Image Source: https://www.roboticsbusinessreview.com/unmanned/unmanned-ground/pbs-science-show-nova-shines-its-spotlight-on-self-driving-cars/

Computer Vision – WiSe 2021/2022
Prof'in. Dr. Gemma Roig

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Object Distance Estimation

Carla Frenzel (5602924)
Nils Möbus (6466745)
Tim Rosenkranz (6929884)
Benedikt Schröter (7487868)

10.02.2022

Overview

- Motivation
- Research
- Evaluating Other Approaches
- Model Modification
- Model Augmentation

Overview

- Motivation
- Research
- Evaluating Other Approaches
- Model Modification
- Model Augmentation

Area & Context

- Autonomous tasks
  → Extremely high relevance of computer vision

Image Source: https://www.computerbild.de/artikel/cb-News-Internet-Amazon-Zoox-Robotaxi-vorgestellt-29422449.html

Image Source: https://www.dji.com/de/mavic-3

Image Source: https://www.faz.net/aktuell/wirtschaft/amazon-praesentiert-haushaltsroboter-das-kann-astro-17560893.html

Requirement Analysis

LiDAR Sensor

- Expensive (500 - 8.000 $)

Image Source: https://www.sick.com/de/de/mess-und-detektionsloesungen/3d-lidar-sensoren/mrs1000p/c/g549379

Image Source: https://www.shunlongwei.com/de/more-efficient-lidar-sensing-for-self-driving-cars/

Requirement Analysis

Monocular RGB Camera

- Low-cost (≈ 125 $)

Image Source: https://de.aliexpress.com/item/4000997067112.html

Challenge

- Single RGB image

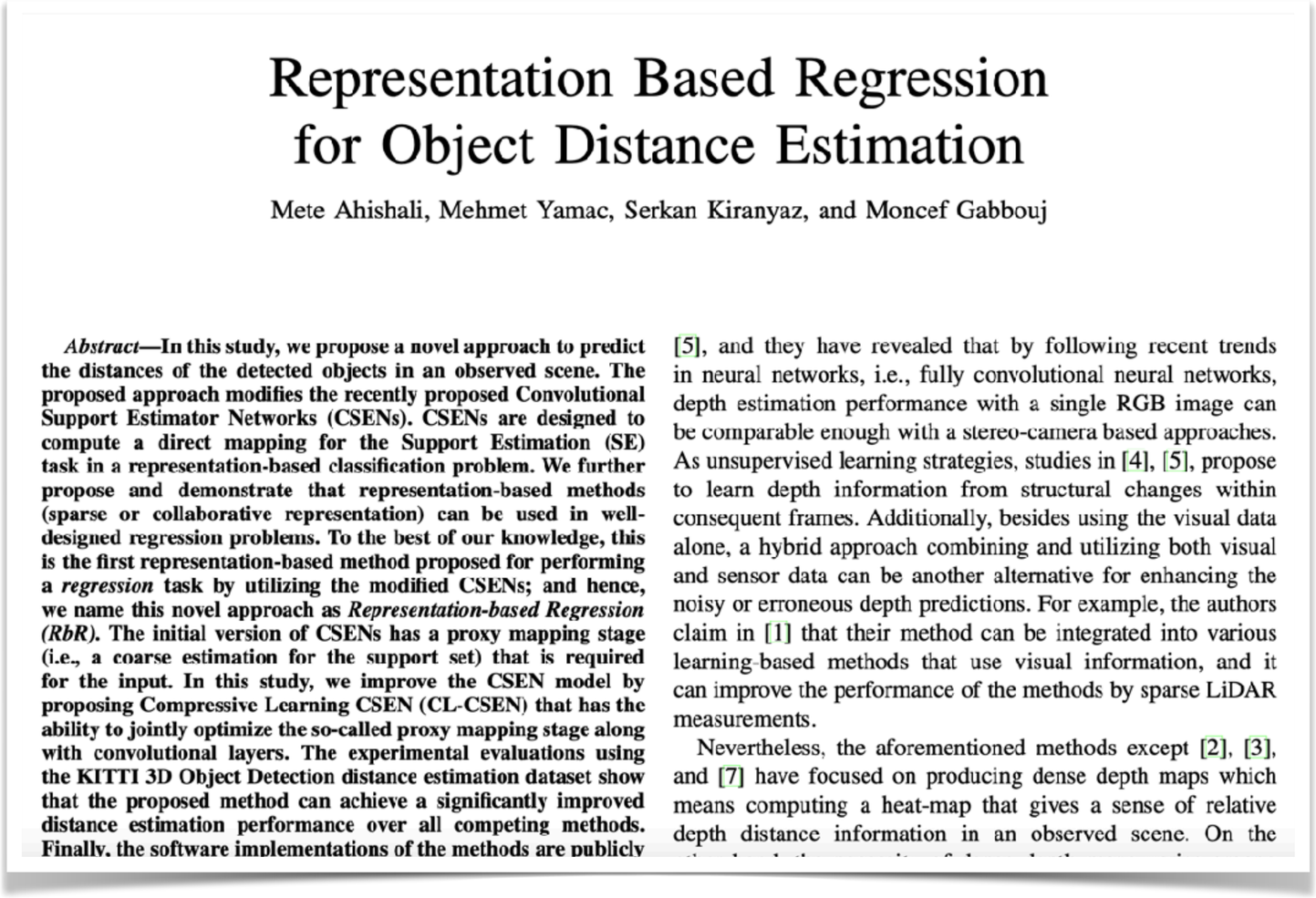
Benefits

- Can read colors
- More reliable in bad weather

Overview

- Motivation
- Research
- Evaluating Other Approaches
- Model Modification
- Model Augmentation

Paper



Representation Based Regression for Object Distance Estimation

Mete Ahishali, Mehmet Yamac, Serkan Kiranyaz, and Moncef Gabbouj

***Abstract*—In this study, we propose a novel approach to predict the distances of the detected objects in an observed scene. The proposed approach modifies the recently proposed Convolutional Support Estimator Networks (CSENs). CSENs are designed to compute a direct mapping for the Support Estimation (SE) task in a representation-based classification problem. We further propose and demonstrate that representation-based methods (sparse or collaborative representation) can be used in well-designed regression problems. To the best of our knowledge, this is the first representation-based method proposed for performing a *regression* task by utilizing the modified CSENs; and hence, we name this novel approach as *Representation-based Regression (RbR)*. The initial version of CSENs has a proxy mapping stage (i.e., a coarse estimation for the support set) that is required for the input. In this study, we improve the CSEN model by proposing Compressive Learning CSEN (CL-CSEN) that has the ability to jointly optimize the so-called proxy mapping stage along with convolutional layers. The experimental evaluations using the KITTI 3D Object Detection distance estimation dataset show that the proposed method can achieve a significantly improved distance estimation performance over all competing methods. Finally, the software implementations of the methods are publicly**

[5], and they have revealed that by following recent trends in neural networks, i.e., fully convolutional neural networks, depth estimation performance with a single RGB image can be comparable enough with a stereo-camera based approaches. As unsupervised learning strategies, studies in [4], [5], propose to learn depth information from structural changes within consequent frames. Additionally, besides using the visual data alone, a hybrid approach combining and utilizing both visual and sensor data can be another alternative for enhancing the noisy or erroneous depth predictions. For example, the authors claim in [1] that their method can be integrated into various learning-based methods that use visual information, and it can improve the performance of the methods by sparse LiDAR measurements.

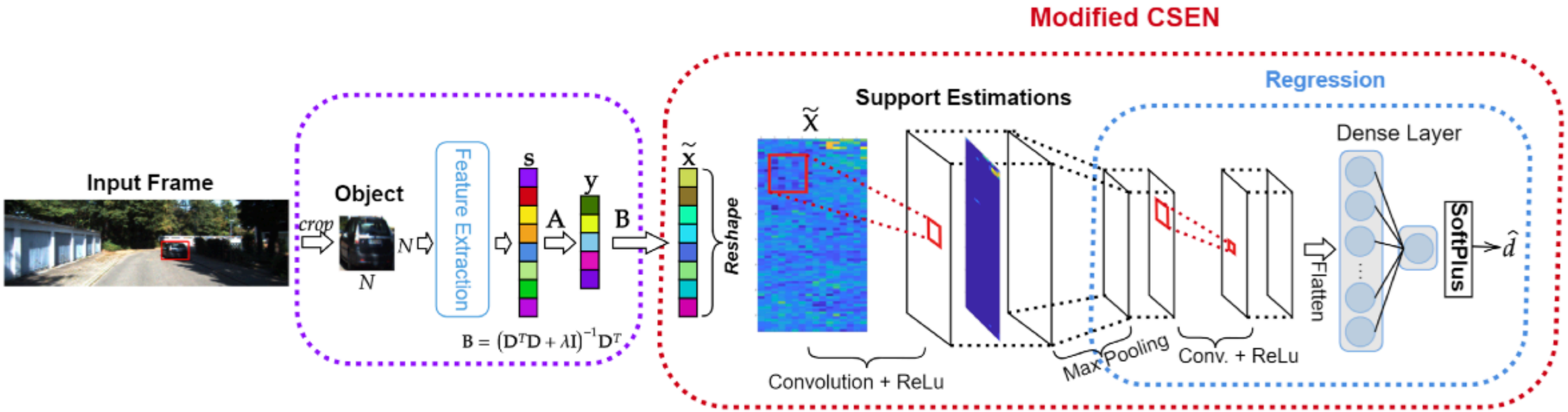
Nevertheless, the aforementioned methods except [2], [3], and [7] have focused on producing dense depth maps which means computing a heat-map that gives a sense of relative depth distance information in an observed scene. On the

- Published 2021
- Direct Link: https://arxiv.org/abs/2106.14208v1
- Public Implementation: https://github.com/meteahishali/CSENDistance

[Ahi21, Git21]

Architecture & Algorithm

- Model: **CSEN** (Convolutional Support Estimator Networks)

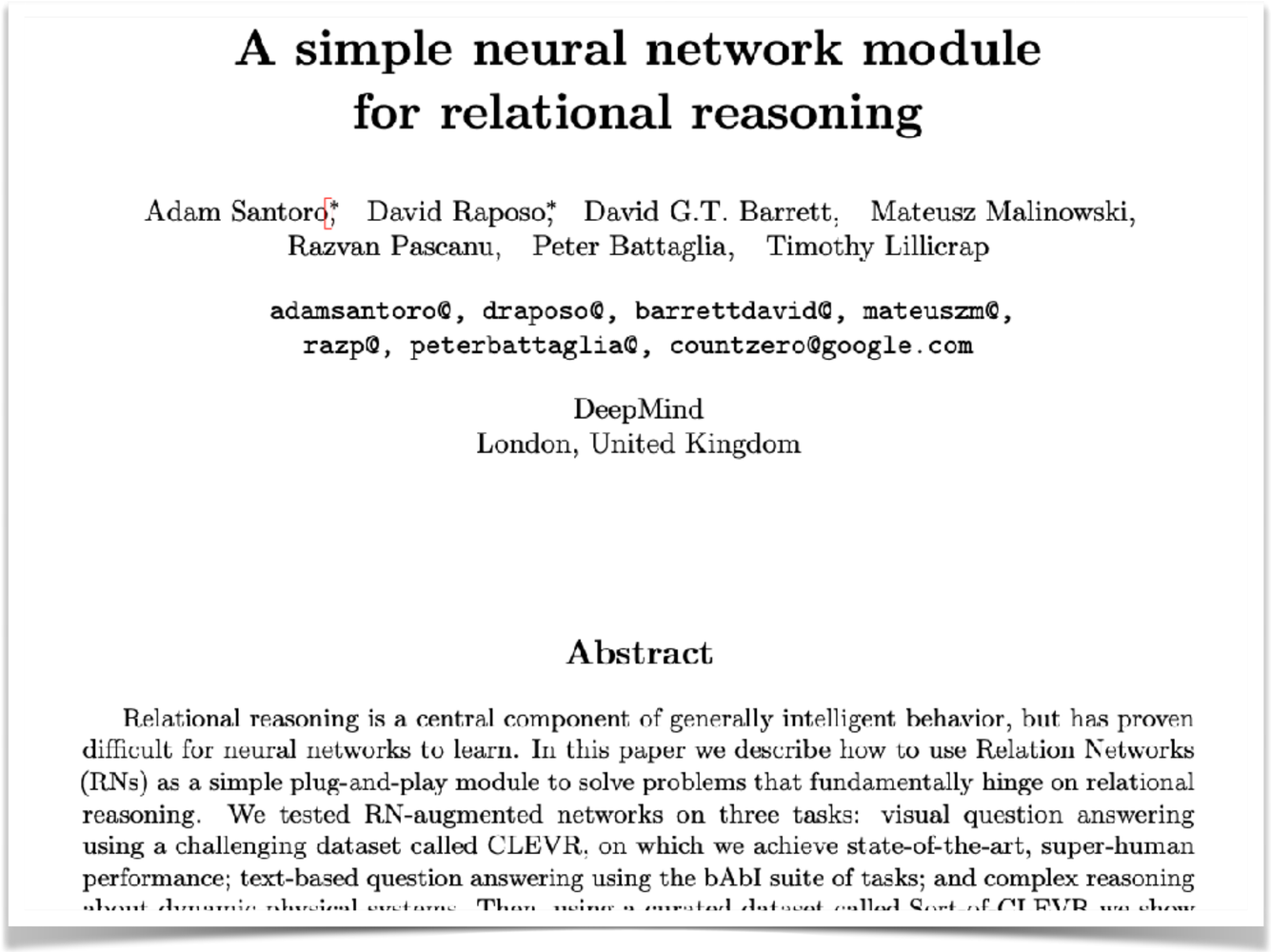


[Ahi21]

Overview

- Motivation
- Research
- **Evaluating Other Approaches**
- Model Modification
- Model Augmentation

Paper: Relational Reasoning



A simple neural network module for relational reasoning

Adam Santoro,* David Raposo,* David G.T. Barrett, Mateusz Malinowski,
Razvan Pascanu, Peter Battaglia, Timothy Lillicrap

adamsantoro@, draposo@, barrettdavid@, mateuszm@,
razp@, peterbattaglia@, countzero@google.com

DeepMind
London, United Kingdom

Abstract

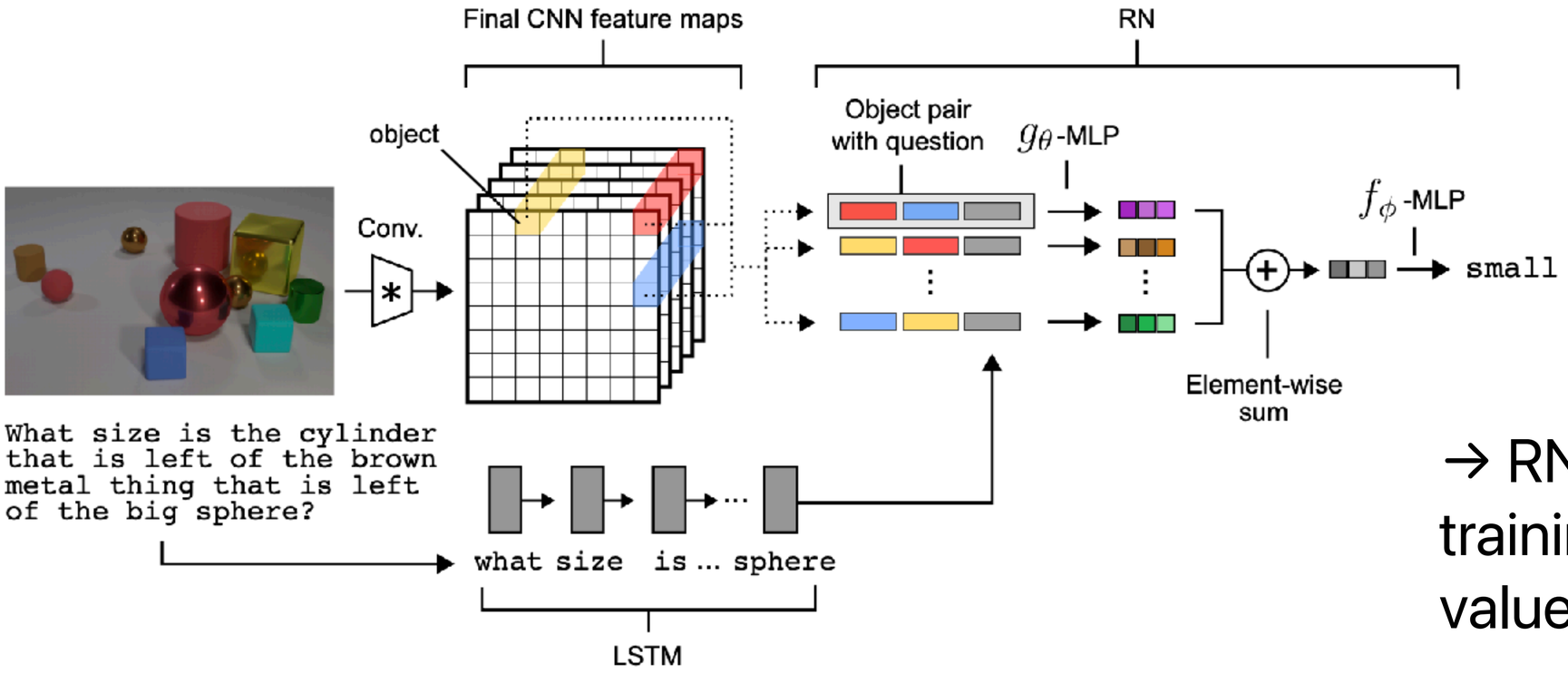
Relational reasoning is a central component of generally intelligent behavior, but has proven difficult for neural networks to learn. In this paper we describe how to use Relation Networks (RNs) as a simple plug-and-play module to solve problems that fundamentally hinge on relational reasoning. We tested RN-augmented networks on three tasks: visual question answering using a challenging dataset called CLEVR, on which we achieve state-of-the-art, super-human performance; text-based question answering using the bAbI suite of tasks; and complex reasoning about dynamic physical systems. Then, using a curated dataset called Sort-of-CLEVR we show

- Published 2017
- Direct Link: https://arxiv.org/abs/1706.01427
- **Using Relational Networks (RNs) to solve problems of relational reasoning**

[San17]

Paper: Relational Reasoning

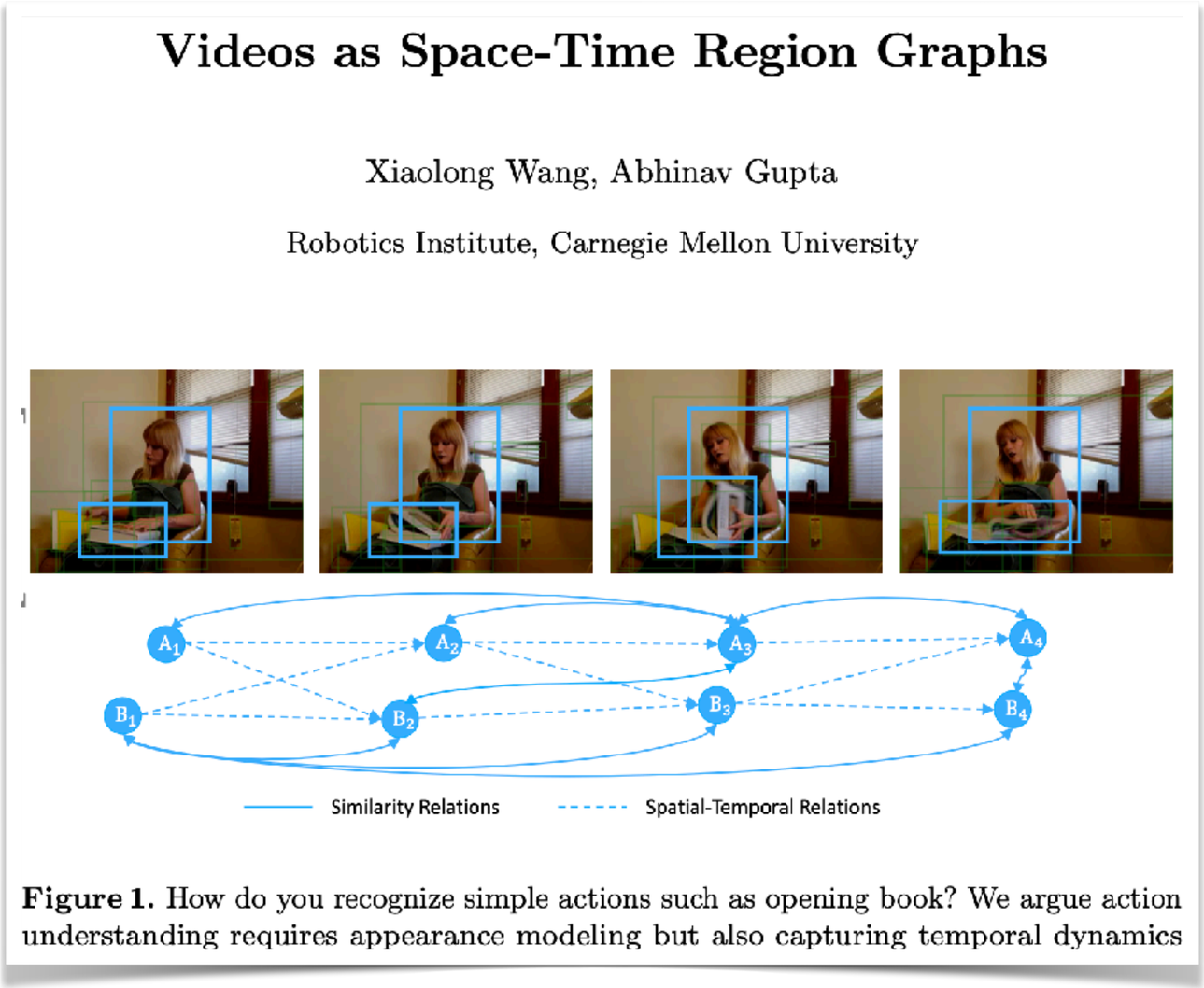
- **Relational reasoning**: logical thinking about relationships between objects
- Extending a deep learning architecture with **Relation Networks (RNs)**



→ RNs not suitable for training with exact distance values (no regression)

[San17]

Paper: Space-Time Region Graphs

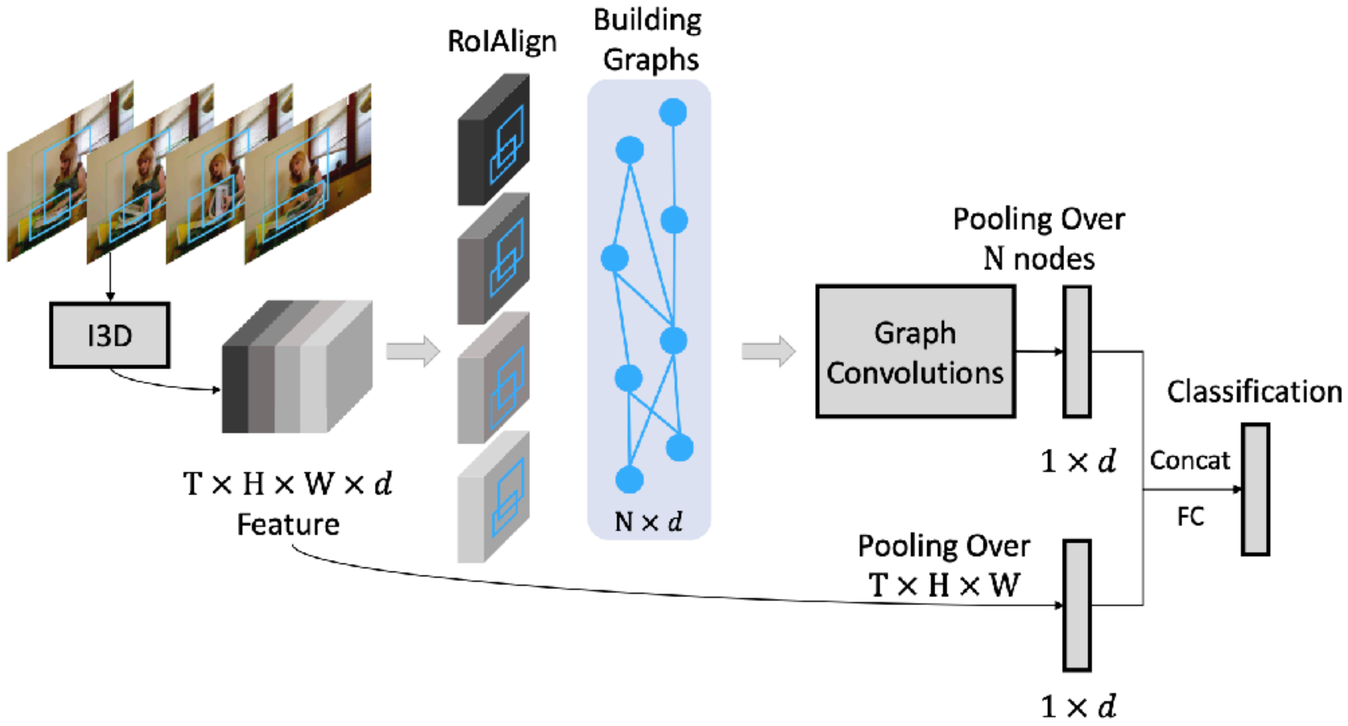


- Published 2018
- Direct Link: https://arxiv.org/abs/1806.01810
- **Representing videos as space-time region graphs to recognize actions**

[Wan18]

Paper: Space-Time Region Graphs

- Space-Time Region Graphs:
  - **Nodes**: object regions from different video frames
  - **Edges**: changes in shape or object relationship



→ Space-Time Region Graphs are optimized for relationship, not exact distance

[Wan18]

Overview

- Motivation
- Research
- Evaluating Other Approaches
- **Model Modification**
- Model Augmentation

Data Set

- **KITTI** 3D Object Detection distance estimation
  - 15.000 images
  - 80.000 labeled objects
  - Various object types

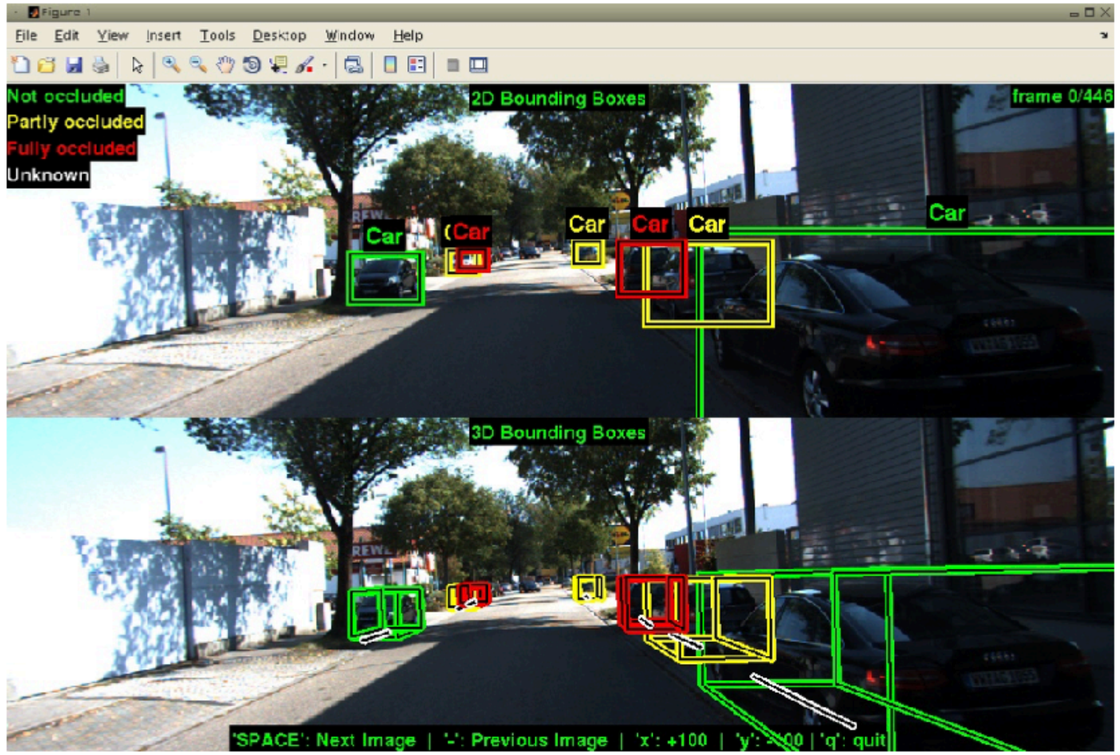


Image Source: http://www.cvlibs.net/publications/Geiger2013IJRR.pdf

[Gei17]

Model Modification

- Already suggested models: VGG19, DenseNet121, and ResNet50
  → **Addition of Xception and InceptionV3**

Keras

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	0.790	0.945	22,910,480	126	109.42	8.06
⋮							
VGG19	549	0.713	0.900	143,667,240	26	84.75	4.38
ResNet50	98	0.749	0.921	25,636,712	-	58.20	4.55
⋮							
InceptionV3	92	0.779	0.937	23,851,784	159	42.25	6.86

(Extreme Version of Inception)

- Keras models: Pre-trained for image classification (ImageNet)
- Requirements: Higher input image sizes

[Ker22]

Model Comparison

(a) Statistical performance for the CSEN method

	ARD ↓	SRD ↓	RMSE ↓	RMSE_{log} ↓	# params
DenseNet121	0.3308 ± 0.036	3.0487 ± 0.534	6.9950 ± 0.367	0.5370 ± 0.217	3,326
VGG19	0.3401 ± 0.039	3.1667 ± 0.563	7.2027 ± 0.331	0.6763 ± 0.264	3,326
ResNet50	0.2835 ± 0.035	2.2479 ± 0.437	6.2142 ± 0.333	0.5074 ± 0.155	3,326
Xception	0.272 ± 0.005	2.152 ± 0.073	6.229 ± 0.026	0.390 ± 0.013	3,326
InceptionV3	0.320 ± 0.004	2.880 ± 0.059	6.838 ± 0.029	0.507 ± 0.017	3,326

- Comparable performance of Xception & ResNet50
  → but deviation is lower
- Performance of InceptionV3 is solid
  → lower number of parameters than VGG19

[Ahi21, Git21]

Overview

- Motivation
- Research
- Evaluating Other Approaches
- Model Modification
- **Model Augmentation**

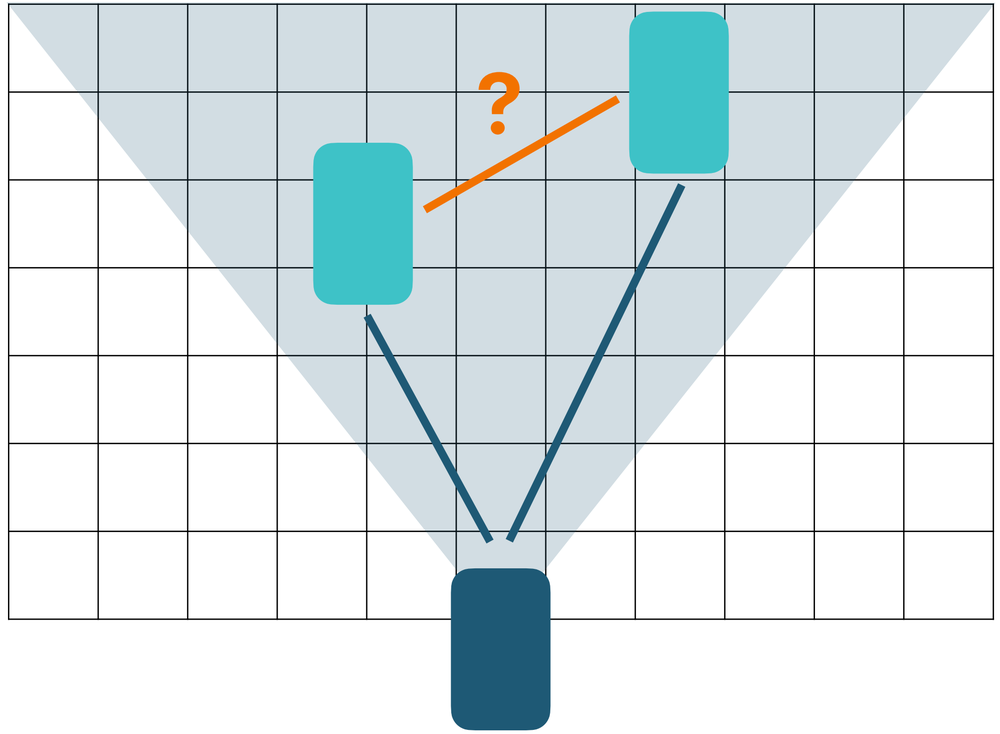
Adding Own Augmentation

- **Distance between objects**

Image

[Ahi21]

Topview



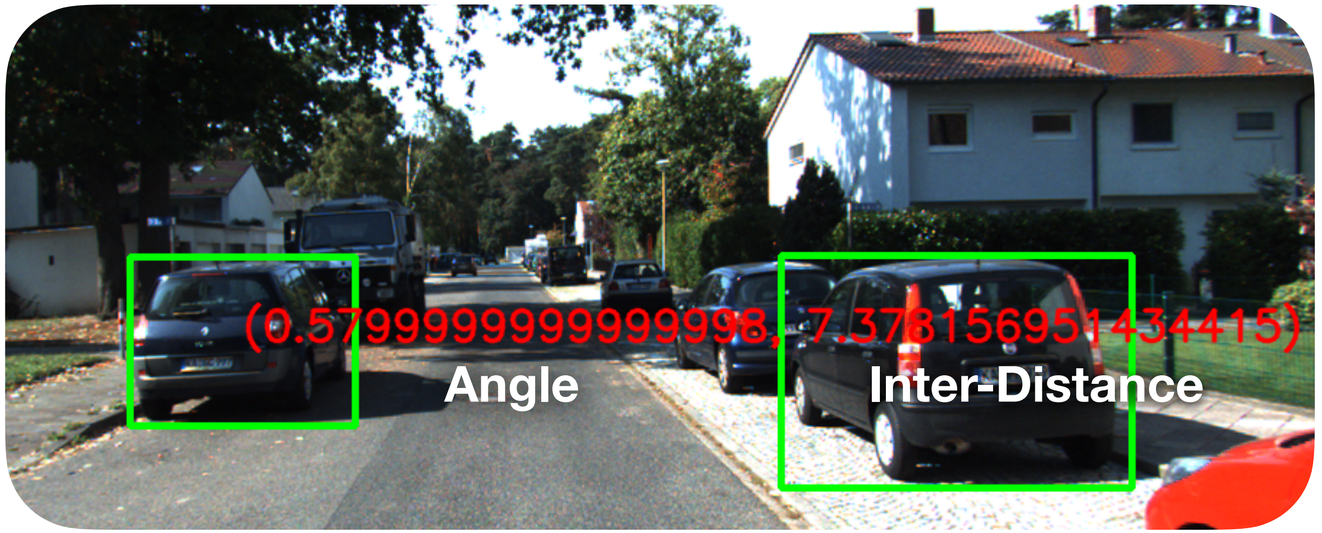
Requirements

- Same as paper (forked repository):
  - Python + Tensorflow & OpenCV
  - MATLAB
  - Data

[Git21, Git22]

Data Preprocessing

- Adapted scripts from original models
- Extract objects pairwise
- Compute object features



[Git21, Git22]

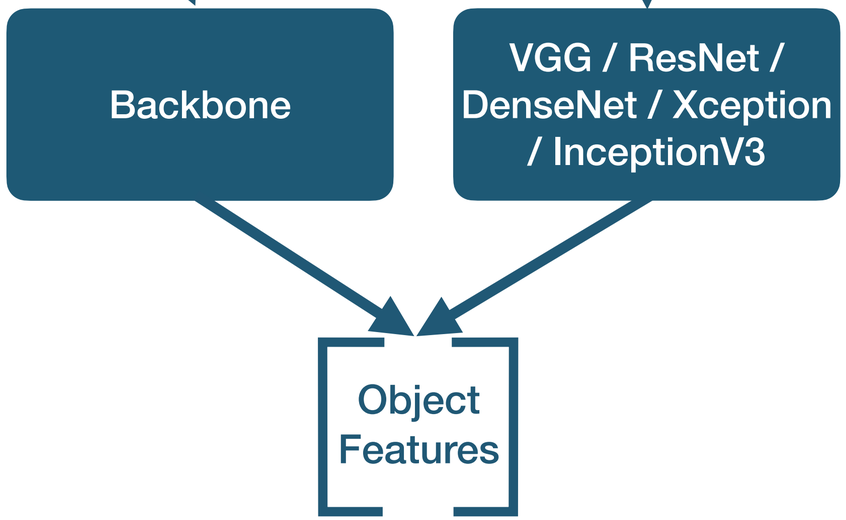
Data Preprocessing

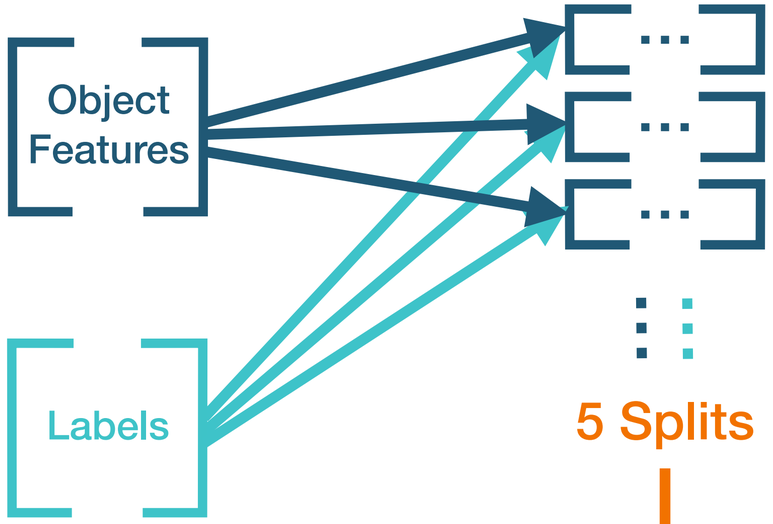
- Preprocessing result ~140,000 pairs

	A	B	C	D	E	F	G	H	I	J
1	filename	object_ids	x1	y1	z1	x2	y2	z2	obj distance 2D	obj distance 3D
2	000001.txt	(1, 2)	0.47	1.49	69.44	-16.53	2.39	58.49	20.22133774012	20.2413561798611
3	000001.txt	(1, 3)	0.47	1.49	69.44	4.59	1.32	45.84	23.9569280167554	23.9575311749771
4	000001.txt	(2, 3)	-16.53	2.39	58.49	4.59	1.32	45.84	24.6186291251158	24.6418708705325
5	000002.txt	(4, 5)	3.23	1.59	8.55	3.18	2.27	34.38	25.8300483932957	25.838997658578
6	000004.txt	(7, 8)	-15.71	2.16	38.26	-15.89	2.23	51.17	12.9112547802295	12.9114445357597
7	000006.txt	(10, 11)	-2.72	0.82	48.22	-2.61	1.13	31.73	16.4903668849422	16.4932804499287
8	000006.txt	(10, 12)	-2.72	0.82	48.22	-12.54	1.64	19.72	30.1443593396841	30.1555102759015
9	000006.txt	(10, 13)	-2.72	0.82	48.22	-12.66	1.13	38.44	13.944604691421	13.9480500429272
10	000006.txt	(11, 12)	-2.61	1.13	31.73	-12.54	1.64	19.72	15.5834848477483	15.5918279877633
11	000006.txt	(11, 13)	-2.61	1.13	31.73	-12.66	1.13	38.44	12.0841466392956	12.0841466392956
12	000006.txt	(12, 13)	-12.54	1.64	19.72	-12.66	1.13	38.44	18.7203846114336	18.7273302955867
13	000007.txt	(14, 15)	-0.69	1.69	25.01	-7.43	1.88	47.55	23.5261386546964	23.5269058739138
14	000007.txt	(14, 16)	-0.69	1.69	25.01	-4.71	1.71	60.52	35.7368227462935	35.7368283427615
15	000007.txt	(14, 17)	-0.69	1.69	25.01	-12.63	1.88	34.09	15.0003333296297	15.0015365879633
16	000007.txt	(15, 16)	-7.43	1.88	47.55	-4.71	1.71	60.52	13.2521432228904	13.2532335676996
17	000007.txt	(15, 17)	-7.43	1.88	47.55	-12.63	1.88	34.09	14.42953914718	14.42953914718
18	000007.txt	(16, 17)	-4.71	1.71	60.52	-12.63	1.88	34.09	27.5911453187431	27.5916690325178
19	000008.txt	(18, 19)	-2.7	1.74	3.68	-1.17	1.65	7.86	4.45121331773709	4.45212308904415

[Git21, Git22]

Prototype: Data Processing





5 Splits

5 Runs

- CL-CSEN
- CSEN

[Git21, Git22]

Metrics

Absolute Relative Distance

$$ARD = \frac{1}{N}\sum_{i=1}^{N}(|\hat{d}_i - d_i|/d_i)$$

Root of Mean Squared Error

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{d}_i - d_i)^2}$$

Squared Relative Distance

$$SRD = \frac{1}{N}\sum_{i=1}^{N}((\hat{d}_i - d_i)^2/d_i)$$

Root of Mean Squared log-Error

$$RMSE_{log} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(log\,\hat{d}_i - log\,d_i)^2}$$

Statistical Performance

- CSEN:

Backbone	ARD ↓	SRD ↓	RMSE ↓	RMSE_{log} ↓	# params
DenseNet121	0.713 ± 0.012	7.936 ± 0.204	9.732 ± 0.016	0.657 ± 0.002	3,326
VGG19	0.738 ± 0.009	8.669 ± 0.147	10.086 ± 0.03	0.684 ± 0.01	3,326
ResNet50	0.719 ± 0.006	8.148 ± 0.115	9.566 ± 0.033	0.654 ± 0.002	3,326

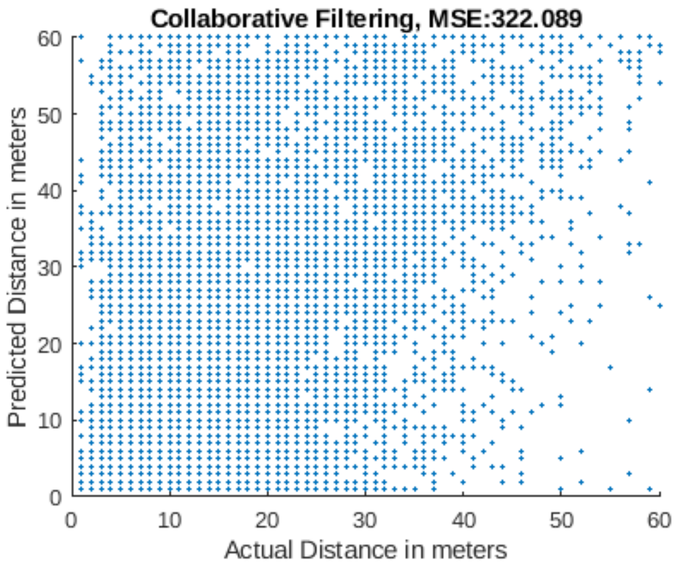
- CL-CSEN:

Backbone	ARD ↓	SRD ↓	RMSE ↓	RMSE_{log} ↓	# params
DenseNet121	0.653 ± 0.006	6.836 ± 0.105	8.208 ± 0.022	0.592 ± 0.002	1,233,326
VGG19	0.55 ± 0.004	5.032 ± 0.053	7.294 ± 0.032	0.533 ± 0.003	618,926
ResNet50	0.678 ± 0.006	7.204 ± 0.116	8.517 ± 0.043	0.605 ± 0.002	2,462,126

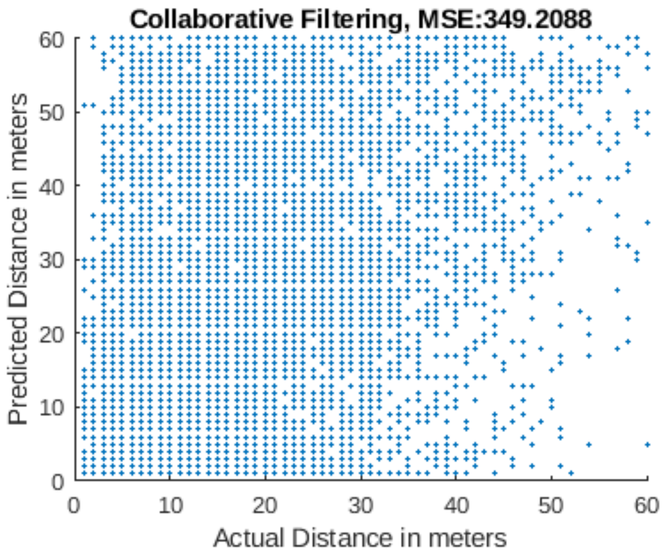
[Git21, Git22]

Statistical Performance

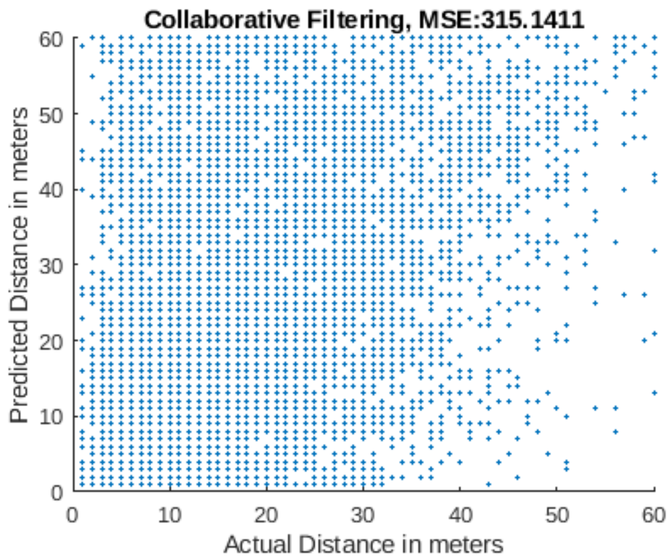
- DenseNet121:



- VGG19:

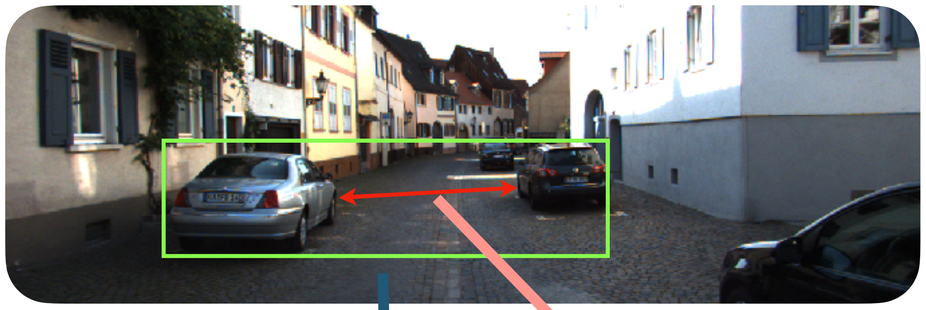


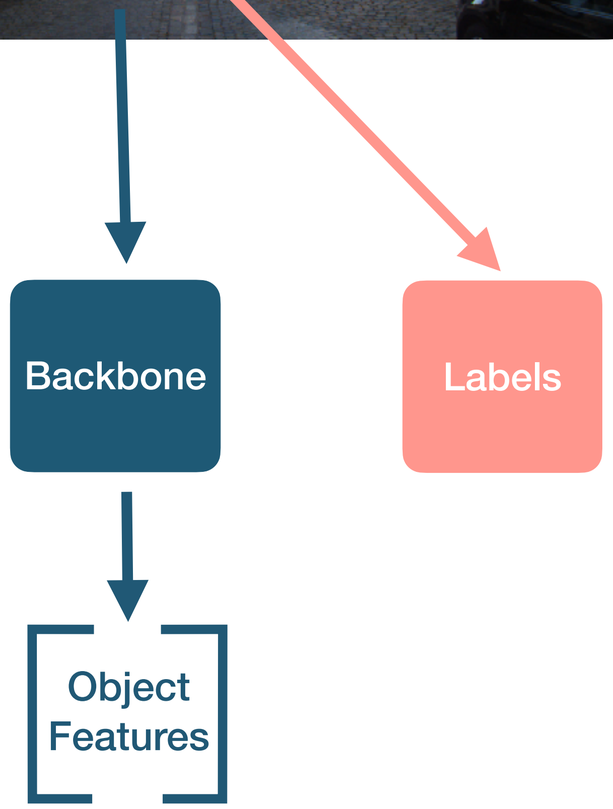
- ResNet50:

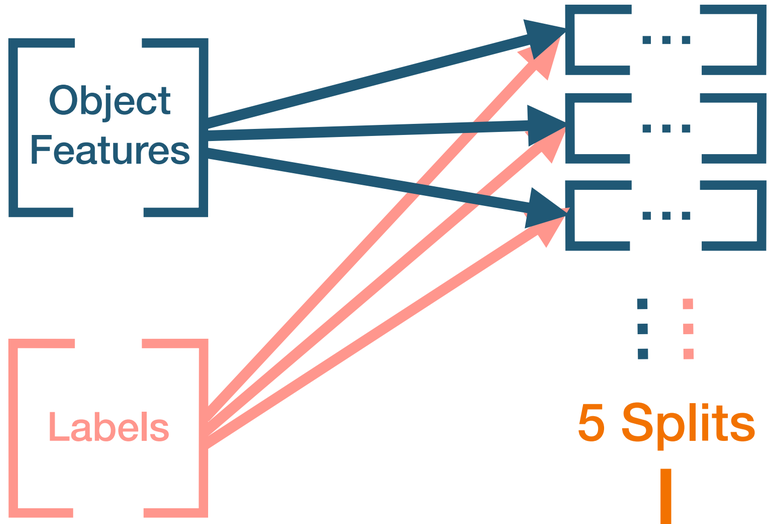


[Git21, Git22]

System Design







5 Splits

5 Runs

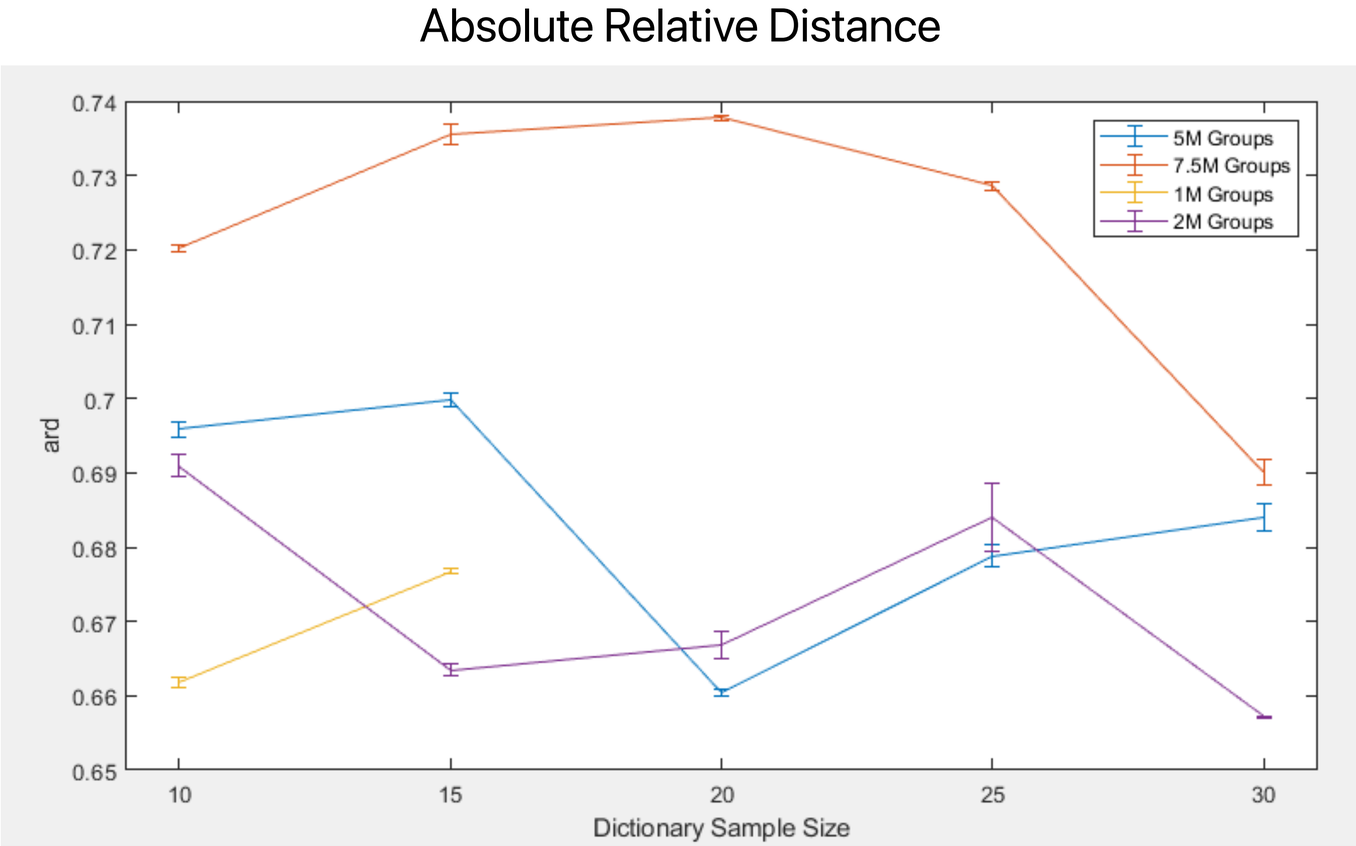
- CL-CSEN
- CSEN

System Design

- Maintains more information
- Comparable computational effort
- Straight forward implementation

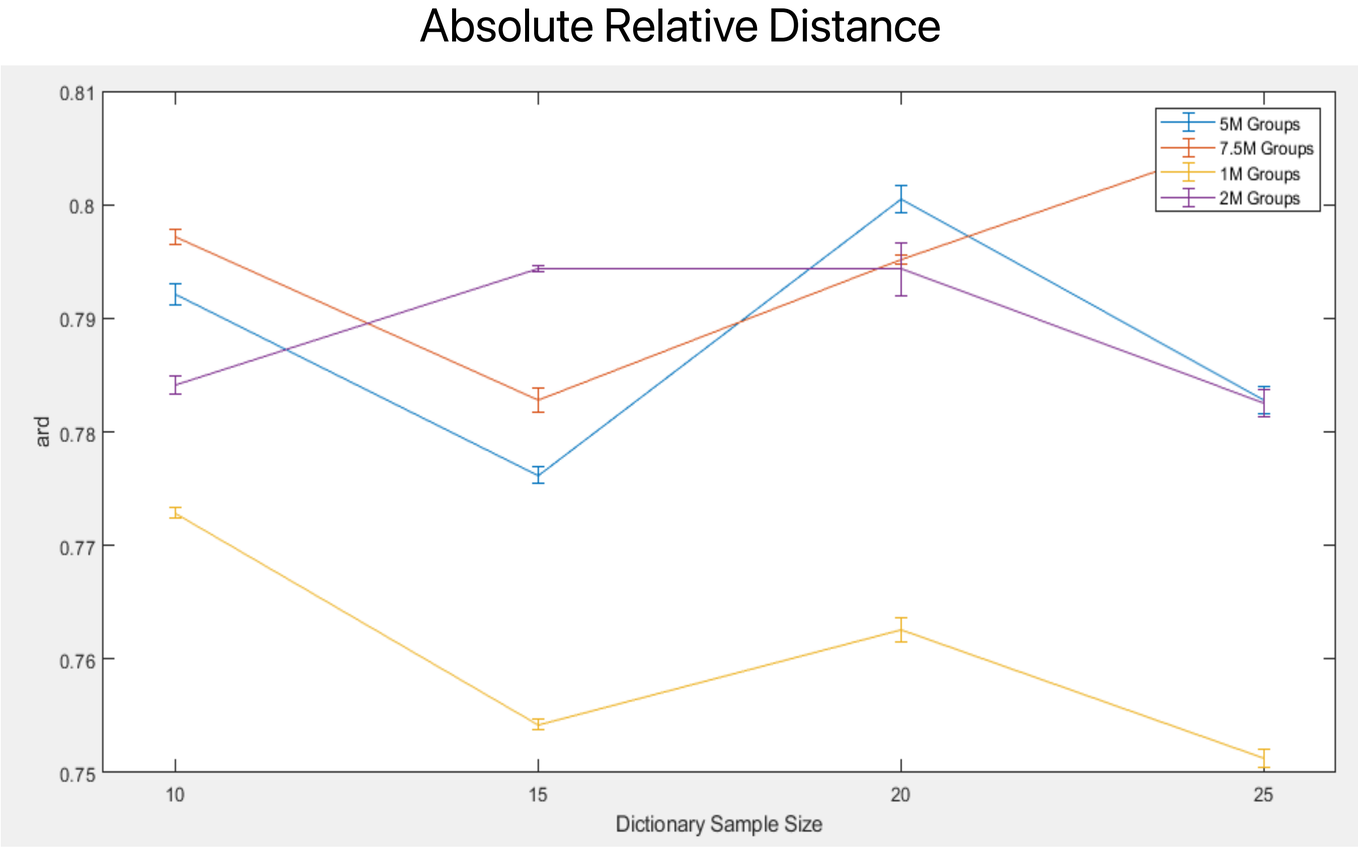
→ Indeed, slightly better results

Statistical Performance



- Variation of number of groups and dictionary sample sizes
- More groups could be better
- Maybe, bigger dictionary sample sizes are slightly advantageous
- Best case: ARD=0.66 for the 2 meter group with a dictionary sample size of 30

Statistical Performance



- Variation of number of groups and dictionary sample sizes
- More groups seem to be better
- No clear trend in terms of dictionary sample size
- Best case: ARD=0.75 for the 1 meter group with a dictionary sample size of 25

Statistical Performance

- Evaluation of Inception and Xception approach
- Inferior for all metrics
- Interesting: Small variance

Evaluated on CSEN

Backbone	ARD ↓	SRD ↓	RMSE ↓	RMSE_{log} ↓
DenseNet121	0.713 ± 0.012	7.936 ± 0.204	9.732 ± 0.016	0.657 ± 0.002
VGG19	0.738 ± 0.009	8.669 ± 0.147	10.086 ± 0.03	0.684 ± 0.01
ResNet50	0.719 ± 0.006	8.148 ± 0.115	9.566 ± 0.033	0.654 ± 0.002
Xception	0.749 ± 0.0002	8.764 ± 0.095	9.992 ± 0.034	0.688 ± 0.0005
Inception V3	0.772 ± 0.0004	9.094 ± 0.116	10.053 ± 0.009	0.705 ± 0.002

[Git21, Git22]

Conclusion

- Succesful implementation
- Performance can be optimized
- Future work:

  →Use angle between objects

Sources

- **[Ahi21]** Mete Ahishali et al. *Representation Based Regression for Object Distance Estimation*, 2021. URL: https://arxiv.org/abs/2106.14208v1, last visited 08.02.2022.
- **[Gei17]** Andreas Geiger et al. *3D Object Detection Evaluation*, 2017. URL: http://cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d, last visited 08.02.2022.
- **[Git21]** GitHub. Mete Ahishali et al. *Representation Based Regression for Object Distance Estimation*, 2021. URL: https://github.com/meteahishali/CSENDistance, last visited 08.02.2022.
- **[Git22]** GitHub. Tim Rosenkranz, Benedikt Schröter, Carla Frenzel, Nils Möbus. *Fork of [Git21]*, 2021. URL: https://github.com/Tim-orius/CSENDistance, last visited 08.02.2022.
- **[Ker22]** Keras. *Keras Applications*, 2022. URL: https://keras.io/api/applications/, last visited 08.02.2022.
- **[San17]** Adam Santoro et al. *A simple neural network module for relational reasoning*, 2017. URL: https://arxiv.org/abs/1706.01427, last visited 08.02.2022.
- **[Wan18]** Xiaolong Wang et al. *Videos as Space-Time Region Graphs*, 2018. URL: https://arxiv.org/abs/1806.01810, last visited 08.02.2022.
- **[Zhu19]** Jing Zhu et al. *Learning Object-Specific Distance From a Monocular Image*, 2019. URL: https://arxiv.org/abs/1909.04182, last visited 08.02.2022.