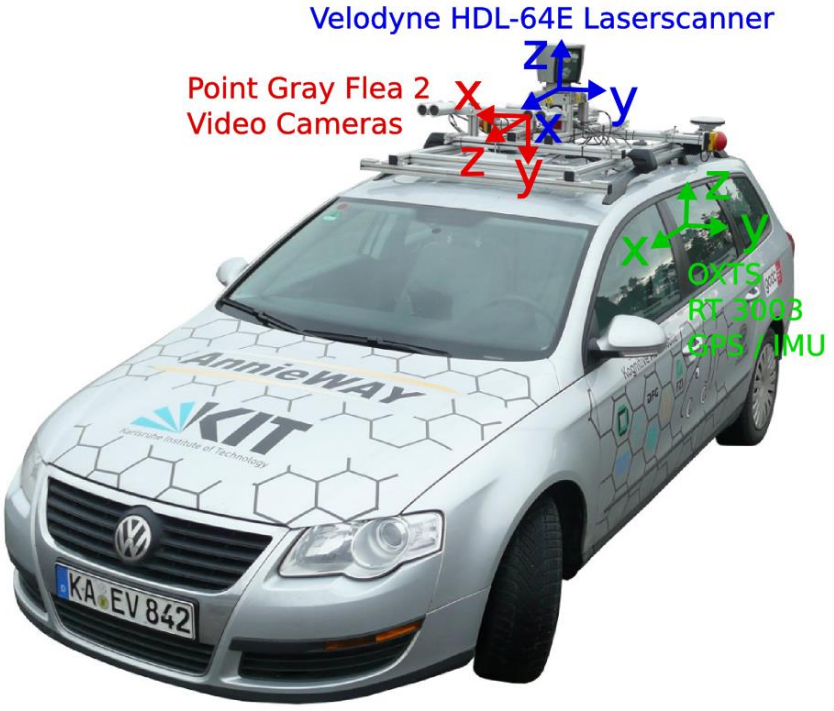
Erik Linder-Norén & Fredrik Gustafsson

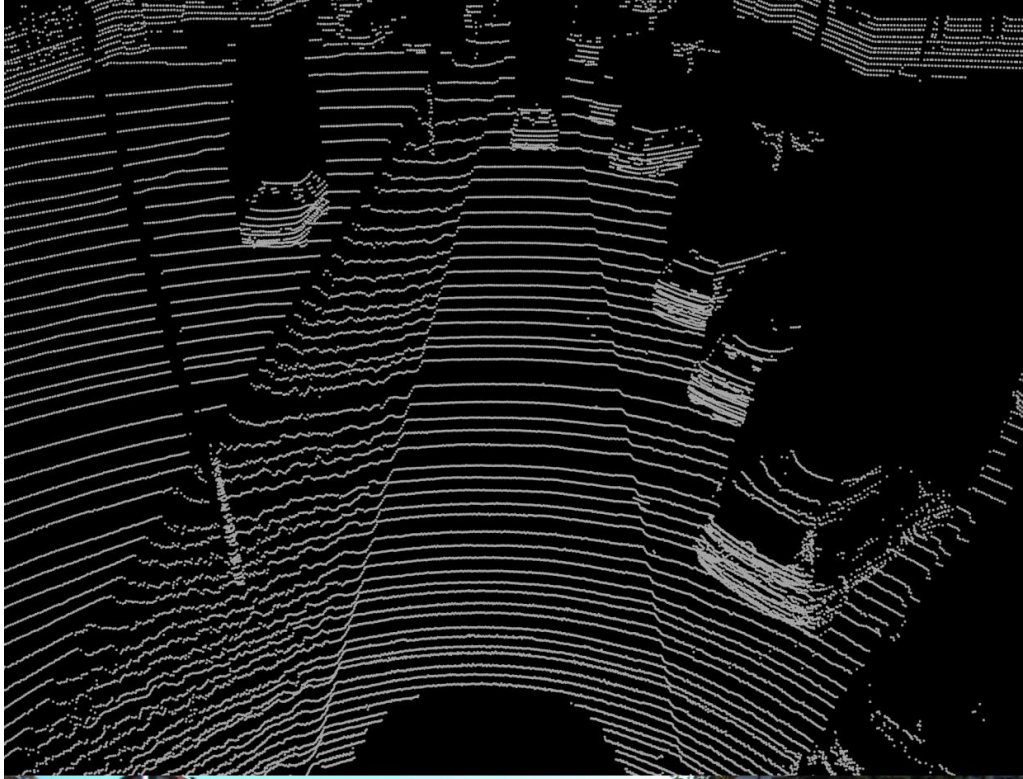
Intro

- Fredrik: EE, Erik: CS.
- Have been working remotely from Linköping.
- 3D detection (3DOD) of vehicles from LiDAR and image data, using deep learning (Fredrik).
- Domain adaptation (DA) via image translations using GANs (Erik).
- Supervisors: Eskil Jörgensen, Amrit Krishnan & Gustav Häger (LiU).
- Examiner: Michael Felsberg (LiU).

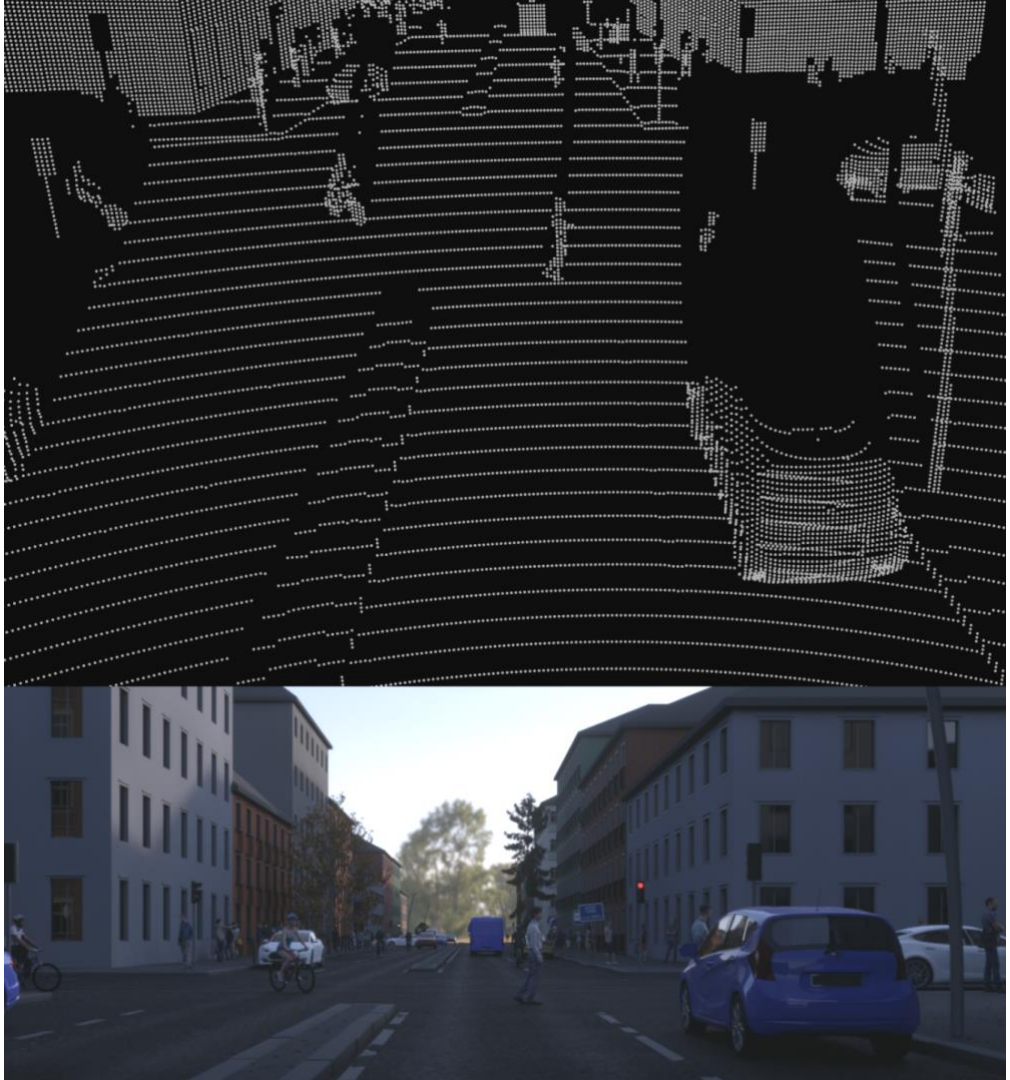
Problem Description – 3D Detection

- Input: Image from forward-facing camera + LiDAR point cloud.
- Output: Estimated 3D position, size and heading of all visible vehicles.
- Two used datasets: KITTI and SYN (7dLabs).





Vision meets Robotics: The KITTI Dataset, Andreas Geiger *et al.*



Video:
https://photos.app.goo.gl/XixvzQuoLFxyLBdH9

Problem Description – Domain Adaptation

- Given: Annotated dataset (source) and a dataset that is **not** annotated (target).
- Goal: Train a model on source images that performs well on target images.
- Method: Narrow the domain gap between source and target by translating source images to look more like target images, using GANs.

Problem Formulation

- Given:
  - SYN (Images, LiDAR point clouds, annotated 2Dbboxes, annotated 3Dbboxes).
  - KITTI (Images, LiDAR point clouds, annotated 2Dbboxes).

- Goal:
  - Train 3DOD model with maximum performance on KITTI (use annotated 3Dbboxes for evaluation).

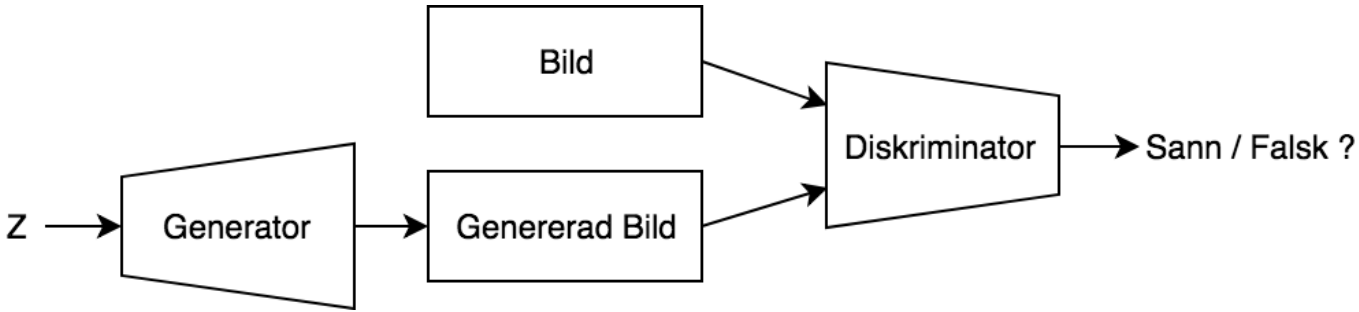
- Motivation:
  - Could be used to automatically annotate 3Dbboxes on Zenuity's internal datasets.
  - Could be used to automatically generate proposal 3Dbboxes on Zenuity's internal datasets, which then can be manually fine-tuned by a human annotator.

- Method:
  - Train a LiDAR model on SYN.
  - Train a LiDAR-and-image model on SYN.
  - Train a LiDAR-and-image model on SYN while applying domain adaptation on the images.

Generative Adversarial Networks

- Introduced by Ian Goodfellow *et al.* in 2014.
- Traditional model included two neural networks:
  - Generator - Generating data closely resembling the data distribution.
  - Discriminator - Discriminates between generated samples and samples from the data distribution.

Generative Adversarial Networks



Translation: GTA 5 and Cityscapes

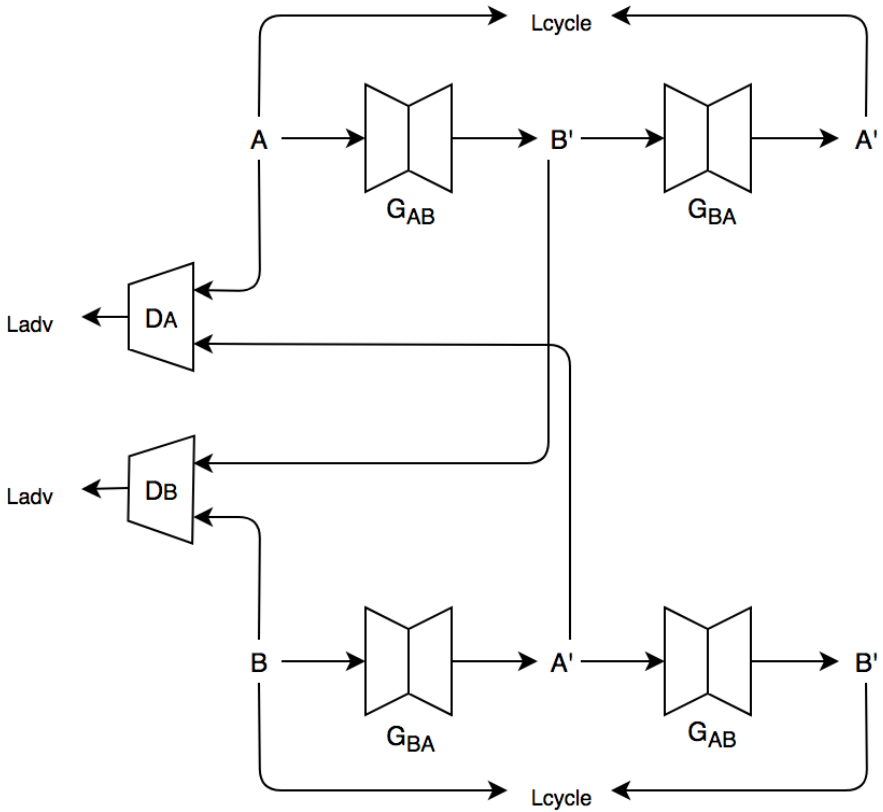
- Translate between a synthetic dataset and a dataset collected in the real world:
  - GTA 5 - Extracted images and annotations from the video game GTA 5.
  - Cityscapes - One of the larger datasets with semantic segmentation annotations.
- A common image-to-image translation problem.

CycleGAN

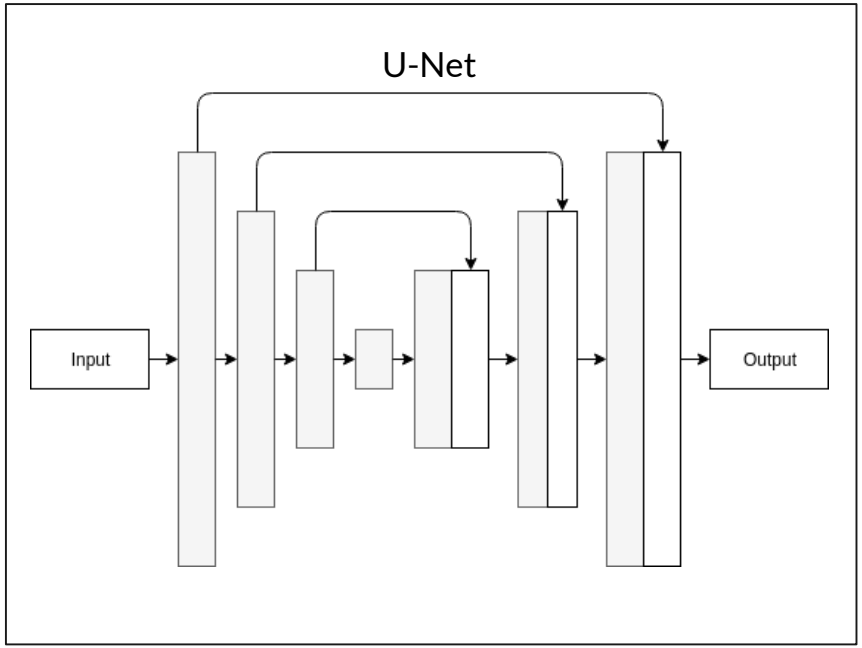
Unpaired Image-to-Image Translation using Generative Adversarial Networks, Zhu *et al*.

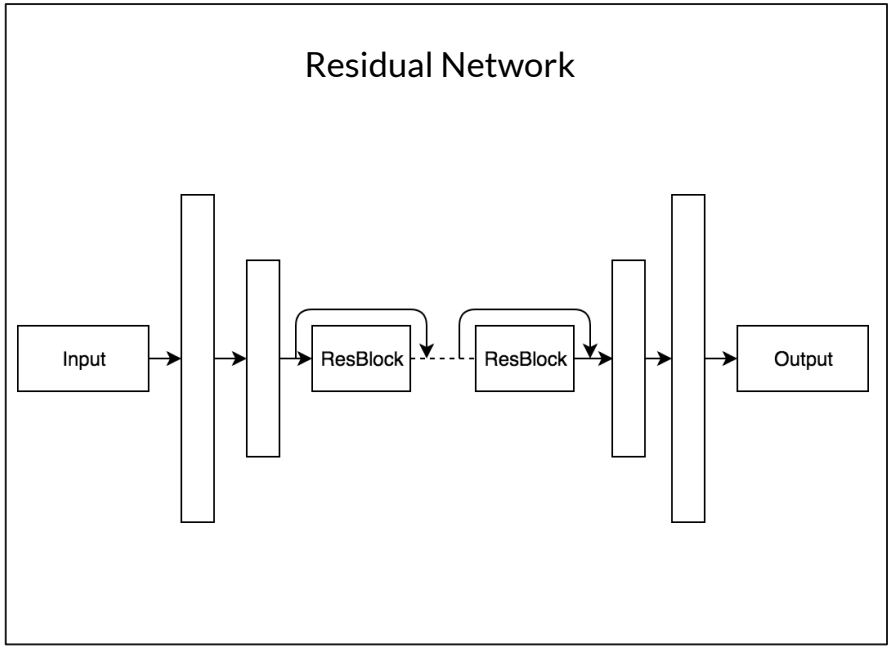
- Has been shown to produce image-to-image translations of high quality.
- Does not require that the domains are paired.
- Two generators and two discriminators.
- Cycle-consistency.

CycleGAN



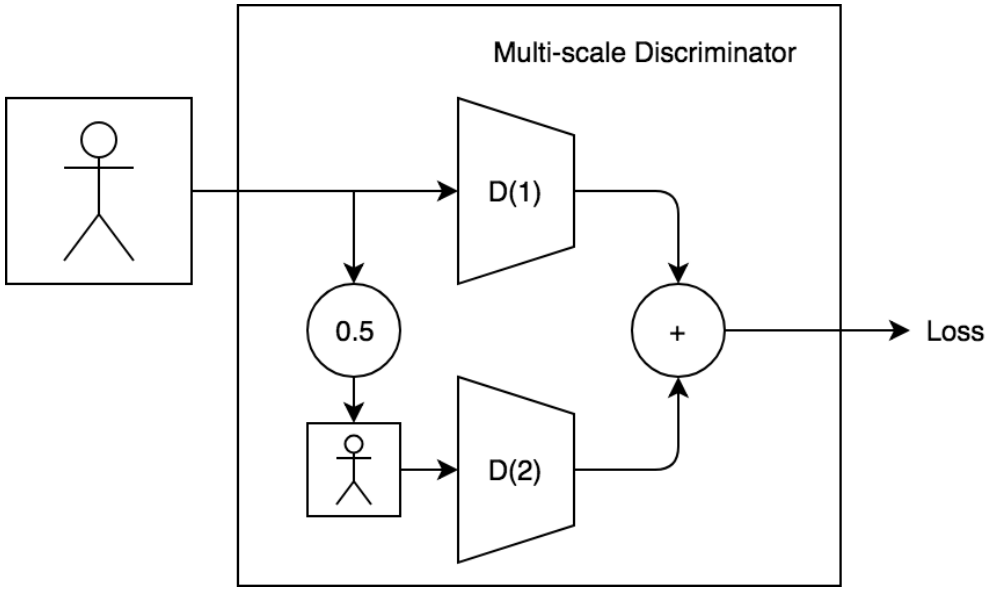
Representations - Generator

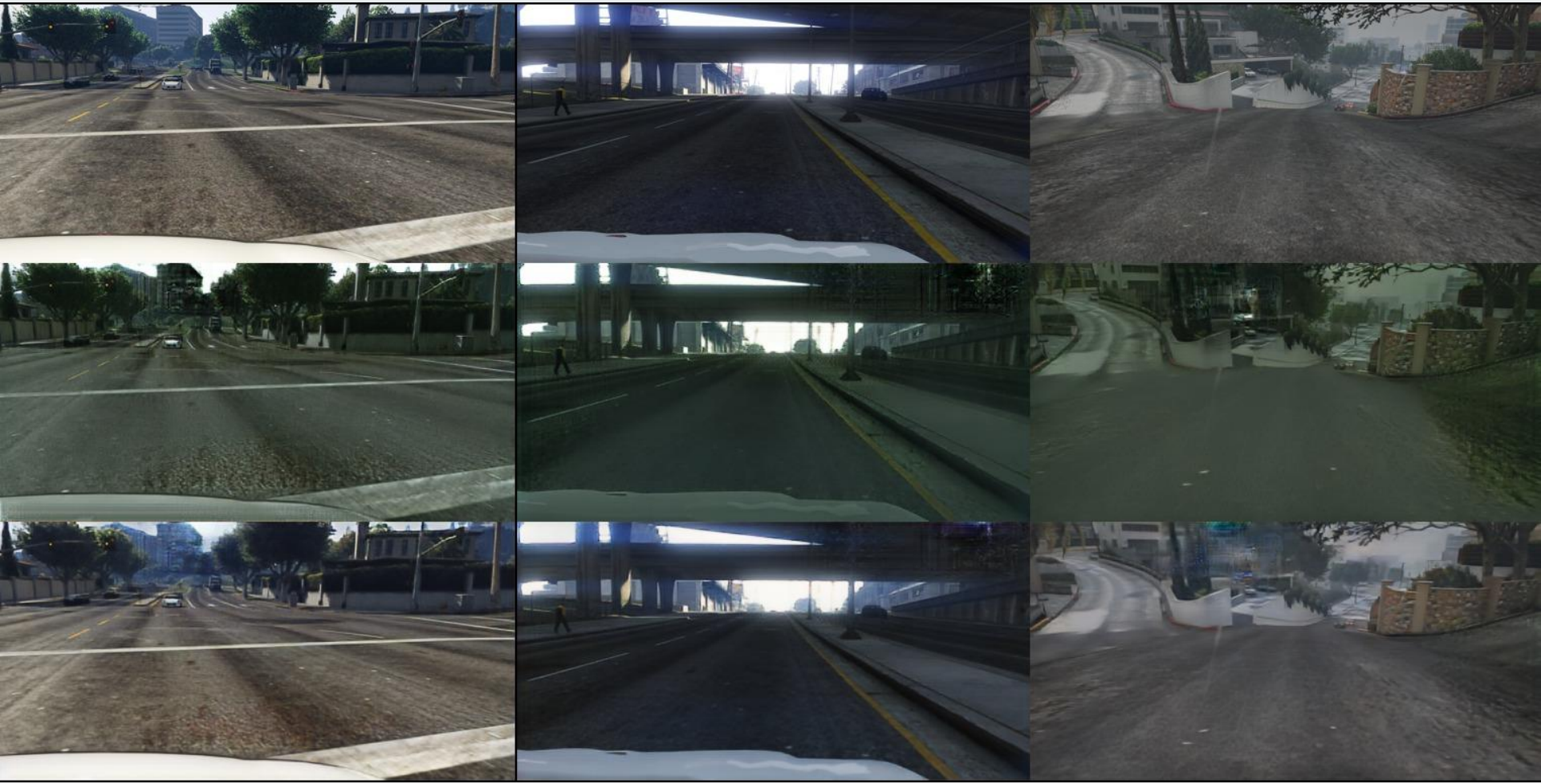




Representation - Discriminator

- Multi-scale discriminator.

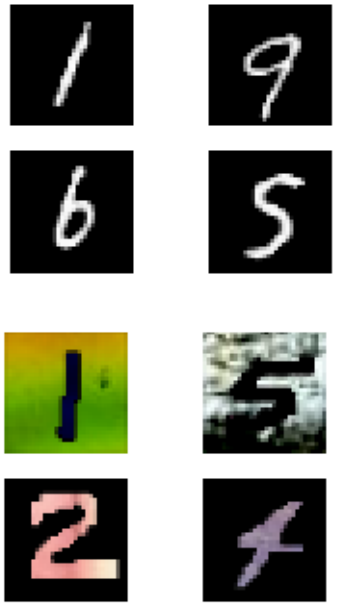




Translation: GTA 5 and Cityscapes

Domain Adaption: MNIST to MNIST-M

- Verify idea.
- Classification task.
- Datasets:
  - MNIST - Images of handwritten digits.
  - MNIST-M - Randomly modified images from MNIST.

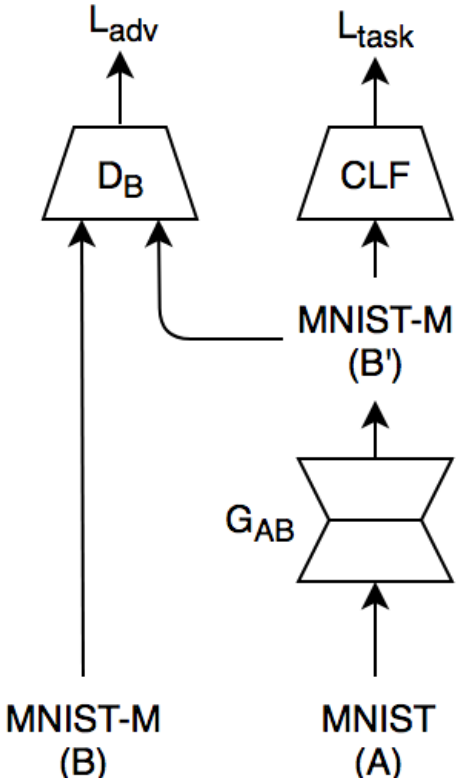


PixelDA

Unsupervised Pixel-level Domain Adaptation With Generative Adversarial Networks, Bousmalis *et al.*

- Classification network trains both on images sampled from source and on translated images.
- Generator is optimized for a correct domain translation and for preserving semantics of input images.
- Qualitative and quantitative evaluation.

PixelDA



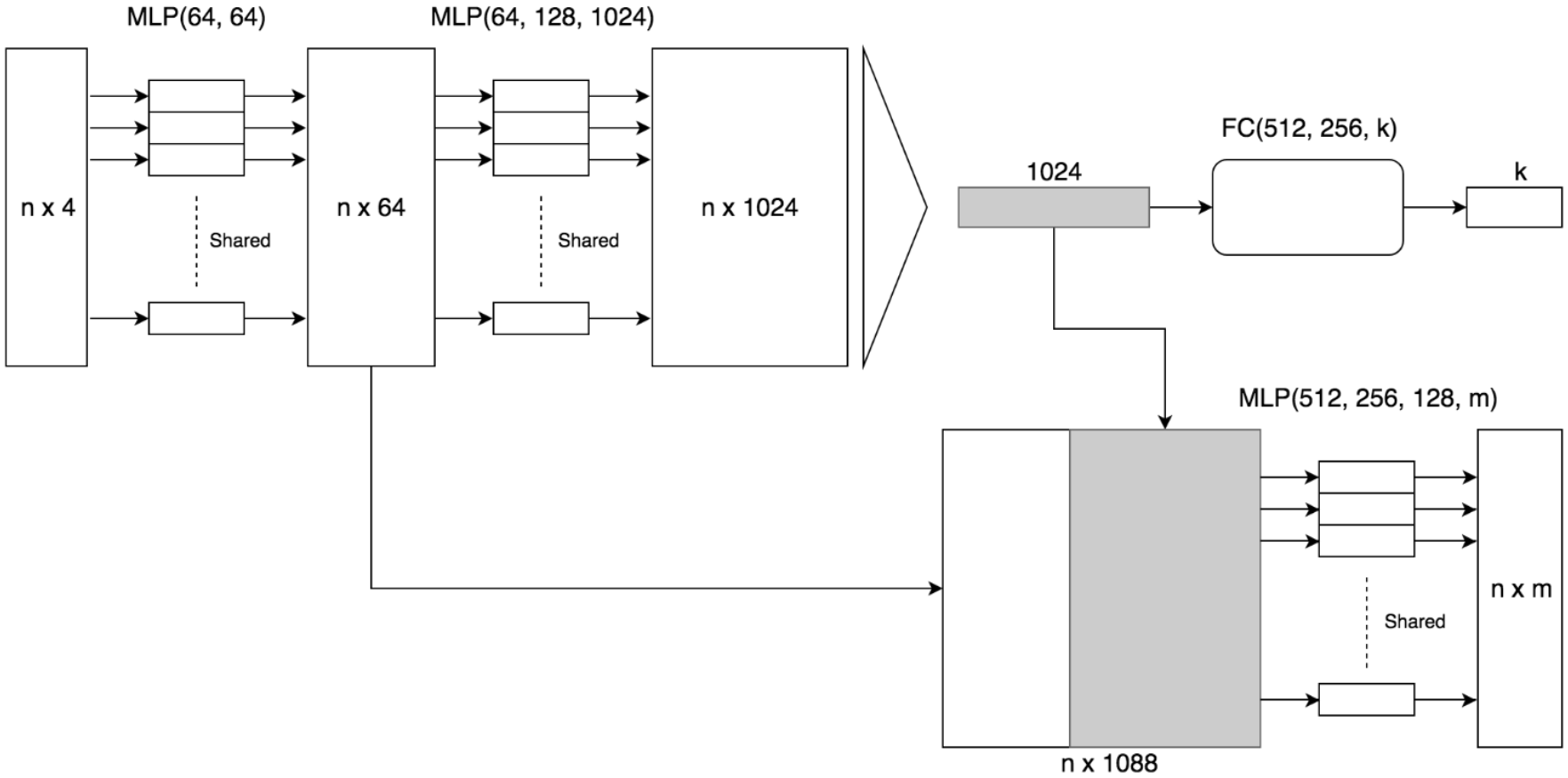
Qualitative Results

Quantitative Results

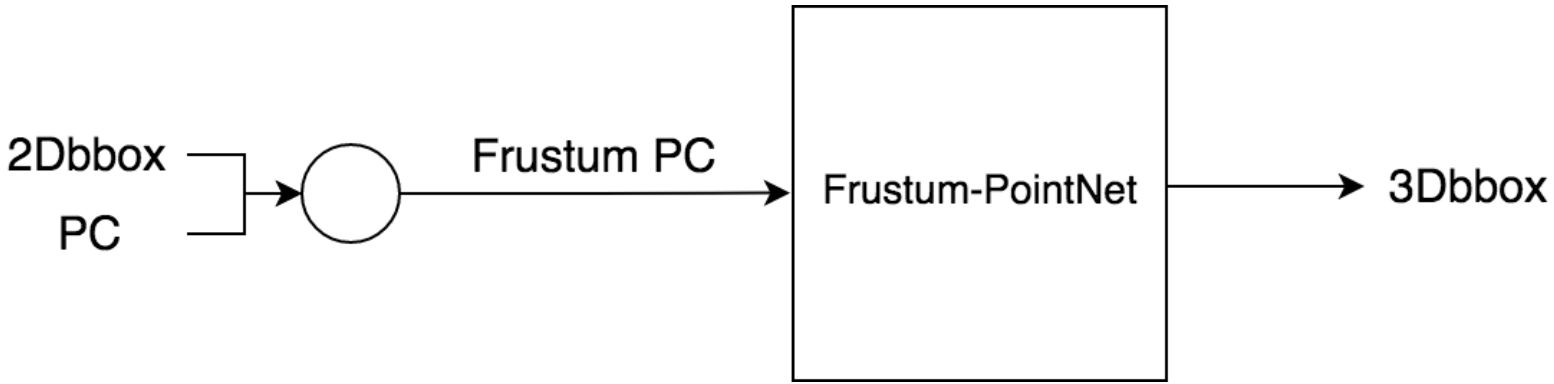
MNIST-M Classification	
Model	Accuracy
Reference	55 %
PixelDA (U-Net)	91 %
PixelDA (ResNet)	95 %

3D Detection - Frustum-PointNet (LiDAR model)

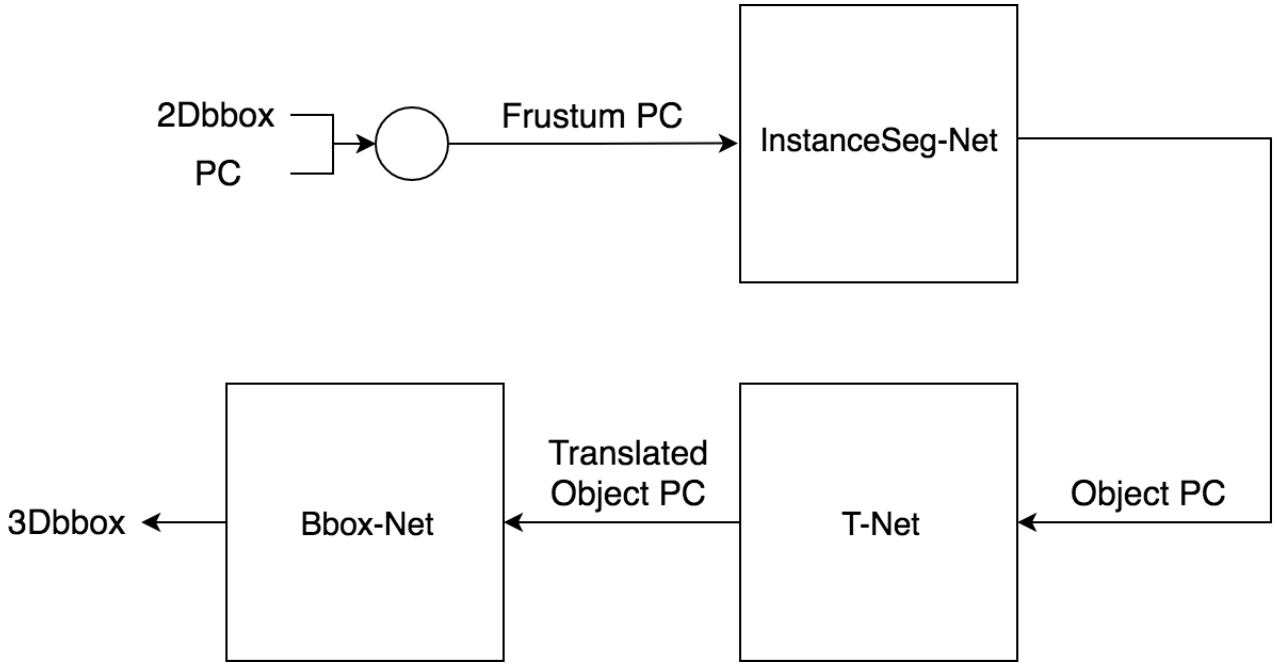
- ***Frustum PointNets for 3D Object Detection from RGB-D Data***, Charles R. Qi *et al.* (CVPR 2018).
- Uses ***PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation***, Charles R. Qi *et al.* (CVPR 2017).


MLP(64, 64)
MLP(64, 128, 1024)
n x 4
Shared
n x 64
Shared
n x 1024
1024
FC(512, 256, k)
k
MLP(512, 256, 128, m)
Shared
n x m
n x 1088

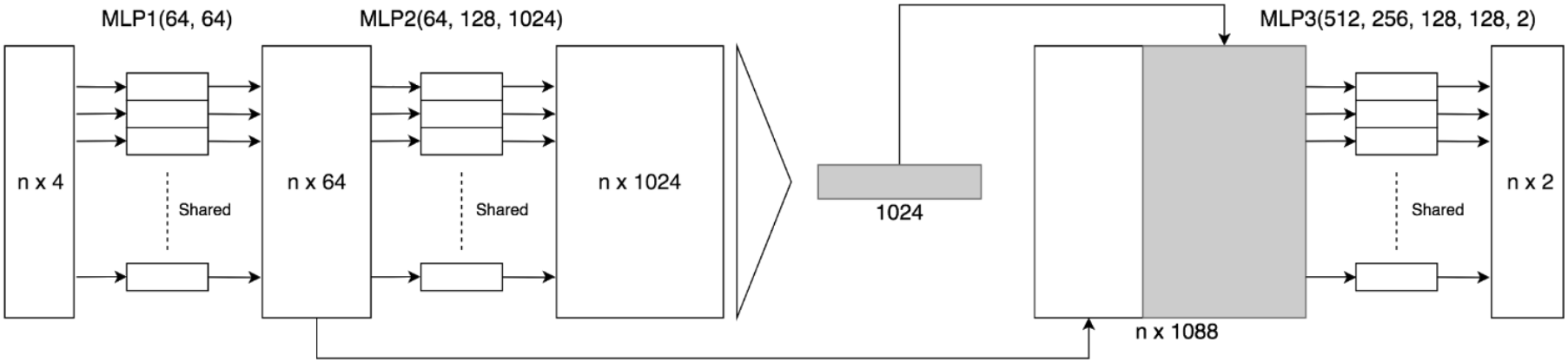
Frustum-PointNet - Overview



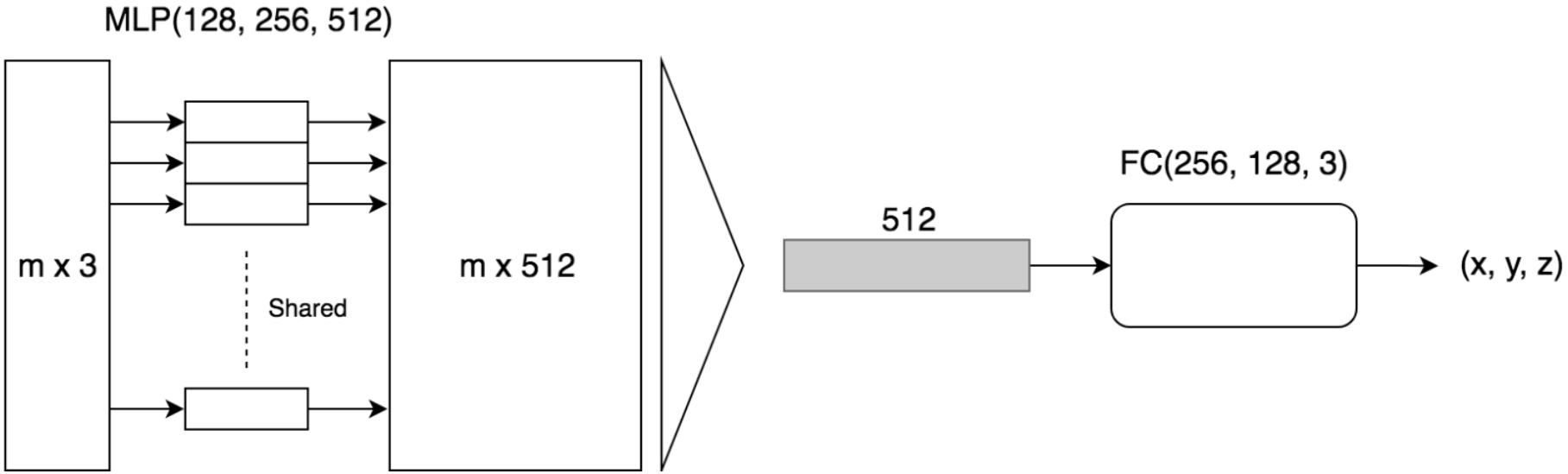
Frustum-PointNet - Detailed Overview



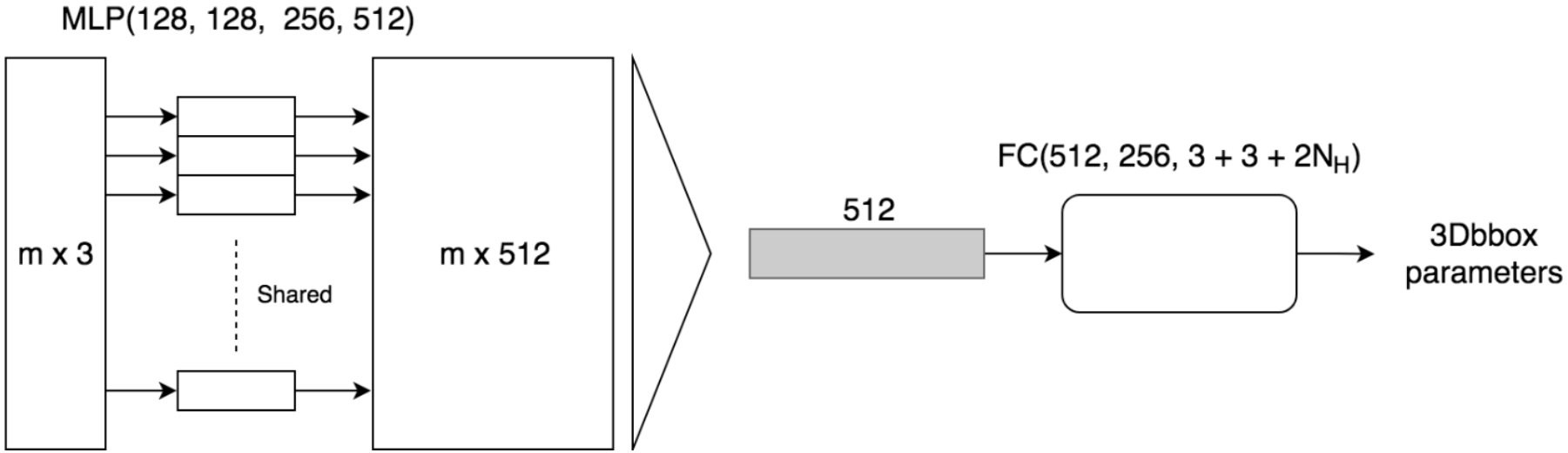
Frustum-PointNet - InstanceSeg-Net



Frustum-PointNet - T-Net



Frustum-PointNet - Bbox-Net



Frustum-PointNet - Quantitative Results

- Results on ***KITTI val*** (3769 examples), trained on ***KITTI train*** (3712 examples).
- 2D detections from the Frustum-PointNet authors are used as input, the corresponding confidence score is used also as 3D confidence score.
- ***3D-AP*** ("Average Precision" based on 3D IoU):

Method	Easy	Moderate	Hard
Frustum-PointNet (version 1) [47]	83.26 %	69.28 %	62.56 %
Frustum-PointNet (version 2) [47]	83.76 %	70.92 %	63.65 %
Our Frustum-PointNet - KITTI	78.01 %	65.22 %	59.06 %
Our Frustum-PointNet - KITTI (50 %)	93.73 %	87.96 %	79.04 %

Frustum-PointNet - Quantitative Results

- Results on ***KITTI val*** (3769 examples), trained on ***KITTI train*** (3712 examples).
- 2D detections from the Frustum-PointNet authors are used as input, the corresponding confidence score is used also as 3D confidence score.
- ***Top-view-AP*** ("Average Precision" based on top-view IoU):

Method	Easy	Moderate	Hard
Frustum-PointNet (version 1) [47]	87.82 %	82.44 %	74.77 %
Frustum-PointNet (version 2) [47]	88.16 %	84.02 %	76.44 %
Our Frustum-PointNet - KITTI	85.30 %	79.89 %	72.38 %
Our Frustum-PointNet - KITTI (50 %)	94.13 %	88.50 %	85.65 %

Frustum-PointNet - Qualitative Results

- Model trained on ***KITTI train random*** (6733 examples), evaluated on sequences from ***KITTI test***, 2D detections from DLO are used as input:

Video:
https://photos.app.goo.gl/VwNfqdhBAdPUTna36

Video:
https://photos.app.goo.gl/hdhA3tZcPekNBn118

Frustum-PointNet - Generalization SYN → KITTI

- Results on ***KITTI val*** (3769 examples), trained on ***SYN train*** (20 000 examples).
- ***3D-AP***:

Method	Easy	Moderate	Hard
Frustum-PointNet (version 1) [47]	83.26 %	69.28 %	62.56 %
Frustum-PointNet (version 2) [47]	83.76 %	70.92 %	63.65 %
Our Frustum-PointNet - KITTI	78.01 %	65.22 %	59.06 %
Our Frustum-PointNet - KITTI (50 %)	93.73 %	87.96 %	79.04 %
Our Frustum-PointNet - SYN	10.23 %	7.96 %	7.23 %
Our Frustum-PointNet - SYN (50 %)	69.13 %	63.46 %	56.41 %

Frustum-PointNet - Generalization SYN → KITTI

- Results on ***KITTI val*** (3769 examples), trained on ***SYN train*** (20 000 examples).
- ***Top-view-AP***:

Method	Easy	Moderate	Hard
Frustum-PointNet (version 1) [47]	87.82 %	82.44 %	74.77 %
Frustum-PointNet (version 2) [47]	88.16 %	84.02 %	76.44 %
Our Frustum-PointNet - KITTI	85.30 %	79.89 %	72.38 %
Our Frustum-PointNet - KITTI (50 %)	94.13 %	88.50 %	85.65 %
Our Frustum-PointNet - SYN	30.43 %	28.54 %	23.43 %
Our Frustum-PointNet - SYN (50 %)	76.63 %	70.68 %	61.88 %

Frustum-PointNet - Generalization SYN → KITTI

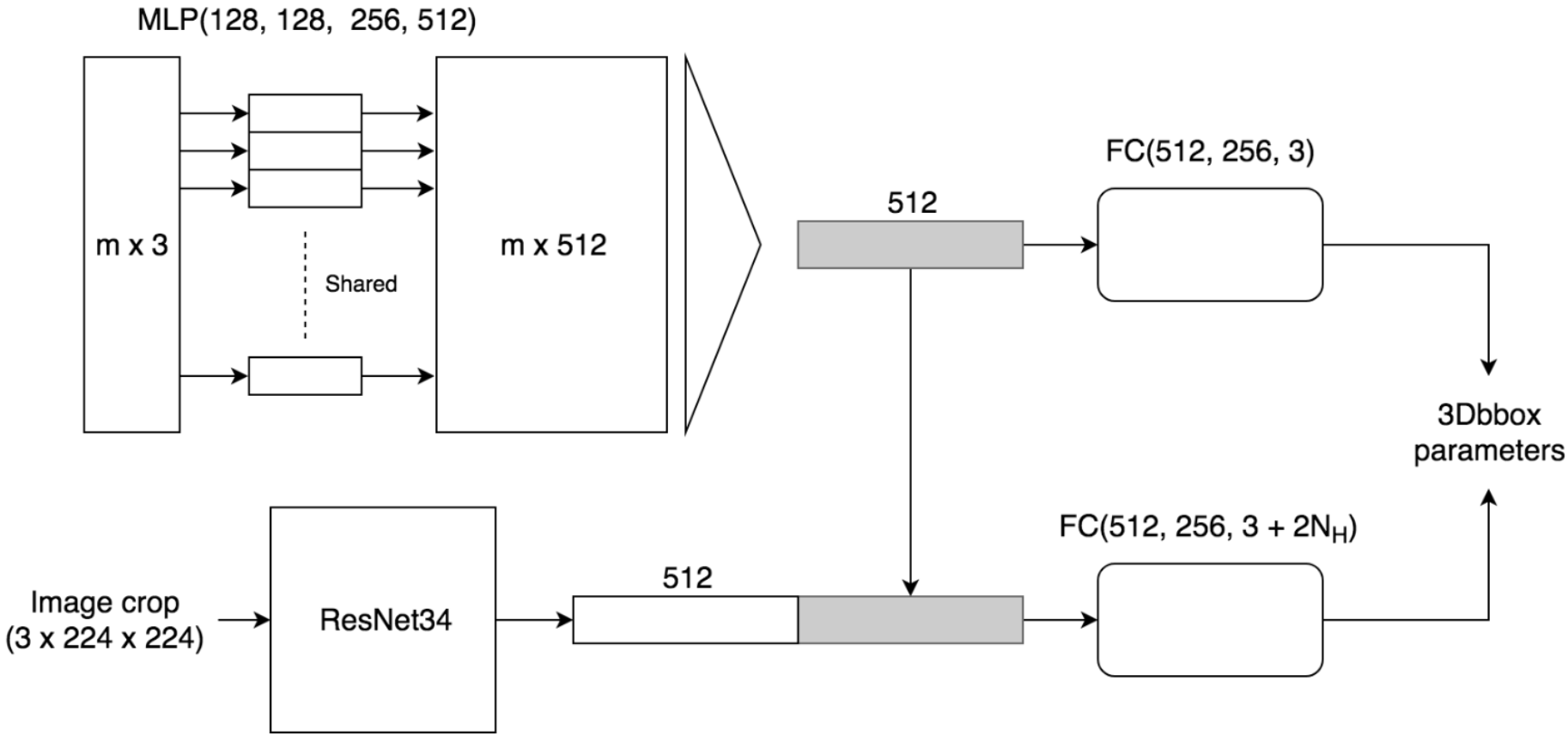
- Estimated 3Dbboxes are actually quite close to ground truth, despite the large difference in terms of performance with a 70 % threshold.
- Main problem: estimated 3Dbboxes are too large, mainly due to the mean car size in SYN being significantly larger than in KITTI.
- Model trained on ***SYN train*** (20 000 examples), evaluated on sequences from ***KITTI test***, 2D detections from DLO are used as input:

Video:
https://photos.app.goo.gl/19z6WhNRtFZvSpkh7

Video:
https://photos.app.goo.gl/MysB16vddebLkUCt8

Extended Frustum-PointNet (LiDAR-and-image)

- Extend Frustum-PointNet to also take image features as input.
- Extract a feature vector for each 2Dbbox using ResNet34.
- Fuse with the LiDAR feature vector and use to estimate **size and heading**.
- Only affects **Bbox-Net**:


MLP(128, 128, 256, 512)
m x 3
Shared
m x 512
512
FC(512, 256, 3)
3Dbbox
parameters
Image crop
(3 x 224 x 224)
ResNet34
512
FC(512, 256, 3 + 2N_H)

Extended Frustum-PointNet – Quantitative Results

- Results on ***KITTI val*** (3769 examples), ***3D-AP***:

Method	Easy	Moderate	Hard
Our Frustum-PointNet - KITTI	78.01 %	65.22 %	59.06 %
Our Frustum-PointNet - KITTI (50 %)	93.73 %	87.96 %	79.04 %
Our Frustum-PointNet - SYN	10.23 %	7.96 %	7.23 %
Our Frustum-PointNet - SYN (50 %)	69.13 %	63.46 %	56.41 %
Our Extended - KITTI	68.72 %	57.21 %	50.57 %
Our Extended - KITTI (50 %)	88.90 %	87.39 %	78.53 %
Our Extended - SYN	5.67 %	4.72 %	3.90 %
Our Extended - SYN (50 %)	61.07 %	56.95 %	49.93 %

Extended Frustum-PointNet – Quantitative Results

- Results on ***KITTI val*** (3769 examples), ***Top-view-AP***:

Method	Easy	Moderate	Hard
Our Frustum-PointNet - KITTI	85.30 %	79.89 %	72.38 %
Our Frustum-PointNet - KITTI (50 %)	94.13 %	88.50 %	85.65 %
Our Frustum-PointNet - SYN	30.43 %	28.54 %	23.43 %
Our Frustum-PointNet - SYN (50 %)	76.63 %	70.68 %	61.88 %
Our Extended - KITTI	81.18 %	72.78 %	64.47 %
Our Extended - KITTI (50 %)	89.27 %	88.25 %	85.36 %
Our Extended - SYN	17.27 %	17.19 %	14.62 %
Our Extended - SYN (50 %)	72.65 %	69.16 %	60.63 %

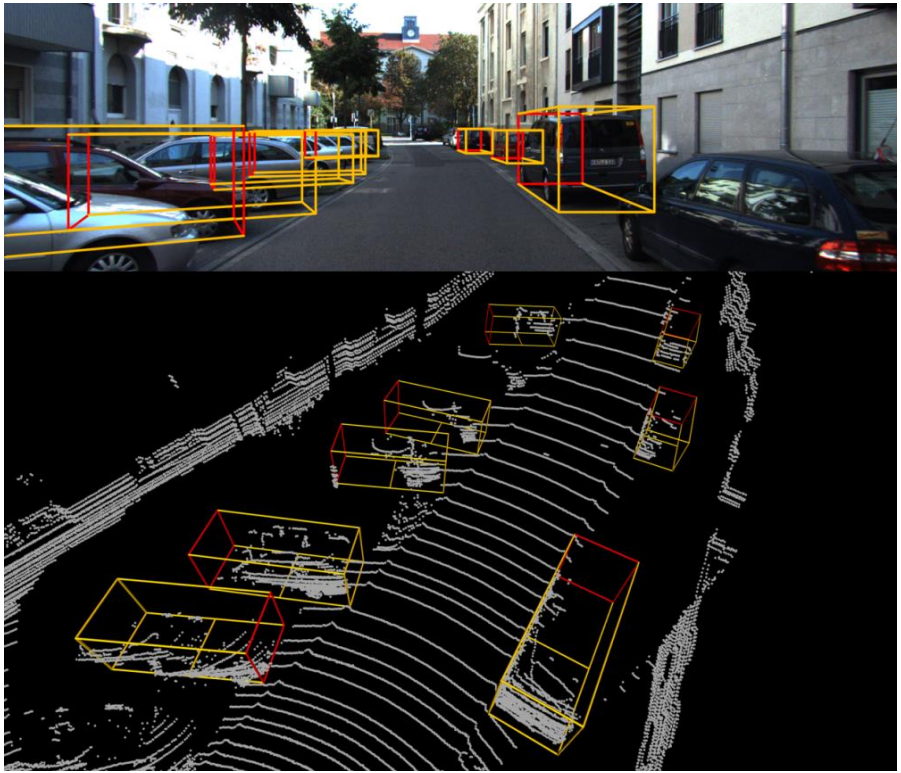
Extended Frustum-PointNet – Quantitative Results

- Explicitly using image features did thus **not** improve the performance, even when trained on KITTI.
- For models trained on SYN, using image features clearly degrades the performance (worse generalization SYN → KITTI).

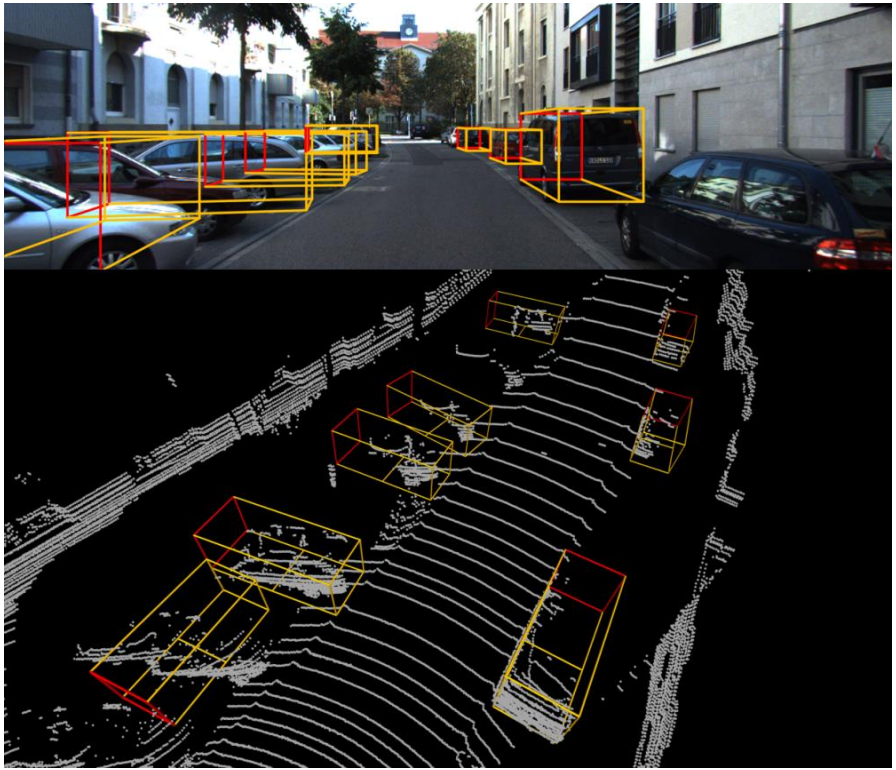
Extended Frustum-PointNet – Qualitative Results

- Seems to improve the heading estimate (less volatile), at least for vehicles far in-front of the car.
- May however have overfitted the model to the most common heading angles in the dataset:

Without image features:



With image features:



Extended Frustum-PointNet – Qualitative Results

- The most important image information might already be utilized in Frustum-PointNet since it takes 2D detections as input.
- The model might has become more susceptible to truncated and/or occluded vehicles.
- Still definitely possible that adding image features could improve the performance, **a more rigorous analysis is needed.**

Image-Only Model

- Assumes to be given a 2Dbbox as input.
- Apply ResNet34 + fully-connected network on each image patch.
- Estimate the 3Dbbox size (h, w, l), distance and image pixel coordinates for the eight 3Dbbox corners.
- Given this, the 3Dbbox parameters are estimated by minimizing the difference between the estimated pixel coordinates and the ones obtained by projecting the 3Dbbox onto the image.
- The estimated size and distance are used both for an initial guess and regularization in the obtained minimization problem (nonlinear least squares).

Image-Only Model - Overview

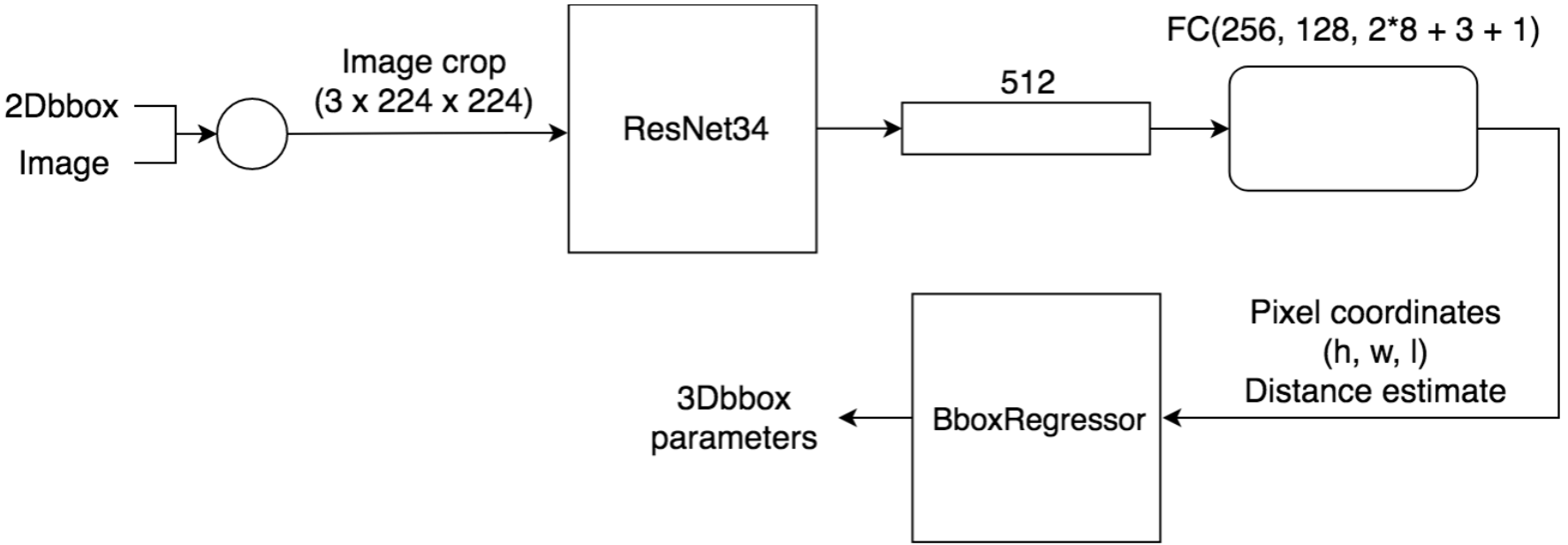


Image-Only Model - Quantitative Results

- Results on ***KITTI val*** (3769 examples), ***3D-AP***:

Method	Easy	Moderate	Hard
Mono3D [10]	2.53 %	2.31 %	2.31 %
3DOP [9]	6.55 %	5.07 %	4.10 %
Our Image-Only - KITTI	10.13 %	8.32 %	8.20 %
Our Image-Only - KITTI (50 %)	40.31 %	30.77 %	26.55 %
Our Frustum-PointNet - KITTI	78.01 %	65.22 %	59.06 %
Our Frustum-PointNet - KITTI (50 %)	93.73 %	87.96 %	79.04 %

Image-Only Model - Quantitative Results

- Results on ***KITTI val*** (3769 examples), ***Top-view-AP***:

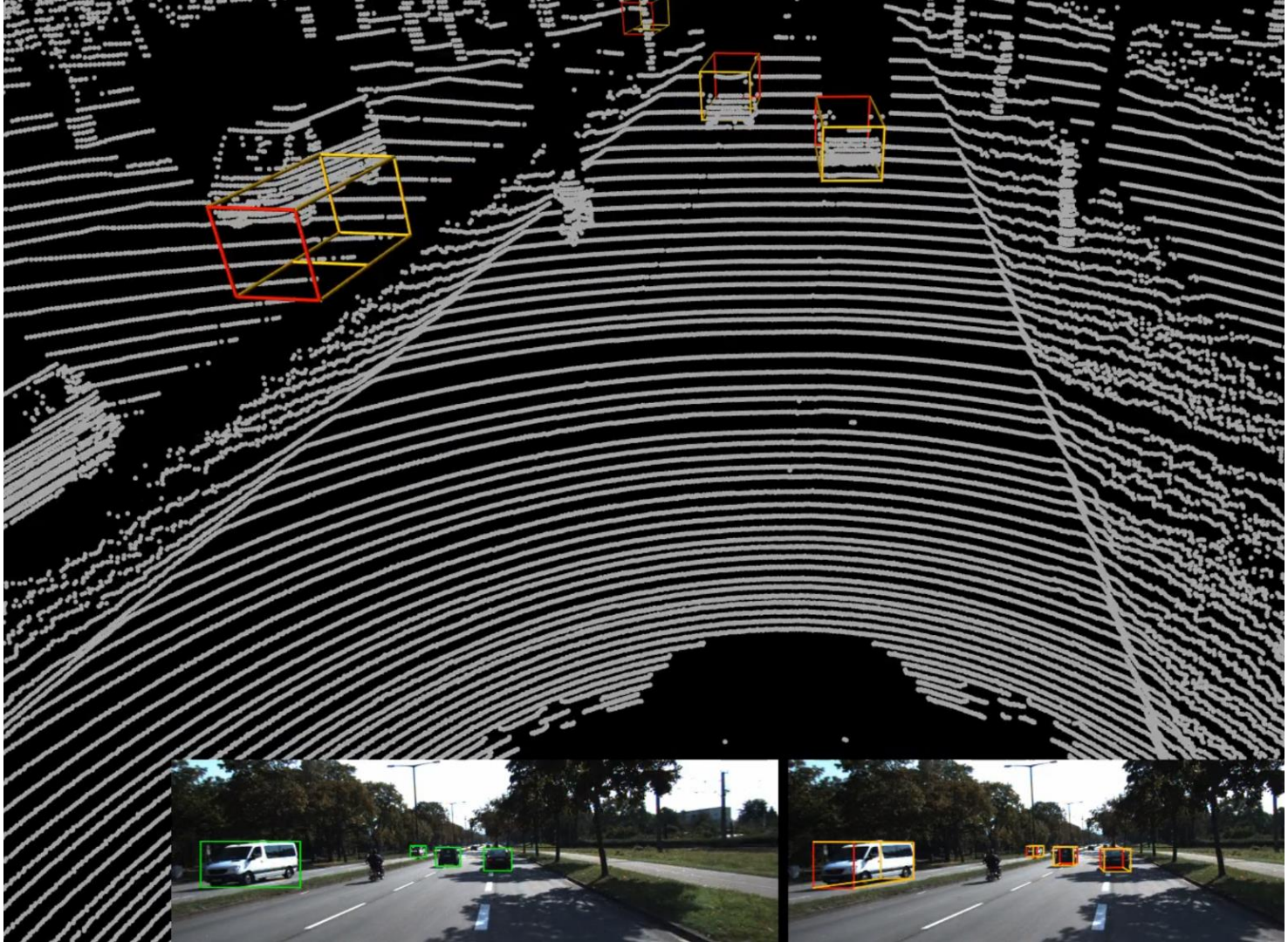
Method	Easy	Moderate	Hard
Mono3D [10]	5.22 %	5.19 %	4.13 %
3DOP [9]	12.63 %	9.49 %	7.59 %
Our Image-Only - KITTI	15.64 %	12.90 %	12.30 %
Our Image-Only - KITTI (50 %)	45.46 %	33.83 %	31.79 %
Our Frustum-PointNet - KITTI	85.30 %	79.89 %	72.38 %
Our Frustum-PointNet - KITTI (50 %)	94.13 %	88.50 %	85.65 %

Image-Only Model - Qualitative Results

- The pixel coordinates estimation works well (results almost always look good visualized in the image), but distance estimation is quite tricky.
- Overall though, the results seem quite promising.
- Model trained on ***KITTI train random*** (6733 examples), evaluated on sequences from ***KITTI test***, 2D detections from DLO are used as input:

Video:
https://photos.app.goo.gl/MLHfN6k98jJnNkiy6

Video:
https://photos.app.goo.gl/Yu2o9EX91bLJsf188



Domain Adaption: SYN to KITTI

- Implement 3DOD model in CycleGAN architecture.
- Train 3DOD model on translated images from SYN to KITTI.
- Evaluate performance of both the image-only model and Extended Frustum-PointNet by DA.
    - Qualitative and quantitative evaluations.

Architecture

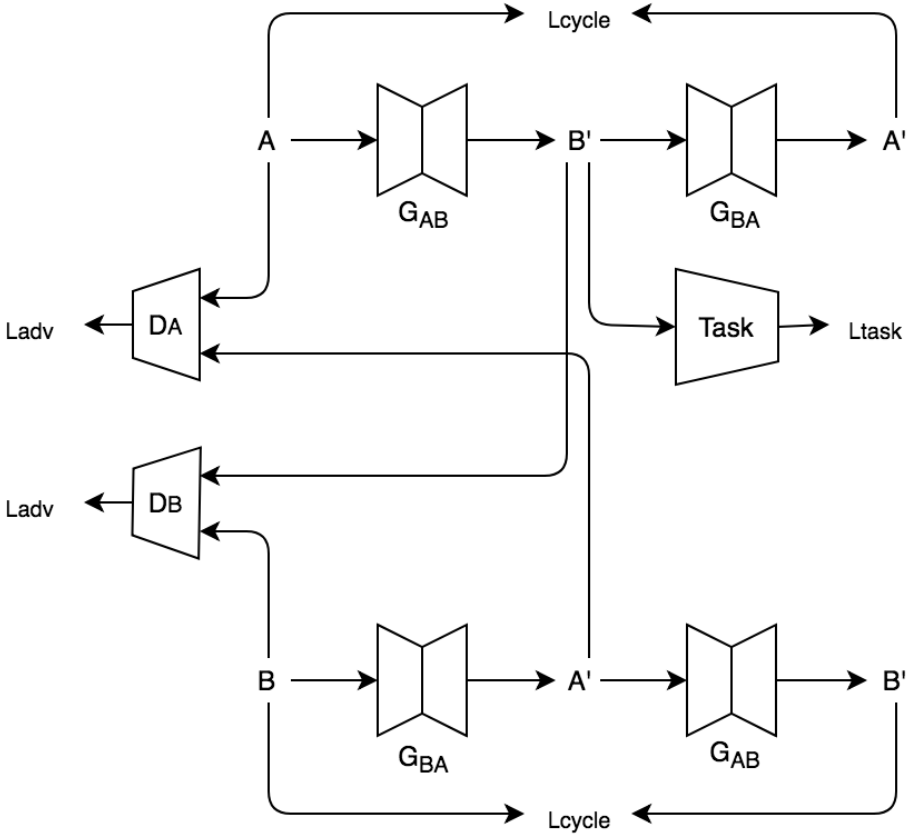


Image-Only Model - Quantitative results

- Results on ***KITTI val*** (3769 examples), ***3D-AP***:

Method	Easy	Moderate	Hard
Our Image-Only - SYN	0.06997 %	0.05798 %	0.05798 %
Our Image-Only - SYN (50 %)	0.3683 %	0.3422 %	0.3422 %
Our Image-Only - DA SYN	0.1007 %	0.1076 %	0.1076 %
Our Image-Only - DA SYN (50 %)	1.36 %	0.8004 %	0.8082 %

- Results on ***KITTI val*** (3769 examples), ***Top-view-AP***:

Method	Easy	Moderate	Hard
Our Image-Only - SYN	0.1094 %	0.09735 %	0.09735 %
Our Image-Only - SYN (50 %)	0.4952 %	0.4901 %	0.4999 %
Our Image-Only - DA SYN	0.2690 %	0.2671 %	0.2692 %
Our Image-Only - DA SYN (50 %)	2.23 %	1.02 %	1.03 %

Extended Frustum-PointNet - Quantitative results

- Results on ***KITTI val*** (3769 examples), ***3D-AP***:

Method	Easy	Moderate	Hard
Our Extended - SYN	5.67 %	4.72 %	3.90 %
Our Extended - SYN (50 %)	61.07 %	56.95 %	49.93 %
Our Extended - DA SYN	6.28 %	4.87 %	3.89 %
Our Extended - DA SYN (50 %)	55.26 %	53.06 %	44.82 %

- Results on ***KITTI val*** (3769 examples), ***Top-view-AP***:

Method	Easy	Moderate	Hard
Our Extended - SYN	17.27 %	17.19 %	14.62 %
Our Extended - SYN (50 %)	72.65 %	69.16 %	60.63 %
Our Extended - DA SYN	18.13 %	16.88 %	13.97 %
Our Extended - DA SYN (50 %)	65.95 %	63.67 %	56.08 %

Summary

- Implemented Frustum-PointNet for 3DOD and closely matched its reported performance on KITTI.
- Frustum-PointNet was found to transfer reasonably well from the synthetic SYN dataset to KITTI, is believed to definitely be usable in a semi-automatic annotation process of 3Dbboxes.
- Frustum-PointNet was extended to utilize image features, which surprisingly degraded its performance.
- Designed and implemented an image-only model with relatively good performance on KITTI.
- Implemented CycleGAN for translations between GTA 5 and Cityscapes.
- Successfully applied the PixelDA approach to the MNIST → MNIST-M domain adaptation problem.
- Applied on 3DOD (SYN → KITTI) the domain adaptation techniques did however not result in any significant improvement.

Questions?