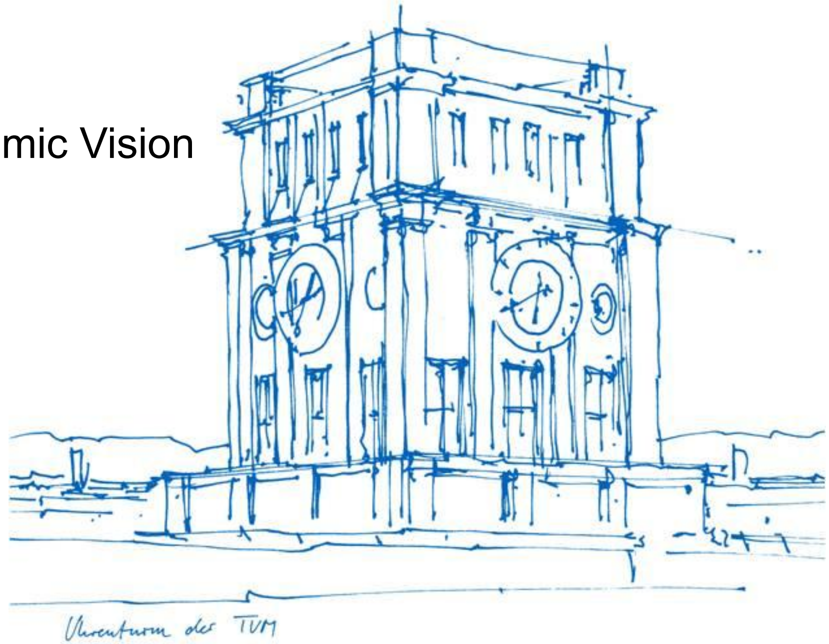
Multi-object tracking 1 - Syn2real
Final Presentation

Students: Felix Dietrich, Yiming Zhang

Supervisor: Tim Meinhardt

Date: 09.02.2022

Advanced Deep Learning for Computer Vision: Dynamic Vision
Technische Universität München
TUM School of Informatics
Winter 2021/22



Multi-Object Tracking

Applications: Ranging from autonomous driving to urban surveillance

MOT Tasks: Pedestrian detection, re-identification, and tracking

MOTS20 Dataset [1]

[1] *Voigtlaender et al., MOTS: Multi-Object Tracking and Segmentation. arXiv: 1902.03604*

Challenges With Real-World Data

- Privacy concerns (GDPR)
- Human annotators
  - Very slow and inefficient
  - Annotation mistakes (noise and errors)
  - Costly
  - Large crowded datasets infeasible

⇒ **Small datasets with noisy ground truth**

MOTS20 Dataset

But we NEED lots of data!

MOTSynth - Synthetic Data as Solution

MOTSynth [2] data based on the GTA V video game

MOTSynth preview video [3]

[2] *Fabbri et al., MOTSynth: How Can Synthetic Data Help Pedestrian Detection and Tracking? ICCV 2021*
[3] *Video source: www.youtube.com/watch?v=dc_Z1iCceL4*

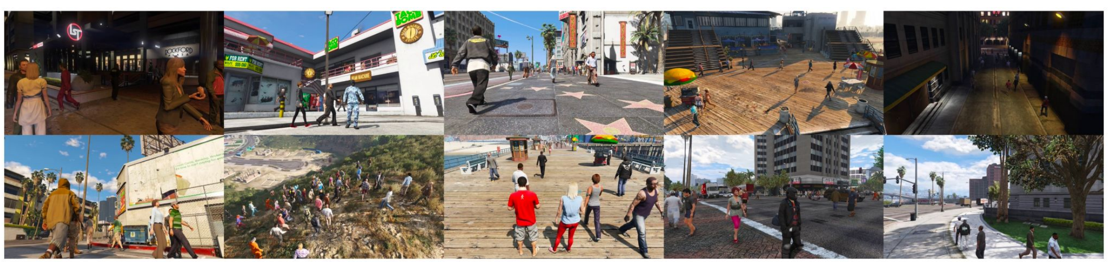
MOTSynth - Advantages

Free high quality annotations without noise

(a) Bounding boxes and pose (b) Segmentation masks (c) Depth maps

Privacy

Many different controlled environments (variety)



Large dataset

Dataset	#Frames	#Inst.	3D	Pose	Segm.	Depth
PoseTrack [6]	46k	276k		✓		
MOTS [84]	2k	26k			✓	
MOT-17 [61]	11k	292k				
MOT-20 [22]	13k	1,652k				
VIPER [49]	254k	2,750k	✓		✓	
GTA [50]	250k	3,875k			✓	✓
JTA [27]	460k	15,341k	✓	✓		
MOTSynth	1,382k	40,780k	✓	✓	✓	✓

MOTSynth - Synthetic to Real Gap

- GTA V is not reallife :(
- **Question:** Can synthetic data replace real-world data for deep learning?
    - MOTSynth claims it can

MOTSynth: Synthetic Data

syn2real gap

MOTS20: Real-World Data

Challenging MOTSynth Segmentation

MOTSynth paper: *"Our Tracktor Mask R-CNN trained **only** on synthetic data significantly outperforms TrackR-CNN, that is trained on COCO"* (COCO [4], Tracktor [5])

But analysis of multi-object segmentation is missing from MOTSynth paper

Questions:

- How does MOTSynth compare to COCO in regards to segmentation?
- Can MOTSynth bridge the syn2real gap for segmentation?

⇒ Tasks:

1. **Evaluate pretrained COCO and MOTSynth models on MOTS20**
2. **Finetune pretrained models on MOTS20**
3. **Analyze syn2real gap (compare finetuning to no-finetuning)**

[4] *Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.*
[5] *Bergmann, P., Meinhardt, T., & Leal-Taixe, L. (2019). Tracking without bells and whistles. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 941-951).*

Roadmap

3.11.2021	**Project Assignment**
12.11.2021	**Initial Report**
8.12.2021	**First Presentation**
19.01.2022	**Second Presentation**
9.02.2022	**Final Presentation**

Until first presentation:

- Analyze neural rendering
- Setup codebase on Google Colab
- First evaluations of pretrained models (segmentation)

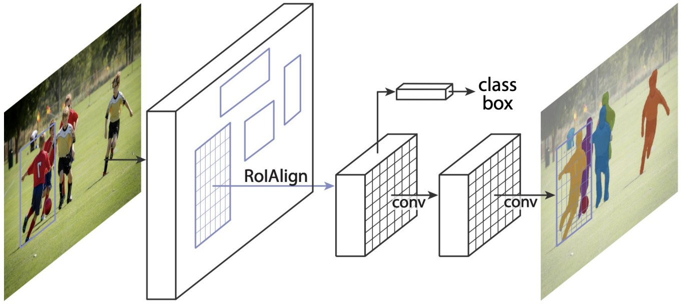
Until second presentation:

- Hyperparameter search (score threshold and learning rate)
- Finetuned pretrained COCO and MOTSynth using 4-fold cross-validation
- Compared finetuned to no-finetuned results

Until final presentation:

- Tried to boost finetuning results for models pretrained on MOTSynth (fairly)
- Finalized segmentation evaluation
- Reproduced MOTS results from the MOTSynth paper and extend it

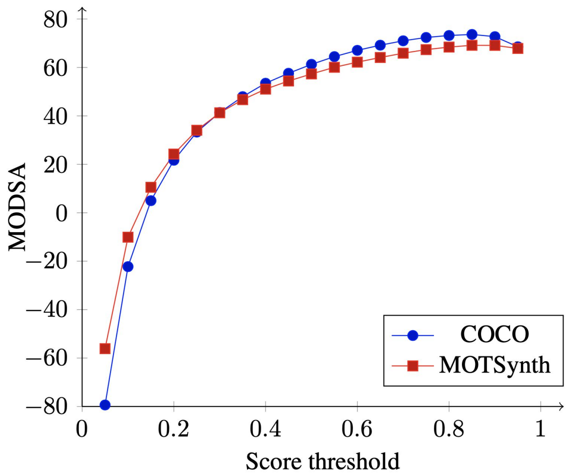
Score Threshold Search



- Score threshold for returning bounding box
- Hyperparameter of **Mask R-CNN** [6] model
- Pretrained on **COCO** or **MOTSynth** evaluated on **MOTS20** training sequences
- Main metric: **MODSA** (Mask-overlap based Multi-Object Detection Accuracy, mask overlap IoU)

Score Threshold:

- **Default:** 0.05 (negative MODSA)
- **Best:** 0.85 (best for both models)



[6] *He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).*

4-Fold Cross-Validation

We only have access to the MOTS20 training set

⇒ 4-fold cross-validation (average validation results from the 4 splits)

	Fold 1	Fold 2	Fold 3	Fold 4
Split 1	Fold 1	Fold 2	Fold 3	Fold 4
Split 2	Fold 1	Fold 2	Fold 3	Fold 4
Split 3	Fold 1	Fold 2	Fold 3	Fold 4
Split 4	Fold 1	Fold 2	Fold 3	Fold 4

4 fold cross-validation [7]

Sample	Name	FPS	Resolution	Length	Tracks	Boxes	Density	Description
	MOTS20-11	30	1920x1080	900 (00:30)	62	8511	9.5	Forward moving camera in a busy shopping mall
	MOTS20-09	30	1920x1080	525 (00:18)	26	4774	9.1	A pedestrian street scene filmed from a low angle.
	MOTS20-05	14	640x480	837 (01:00)	103	6570	7.8	Street scene from a moving platform
	MOTS20-02	30	1920x1080	600 (00:20)	37	7039	11.7	People walking around a large square.
	Total			2862 frm. (128 s.)	26894	9.4		

MOTS20: 4 sequences [8]

[7] *Image source:* *https://scikit-learn.org/stable/modules/cross_validation.html*
[8] *Image source:* *https://motchallenge.net/data/MOTS/*

Fine-tuning Experiments

Dataset	FT	MODSA ↑
COCO	✗	73.82
	M✓	74.56
MOTSynth	✗	69.20
	M✓	69.71
	BB=5✓	71.56
	BB=3✓	72.02

- Score threshold: 0.85
- Fine-tune for 15 epochs on **MOTS20**
- Evaluate on MOTS20 using 4-fold cross validation

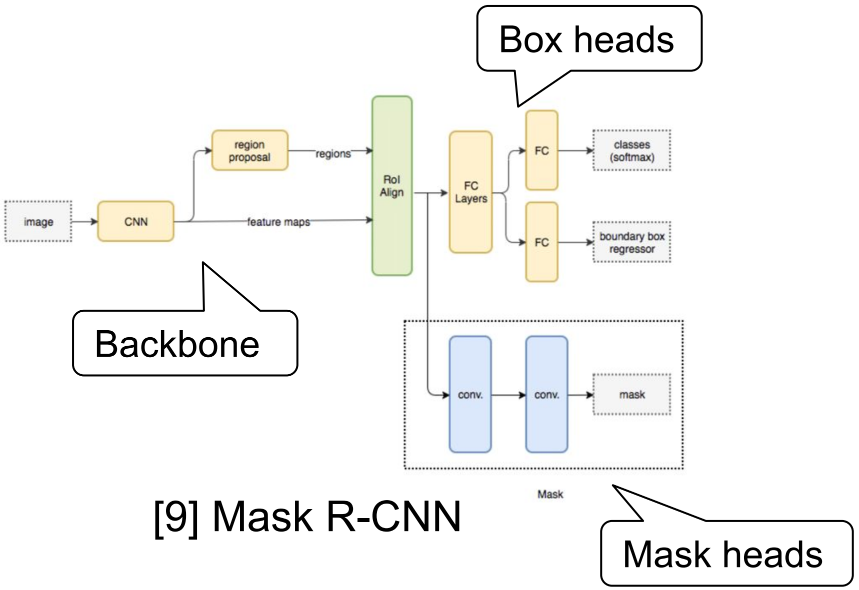
Fine-tuning (FT) Configurations:

✗ No fine-tuning

BB=3✓ Fine-tune up to last 3 back-bone layers

BB=5✓ Fine-tune everything

M✓ Fine-tune only the mask heads



[9] Mask R-CNN

[9] *Gonzalez, S., Arellano, C., & Tapia, J. E. (2019). Deepblueberry: Quantification of blueberries in the wild using instance segmentation. Ieee Access, 7, 105776-105788.*

Segmentation Evaluation

Dataset	FT	MODSA ↑
COCO	✗	73.82
	M✓	74.56
MOTSynth	✗	69.20
	M✓	69.71
	BB=5✓	71.56
	BB=3✓	72.02

- Pretrained COCO improved only through fine-tuning the mask heads

  Possible Reasons:
  - MOTS20 ignore regions might punish FP box predictions
  - COCO is a real-world dataset but not a MOT dataset
  - COCO is already performing very well

- Pretrained MOTSynth improved best with fine-tuning up to the last 3 back-bone layers

  Possible Reasons:
  - MOTSynth is a MOT dataset
  - Lots to improve from real data
  - MOTSynth might be able to replace low level features (first 2 back-bone layers) but not more high-level ones

Image source: MOTS20

Multi-object Tracking and Segmentation Evaluation

Dataset	Tracktor Model	FT	sMOTSA ↑	MOTSA ↑	MOTSP ↑	MODSA ↑	MODSP ↑	IDF1 ↑	TP ↑	FP ↓	FN ↓	IDS ↓
COCO	FRCNN	✗	55.58	68.80	81.93	69.39	82.59	63.24	19677	1016	7217	159
	Mask R-CNN	✗	59.75	73.46	81.99	73.88	82.58	69.93	20475	606	6419	112
	Mask R-CNN	M✓	61.59	74.18	83.54	74.60	84.12	70.15	20564	501	6330	114
MOTSynth	FRCNN	✗	55.54	68.73	81.93	69.31	82.41	63.09	19626	987	7268	155
	Mask R-CNN	✗	56.10	69.52	81.67	70.02	82.08	65.96	19687	855	7207	136
	Mask R-CNN	M✓	57.41	70.35	82.43	70.84	82.84	66.30	19801	750	7093	132
	Mask R-CNN	BB=5✓	58.18	70.83	82.82	71.33	83.23	65.46	19805	622	7089	135
	Mask R-CNN	BB=3✓	58.74	71.47	82.87	71.98	83.30	65.35	20000	641	6894	137

- Reproduced MOTSynth’s paper results (Table 7 row 1, 4, and 5)
- We added our fine-tuning experiments and the MODSA/MODSP columns
- First we predict MOT outputs then we do mask segmentation prediction (MOTS)
- **Fine-tuning improvements from segmentation transfer to MOTS**
    - MOTS gap is smaller but still there

MOTSynth to COCO Comparison

How does MOTSynth compare to COCO in regards to segmentation?

- COCO outperforms MOTSynth, also after finetuning on MOTS20 → There is a gap
  - Finetuning reduces that MODSA gap from **-4.62** to **-2.54**
- Finetuning:
  - Pretrained COCO improves little (+0.74)
  - Pretrained MOTSynth improves much (+2.82)

Dataset	FT	MODSA ↑
COCO	✗	73.82
	M✓	74.56
MOTSynth	✗	69.20
	M✓	69.71
	BB=5✓	71.56
	BB=3✓	72.02

	MODSA		
Pretrained Dataset	Pretrained	Best Finetuned	Change
COCO	73.82	74.56	+0.74
MOTSynth	69.20	72.02	+2.82
Gap	**-4.62**	**-2.54**	

Synthetic to Real Gap

Can MOTSynth bridge the syn2real gap for segmentation?

- Until now: **No**
  - The claims of the MOTSynth paper that synthetic data can be used as a full replacement is currently not true for segmentation
  - But MOTSynth has room for improvement (e.g. optimize pretraining, etc.)
- Pretrained MOTSynth generalizes a lot from finetuning on the real MOTS20 dataset but COCO barely improves

	MODSA		
Pretrained Dataset	Pretrained	Best Finetuned	Change
COCO	73.82	74.56	+0.74
MOTSynth	69.20	72.02	+2.82
Gap	**-4.62**	**-2.54**	

Future Work

What can we do to reduce the synthetic to real gap in the future?

- Optimize MOTSynth pretraining (hyperparameters, etc.)
- Joint training strategy, mixing synthetic and real-world data
- Fine-tune box heads and back-bone on MOT17 first
  then fine-tune mask heads on MOTS20

Thank you for your attention!

References

[1] *Voigtlaender et al., MOTS: Multi-Object Tracking and Segmentation. arXiv: 1902.03604*

[2] *Fabbri et al., MOTSynth: How Can Synthetic Data Help Pedestrian Detection and Tracking? ICCV 2021*

[3] *Video source: www.youtube.com/watch?v=dc_Z1iCceL4*

[4] *Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.*

[5] *Bergmann, P., Meinhardt, T., & Leal-Taixe, L. (2019). Tracking without bells and whistles. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 941-951).*

[6] *He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).*

[7] *Image source: https://scikit-learn.org/stable/modules/cross_validation.html*

[8] *Image source: https://motchallenge.net/data/MOTS/*

[9] *Gonzalez, S., Arellano, C., & Tapia, J. E. (2019). Deepblueberry: Quantification of blueberries in the wild using instance segmentation. Ieee Access, 7, 105776-105788.*