Training neural networks

Today's lecture

- Learning from small data
- Active learning
- When you are not learning
- Surrogat losses

Curriculum:

- How transferable are features in deep neural networks? (http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf)
- Cost-Effective Active Learning for Deep Image Classification (https://arxiv.org/pdf/1701.03551.pdf)
- Tracking Emerges by Colorizing Videos (https://arxiv.org/abs/1806.09594)
- Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints (http://openaccess.thecvf.com/content_cvpr_2018/papers/Mahjourian_Unsupervised_Learning_of_CVPR_2018_paper.pdf)

Learning from small data

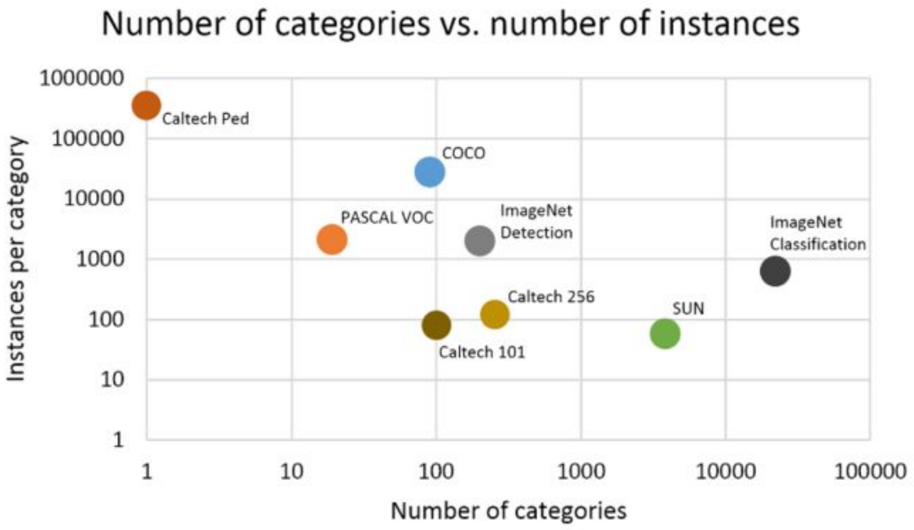
What is small data?

ImageNet challenge: 1.2 m images (14 m in full)

MSCOCO Detection challenge: 80,000 images (328,000 in full)

KITTI Road segmentation: 289 images

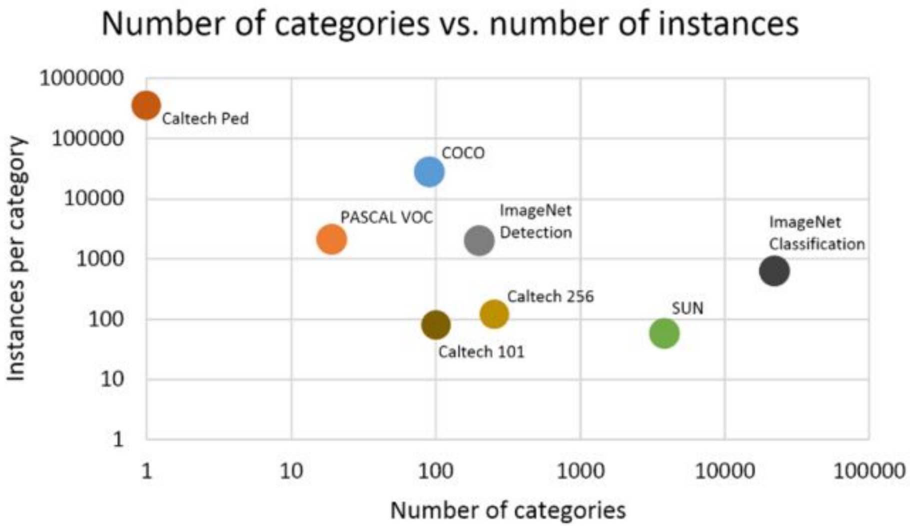
SLIVER07 3D liver segmentation: 20 3D-images



(d)

What is small data?

Sliver liver segmentation still works, why?



(d)

What is small data?

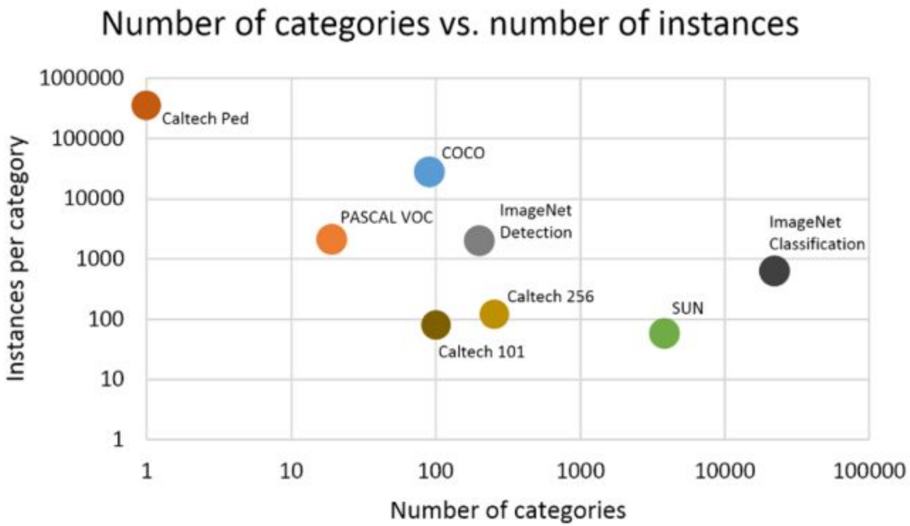
Sliver liver segmentation still works, why?

Homogenous data:

- Same CT-machine
- Standardised procedure

KITTI Road segmentation:

- Similar conditions
- Same camera
- Roads are very similar



(d)

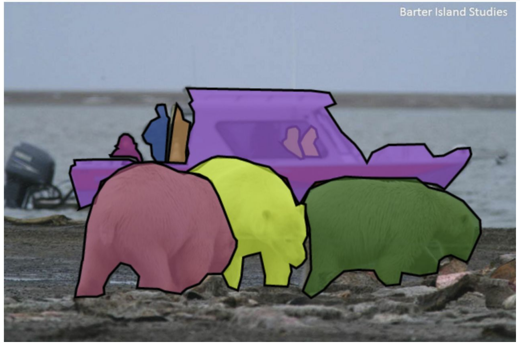
What is small data?

Heterogeneous task, need heterogeneous data.

It’s not not necessarily the amount of images that counts, but rather how many **different** images you have.

What is small data?

- ImageNet have unspecific labels
  - Harder to extract the essence of a given class
- MSCOCO have specific labels
  - Easier to learn how the pixels relate to a class

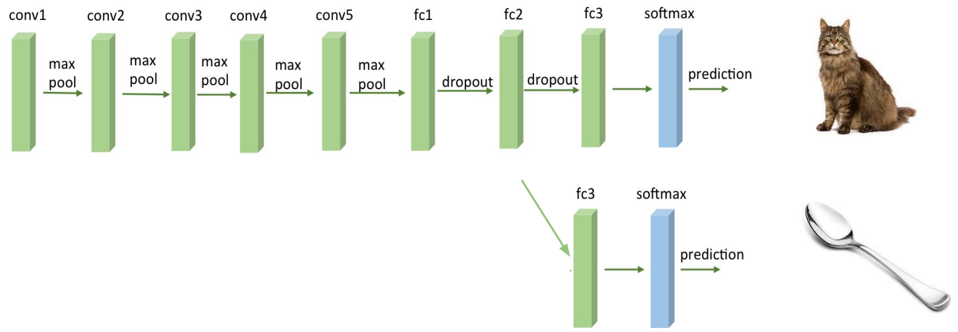


[What I learned from competing against a ConvNet on ImageNet]

[Explore MSCOCO]

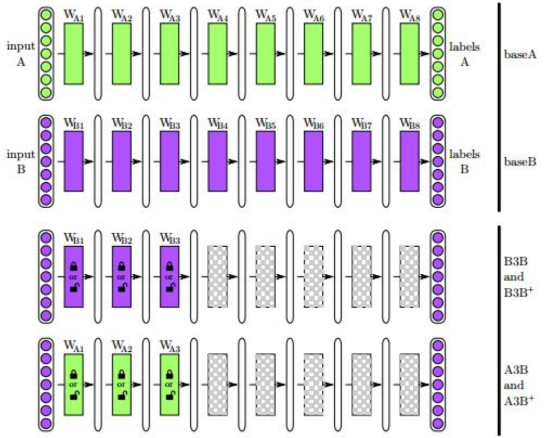
Transfer learning from pretrained network

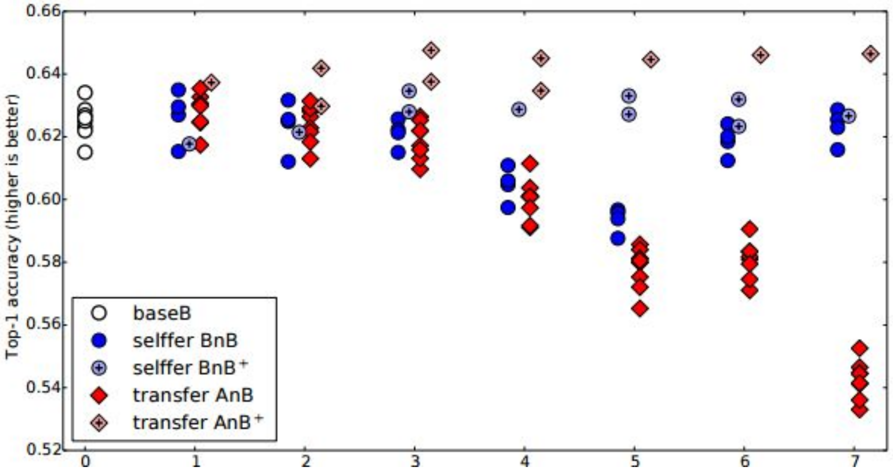
- Neural networks share representations across classes
- A network train on many classes and many examples have more general representation
- You can reuse these features for many different applications
- Retrain train the last layer of the network, for a different number of classes



Transfer learning: Study

- Study done with plentiful data (split ImageNet in two)
- Locking weights deprecate performance
    - Remember lots of data
- More data improves performance, even if it's different classes.



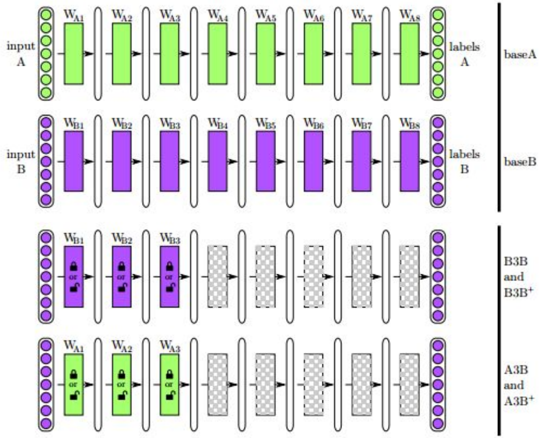


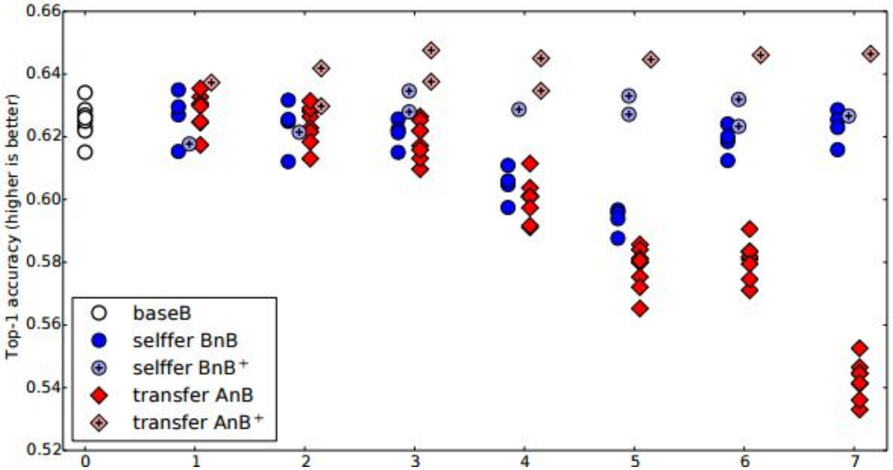
OBS! Everything may not be applicable with new initialization schemes, Resnet and batch-norm

[How transferable are features in deep neural networks?]

Transfer learning: Study

- Study done with plentiful data (split ImageNet in two)
- Locking weights deprecate performance
    - Remember lots of data
- More data improves performance, even if it's different classes!



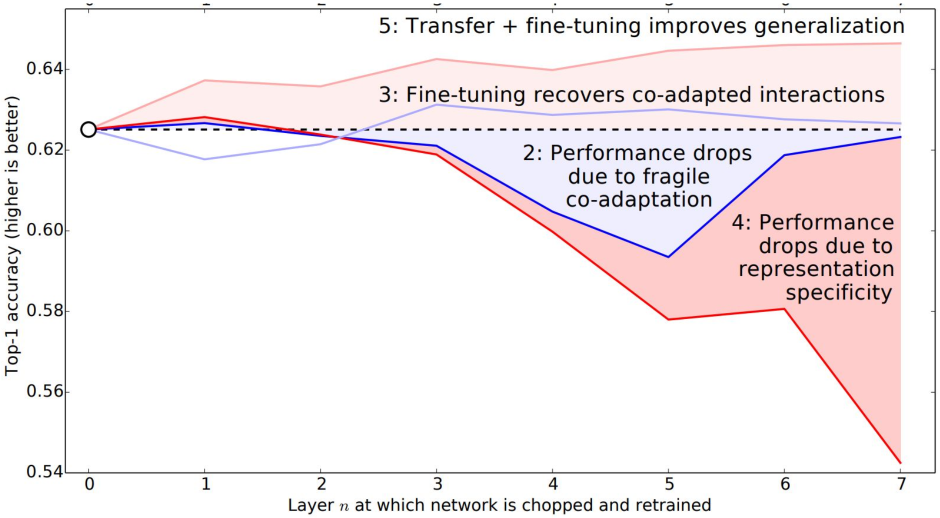


OBS! Everything may not be applicable with new initialization schemes, Resnet and batch-norm

[How transferable are features in deep neural networks?](#)

Transfer learning: Study

- Study done with plentiful data (split ImageNet in two)
- Locking weights deprecate performance
  - Remember lots of data
- More data improves performance, even if it's different classes.

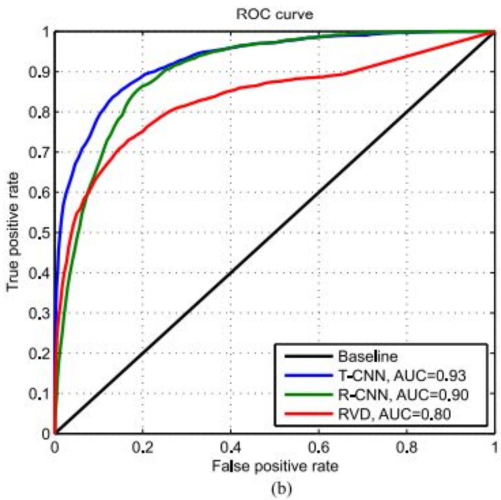


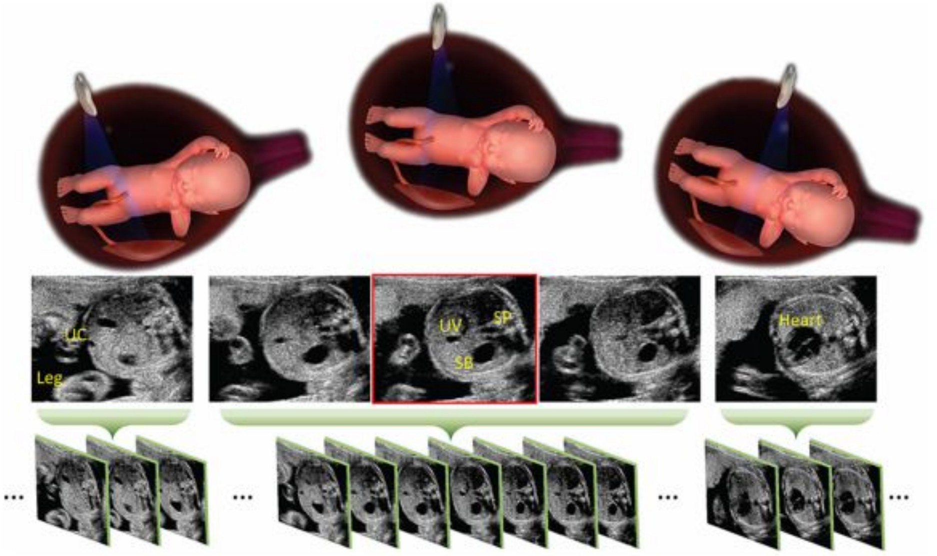
OBS! Everything may not be applicable with new initialization schemes and batch-norm

[How transferable are features in deep neural networks?]

What can you transfer to?

- Detecting special views in Ultrasound
- Initially far from ImageNet
- Benefit from fine-tuning imagenet features
- 300 patients, 11000 images





[Standard Plane Localization in Fetal Ultrasound via Domain Transferred Deep Neural Networks]

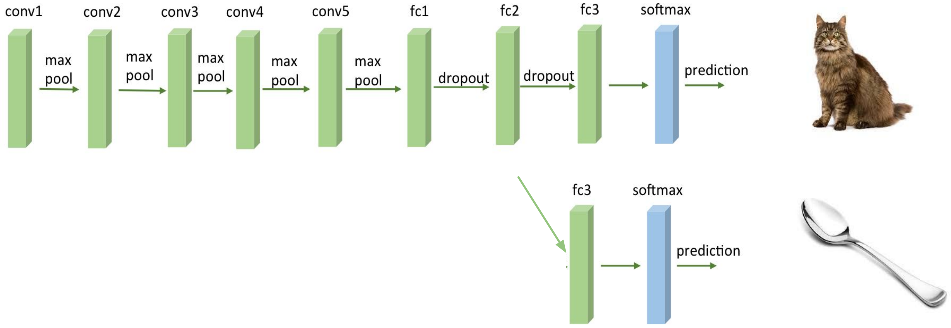
Transfer learning from pretrained network

With less parameters to train, you are less likely to overfit.

Features is often invariant to many different effects.

Need a lot less time to train.

OBS! Since networks trained on ImageNet have a lot of layers, it is still possible to overfit.

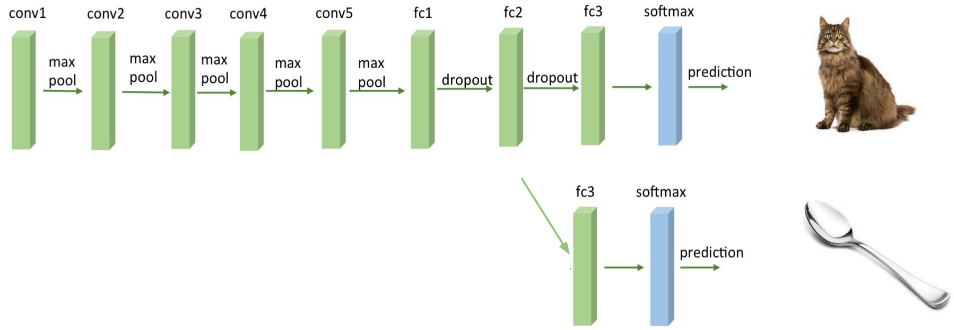


Transfer learning from pretrained network

Generally:

Very little data: train only last layer

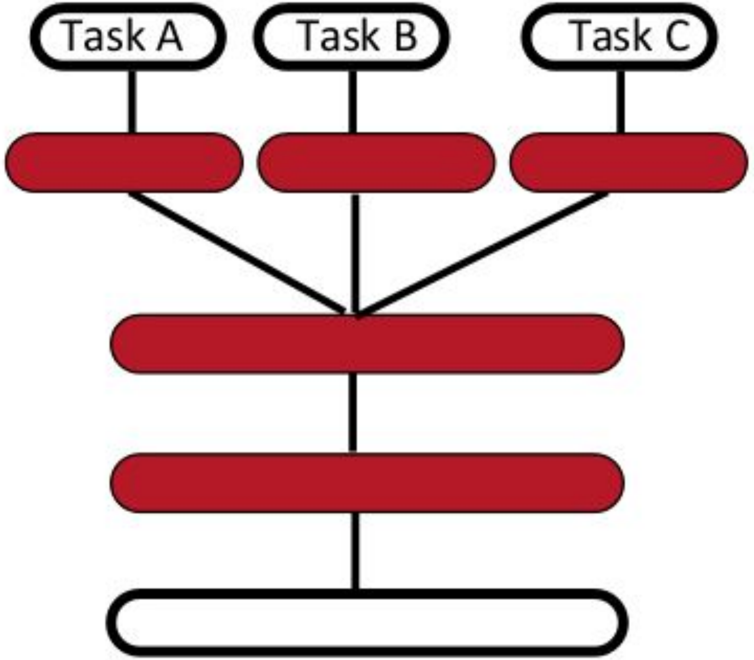
Some data: train the last layer**s**, finetune (small learning rate) the other layers



Multitask learning

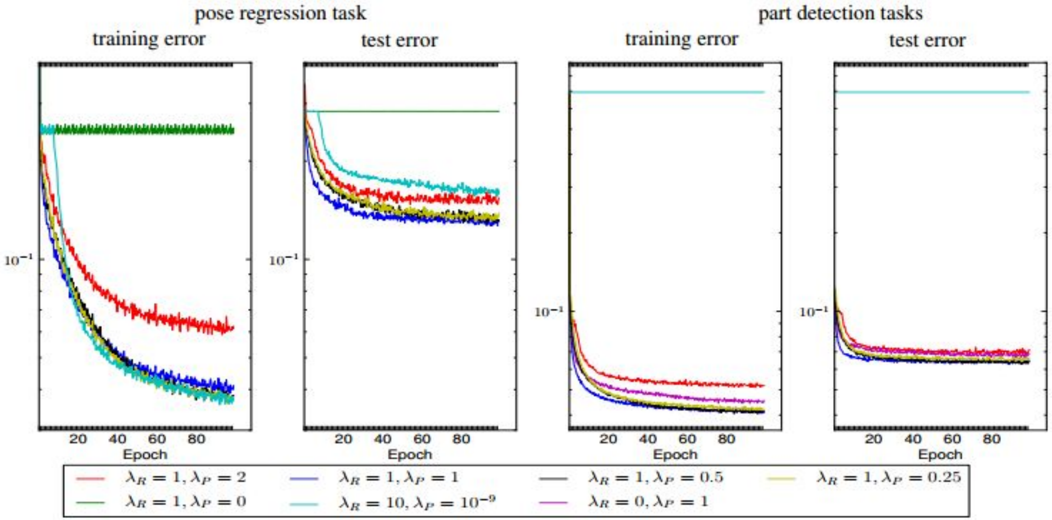
- Many small datasets
- Different targets
- Share base-representation

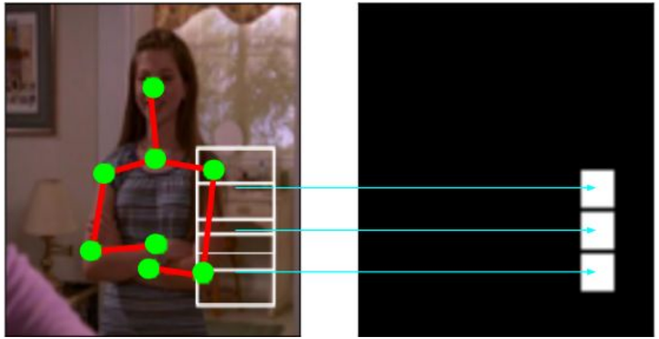
Same data with different labels can also have a regularizing effect.



Multitask learning: pose and body part

- Without multitask learning *regression task* is not learning
- With only a small input (10^{-9}) from the other task they train well
- With equal weight between tasks the test error is best for both tasks

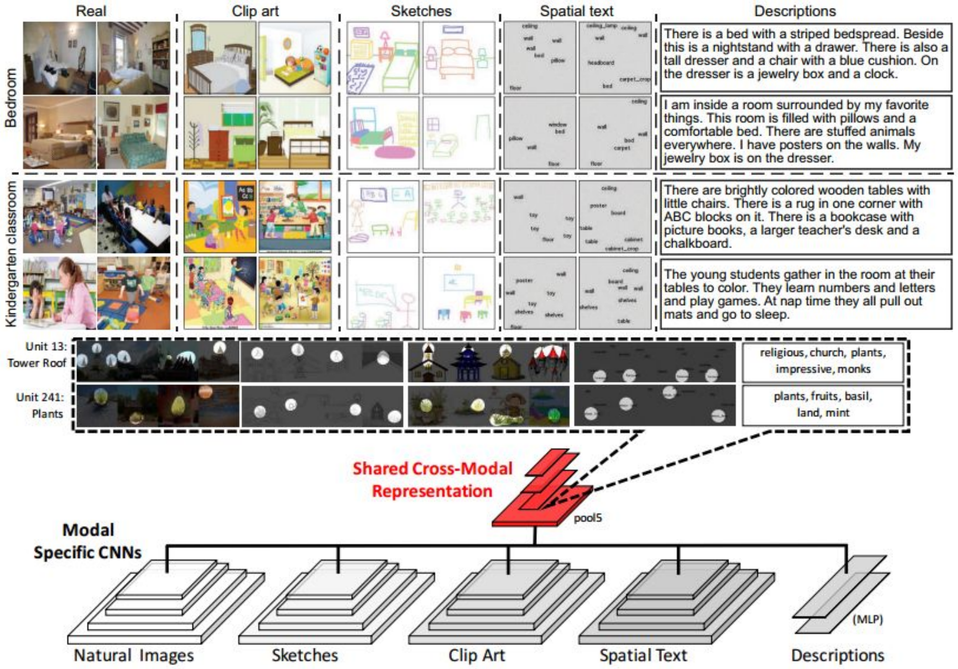




[Heterogeneous Multi-task Learning for Human Pose Estimation with Deep Convolutional Neural Network]

Same task different domain

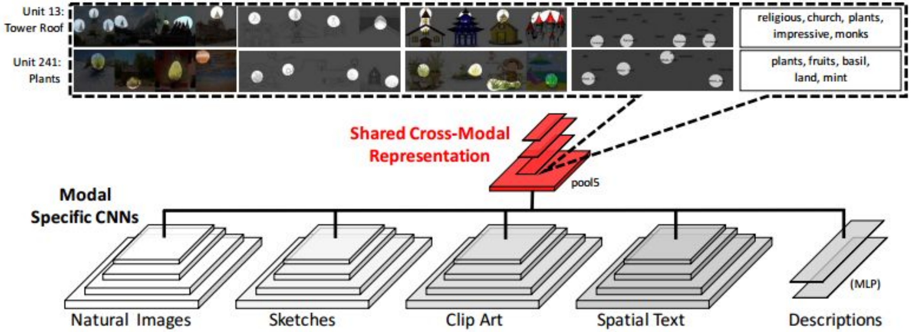
- Different domains with similar tasks
- Both text and different images
- Some categories not available for all modalities
- Learn jointly by sharing mid-level representation
- Training first part of the network from scratch



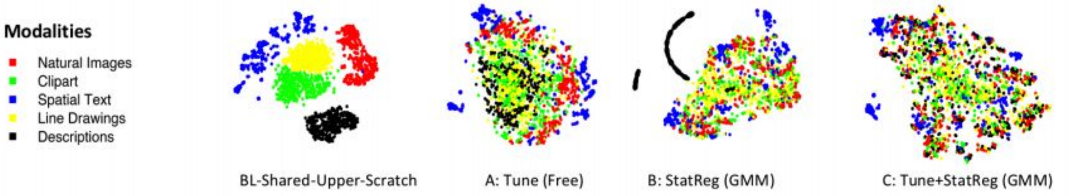
[Cross-Modal Scene Networks]

Same task different domain

- The network display better semantic alignment
- The network differentiate between classes and not modalities
- For B and C they also use regularization to force similar statistics in upper part of base-network



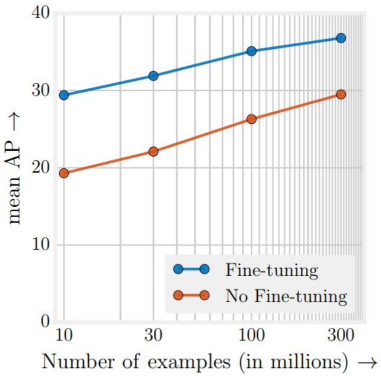
Cross Modal Retrieval (Query / Target)	NAT				CLP				SPT				LDR				DSC				Mean
	CLP	SPT	LDR	DSC	NAT	SPT	LDR	DSC	NAT	CLP	LDR	DSC	NAT	CLP	SPT	DSC	NAT	CLP	SPT	LDR	mAP
BL-Individual	17.9	11.9	10.0	1.3	12.2	10.3	9.2	1.3	7.0	**9.1**	**5.2**	1.1	5.7	8.8	5.4	1.2	0.9	1.4	1.5	1.2	6.1
BL-Shared-Upper-Scratch	7.0	7.8	4.1	10.9	5.5	5.0	3.2	9.2	5.2	4.5	2.7	8.9	3.1	3.0	3.0	5.2	5.8	5.1	6.3	3.2	5.4
BL-Shared-Upper	10.4	12.4	4.5	14.6	9.1	7.2	3.7	10.1	6.8	5.5	3.0	8.9	3.3	3.8	3.6	4.6	4.3	4.8	6.6	3.3	6.5
A: Tune	13.3	11.3	6.7	21.9	10.1	8.5	5.7	15.8	6.3	4.8	3.4	11.4	5.4	5.2	4.5	**9.5**	8.9	5.5	9.0	3.6	8.5
A: Tune (Free)	14.0	16.0	7.9	20.6	9.6	8.1	4.7	14.8	**11.3**	8.0	**5.2**	18.0	5.2	4.6	4.5	8.7	7.7	4.2	9.4	3.4	9.3
B: StatReg (Gaussian)	17.3	11.9	10.1	1.6	12.6	8.9	9.7	1.3	6.6	8.6	4.9	1.4	5.4	8.0	5.3	1.2	1.2	1.8	1.8	1.6	6.1
B: StatReg (GMM)	**18.2**	11.3	**10.5**	1.2	**14.5**	**10.7**	**10.1**	1.2	7.0	7.9	4.9	1.2	**7.9**	**9.9**	**6.5**	1.0	0.8	1.0	1.2	1.0	6.4
C: Tune + StatReg (GMM)	13.2	**16.9**	7.2	**24.5**	10.9	10.4	5.7	**16.5**	10.1	8.3	5.0	**18.8**	5.7	5.7	6.0	8.8	**19.5**	**15.8**	**21.4**	**8.0**	**11.9**

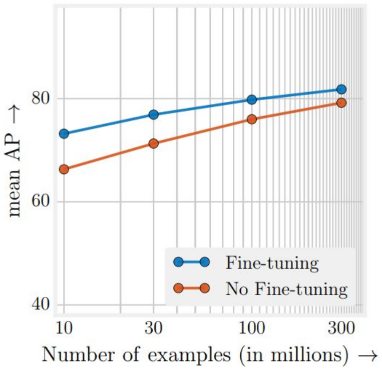


[Cross-Modal Scene Networks]

When do we have enough?

When do we have enough? Never?



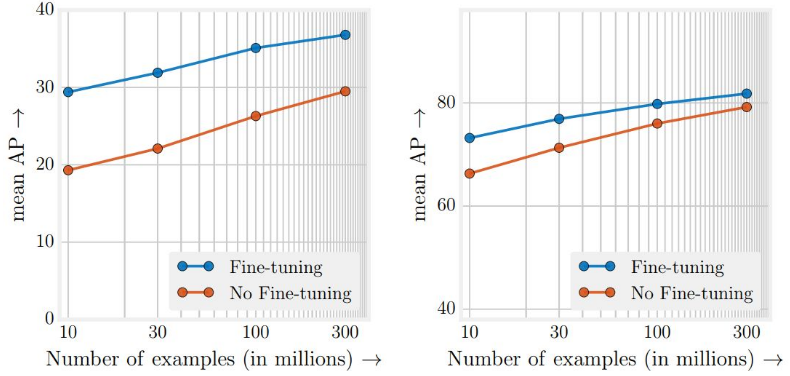


Method	mAP@0.5	mAP@[0.5,0.95]
He *et al.* [16]	53.3	32.2
ImageNet	53.6	34.3
300M	56.9	36.7
ImageNet+300M	**58.0**	**37.4**
Inception ResNet [38]	56.3	35.5

When do we have enough? Never?

When things work good enough.

Algorithm improvement can be more effective.



Detection Leaderboard

BBOX: Dev Standard15 Chal15 Chal16 Chal17

SEGM: Dev Standard15 Chal15 Chal16 Chal17 Chal18

Copy to Clipboard | Export to CSV

Search:

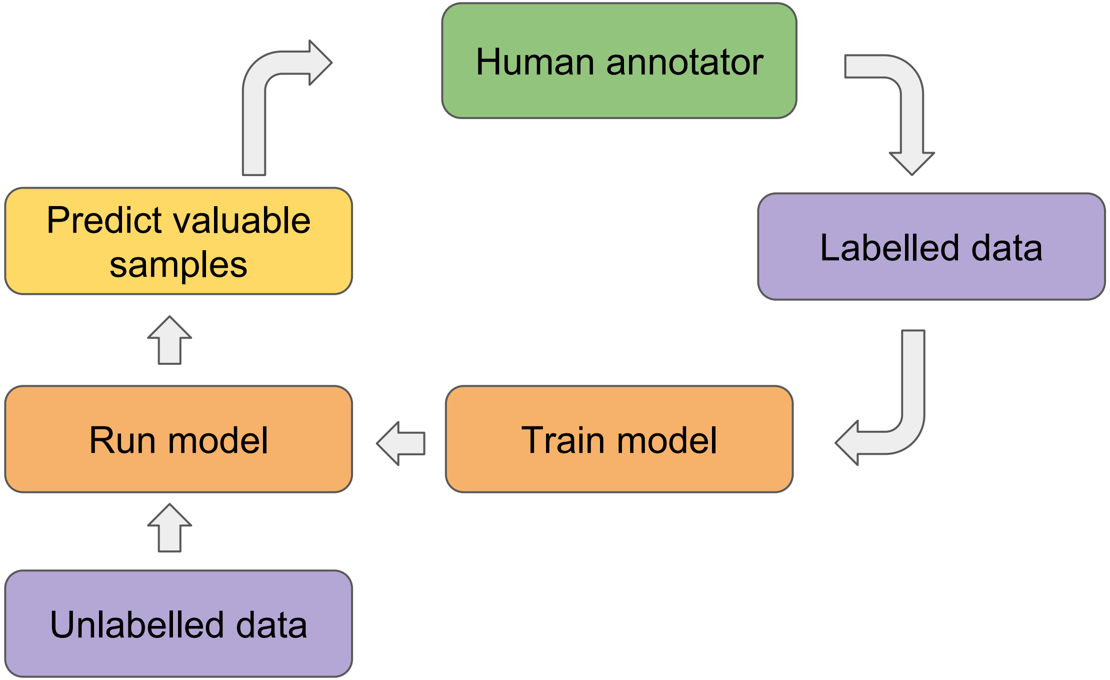
	AP	AP50	AP75	APS	APM	APL	AR1	AR10	AR100	ARS	ARM	ARL	date
Megvii (Face++)	0.526	0.730	0.585	0.343	0.556	0.660	0.391	0.645	0.689	0.513	0.727	0.827	2017-10-05
UCenter	0.510	0.705	0.558	0.326	0.539	0.648	0.392	0.640	0.678	0.497	0.720	0.829	2017-10-05
MSRA	0.507	0.717	0.566	0.343	0.529	0.627	0.379	0.638	0.690	0.524	0.720	0.824	2017-10-05
FAIR Mask R-CNN	0.503	0.720	0.558	0.328	0.537	0.627	0.380	0.622	0.659	0.485	0.704	0.800	2017-10-05
Trimps-Soushen+QINIU	0.482	0.681	0.534	0.310	0.512	0.610	0.373	0.611	0.652	0.466	0.688	0.801	2017-10-05
bharat_umd	0.482	0.694	0.536	0.312	0.514	0.606	0.365	0.605	0.647	0.456	0.696	0.793	2017-10-05

Method	mAP@0.5	mAP@[0.5,0.95]
He *et al.* [16]	53.3	32.2
ImageNet	53.6	34.3
300M	56.9	36.7
ImageNet+300M	**58.0**	**37.4**
Inception ResNet [38]	56.3	35.5

Active learning

Active learning

- Typical active learning scheme
- Not representative…
    - decades of research



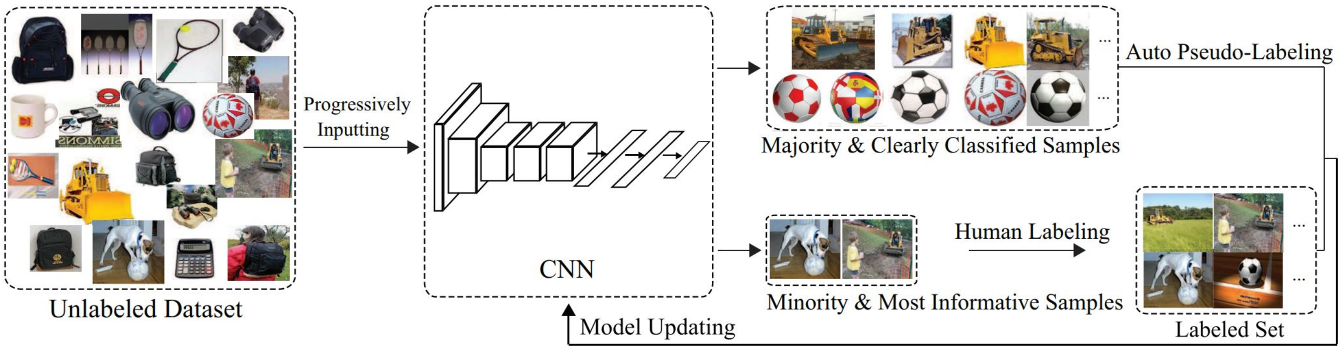
Active learning

Often rely on measures:

- Confidence
- Sample importance

Typically:

- Entropy
- Softmax confidence
- Variance
- Margin



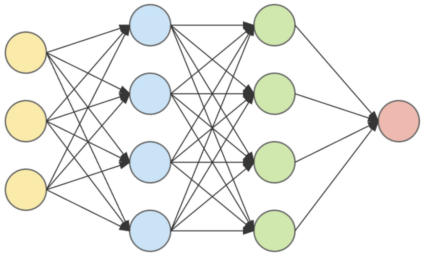
[Cost-Effective Active Learning for Deep Image Classification]()

Measuring uncertainty

- Dropout
- Ensembles
- Stochastic weights

- Far from cluster center ([Suggestive Annotation: A Deep Active Learning Framework for Biomedical Image Segmentation]())

[The power of ensembles for active learning in image classification]()



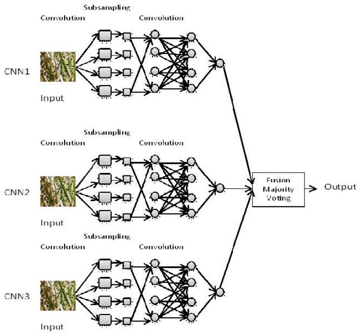
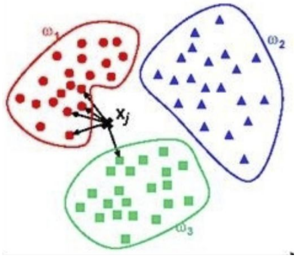


Fig. 1: Automatic Feature Extraction based Ensemble of CNNs



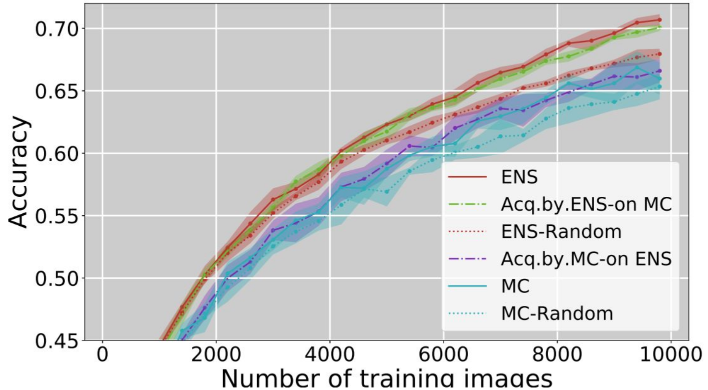
Measuring uncertainty

- Ensembles seem to work best for now
- Relative small effect on large important datasets like ImageNet
- More research needed

My opinion:

- Relevant for institutions that work with different and large quantities of data
- Need a large problem to justify effort

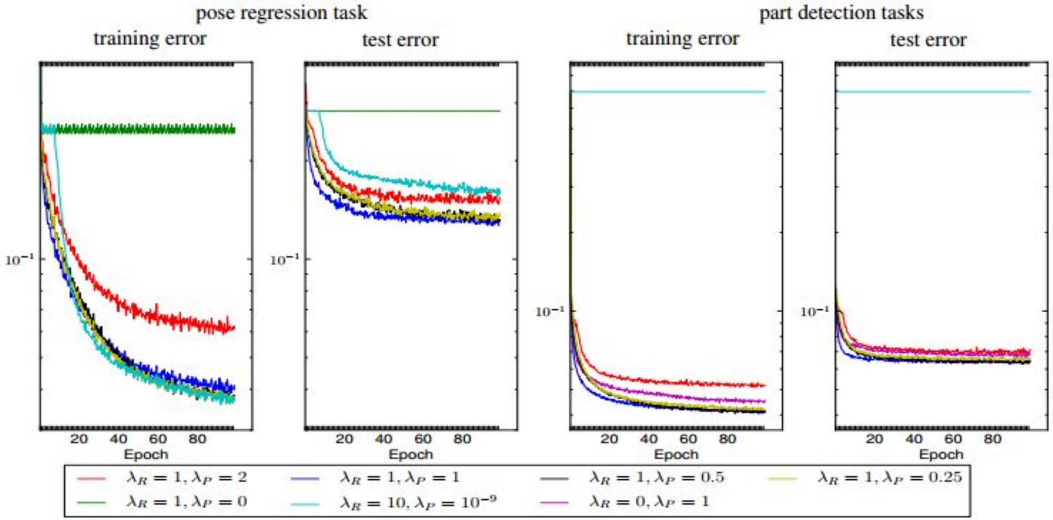
[The power of ensembles for active learning in image classification]()



	40k	80k	120k	160k	200k	240k	280k
Random	0.159 (0.004)	0.257 (0.003)	0.321 (0.006)	0.372 (0.003)	0.407 (0.007)	0.439 (0.001)	0.470
VarR	0.152 (0.003)	0.257 (0.004)	0.324 (0.002)	0.383 (0.002)	0.427 (0.004)	0.458 (0.004)	0.494

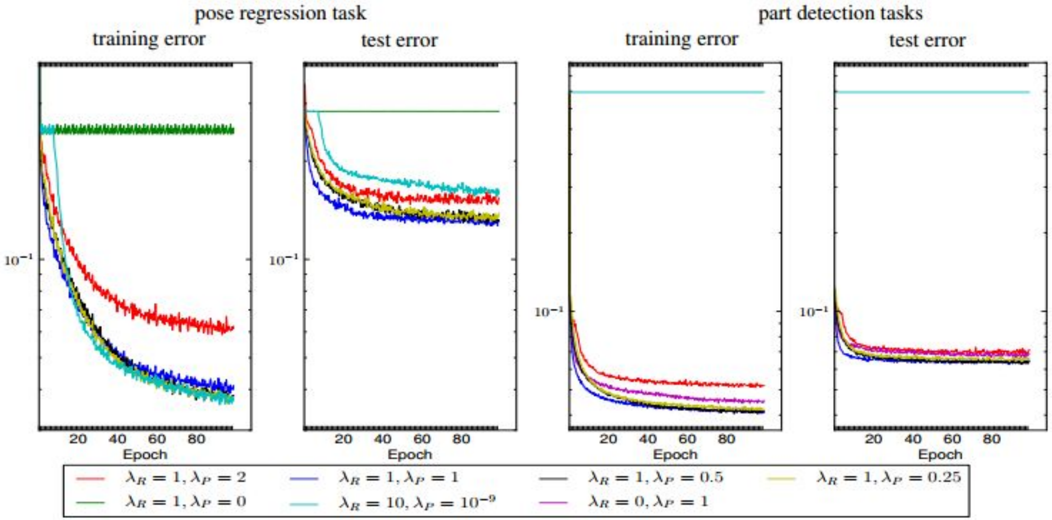
When you are not learning

Network is learning nothing



Network is learning nothing

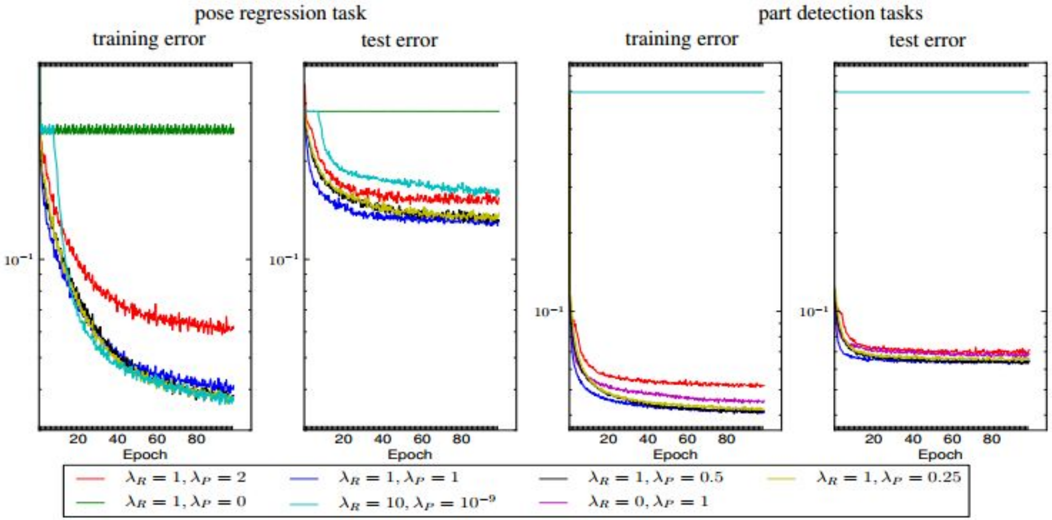
You probably screwed up!



Network is learning nothing

You probably screwed up!

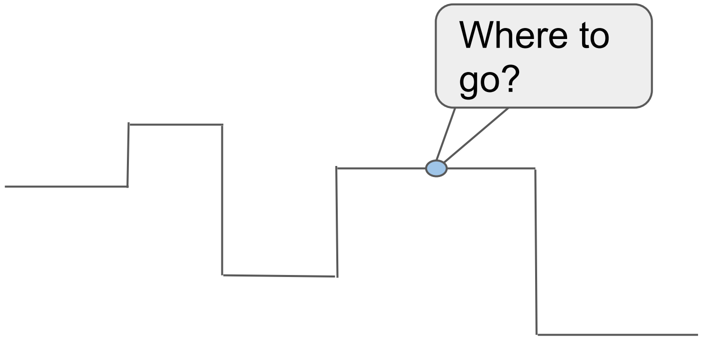
- Data and labels not aligned
- Not updating batch norm parameters
- Wrong learning rate
- etc.



Target is not learnable

Why do we use **softmax**, when performance is often measured in **accuracy** (% of correct)?

- A small change in weights does not change loss function
- Might be an obvious example...

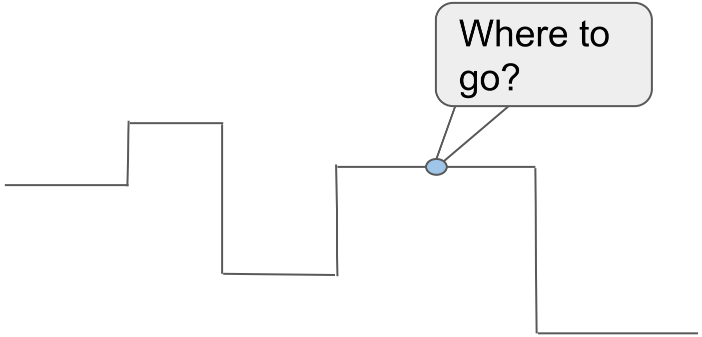


Target is not learnable

Why do we use **softmax**, when performance is often measured in **accuracy** (% of correct)?

- A small change in weights does not change loss function
- Might be an obvious example…

Softmax can "always" improve



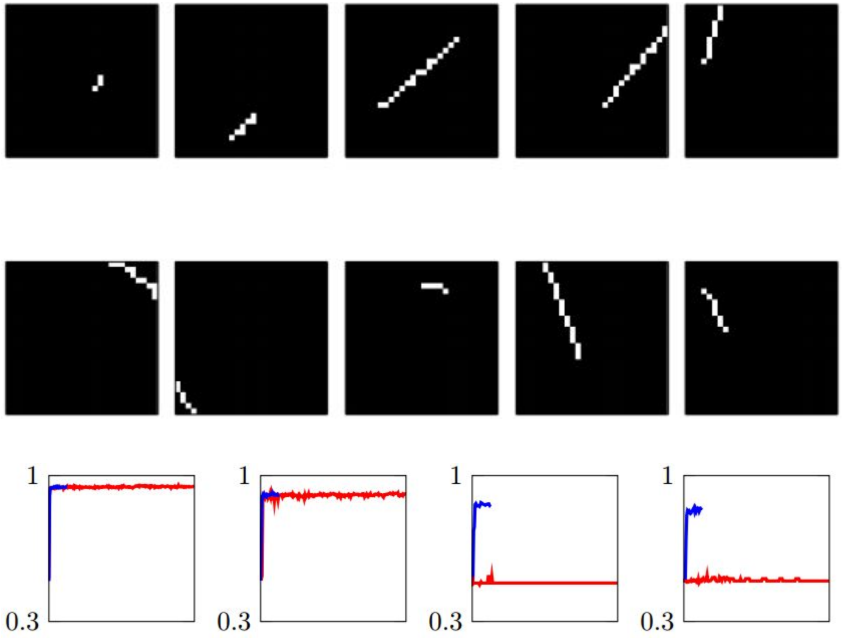
Target is not learnable

Answer the question: do all slopes have the same **sign**.

To train on the correct solution directly is not working if you have more than 2 images.

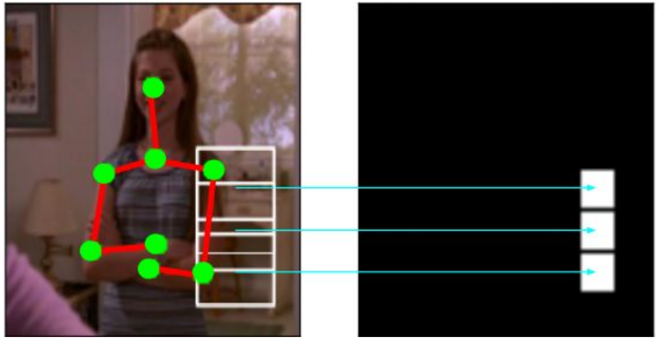
If you train with two targets: Is slope positive and do all slopes have the same sign, works.

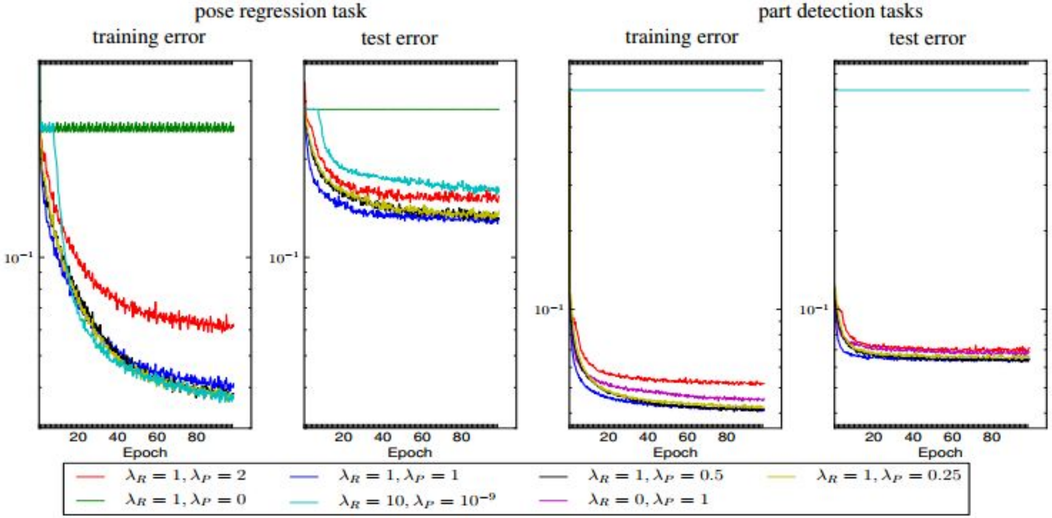
The loss is not very smooth, as a small change in slope on one image totally change the target.



Target is not learnable

- Without multitask learning *regression task* is not learning
- With only a small input (10^{-9}) from the other task they train well
- With equal weight between tasks the test error is best for both tasks





[Heterogeneous Multi-task Learning for Human Pose Estimation with Deep Convolutional Neural Network]

Surrogat losses

Auxiliary task

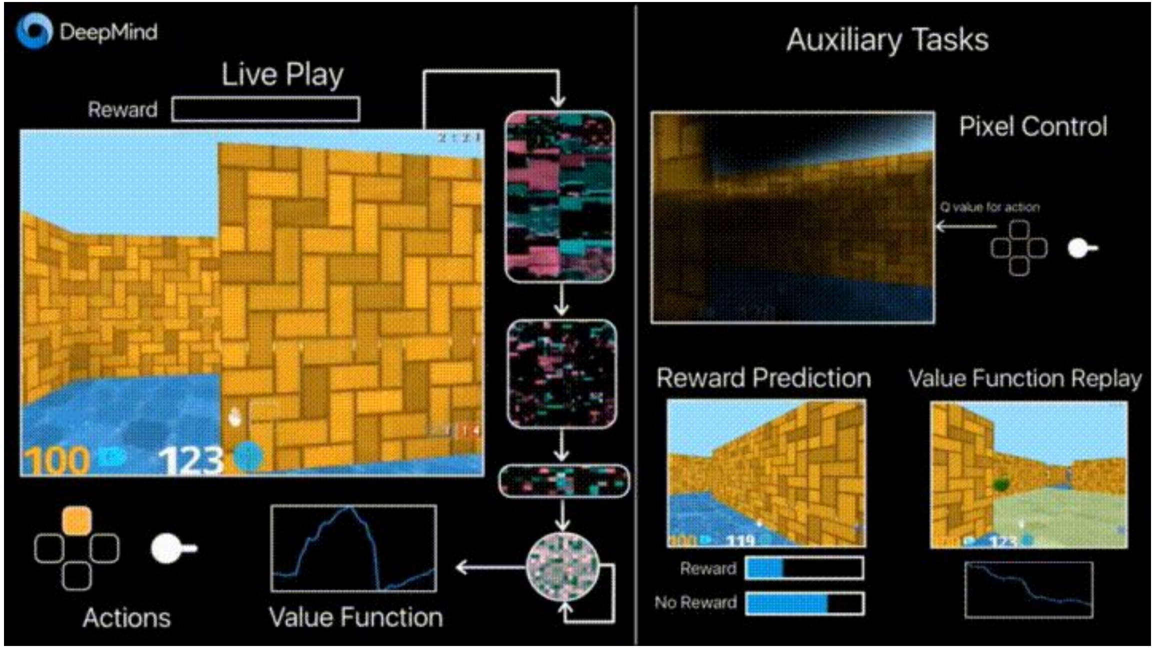
Pixel control:

- Find actions to maximize pixel changes

Reward prediction:

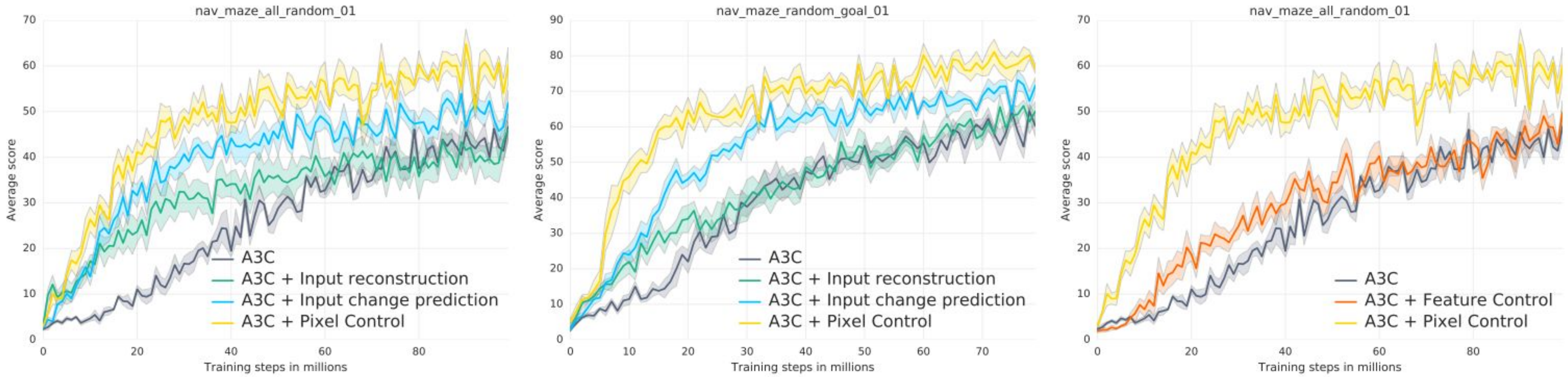
- Sample history and predict reward in the next frame
- Evenly sampled: reward, neutral and punishment

Still used in newer research



[Reinforcement Learning with Unsupervised Auxiliary Tasks]

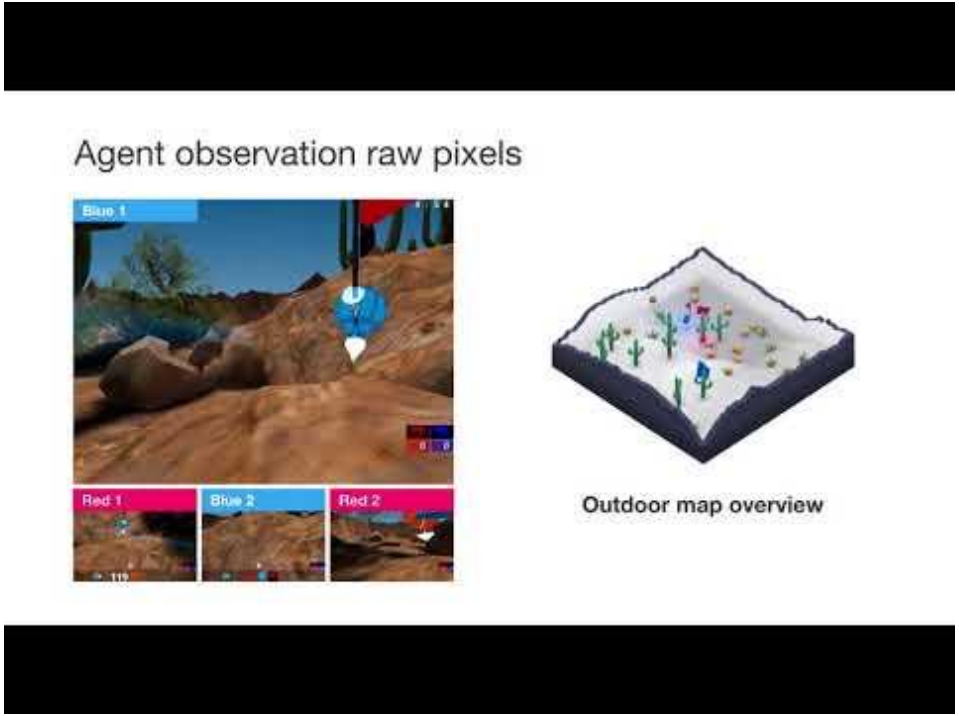
Auxiliary task



[Reinforcement Learning with Unsupervised Auxiliary Tasks]

Auxiliary task - learned

- Using both previous auxiliary targets
- Learning an additional target function by evolution

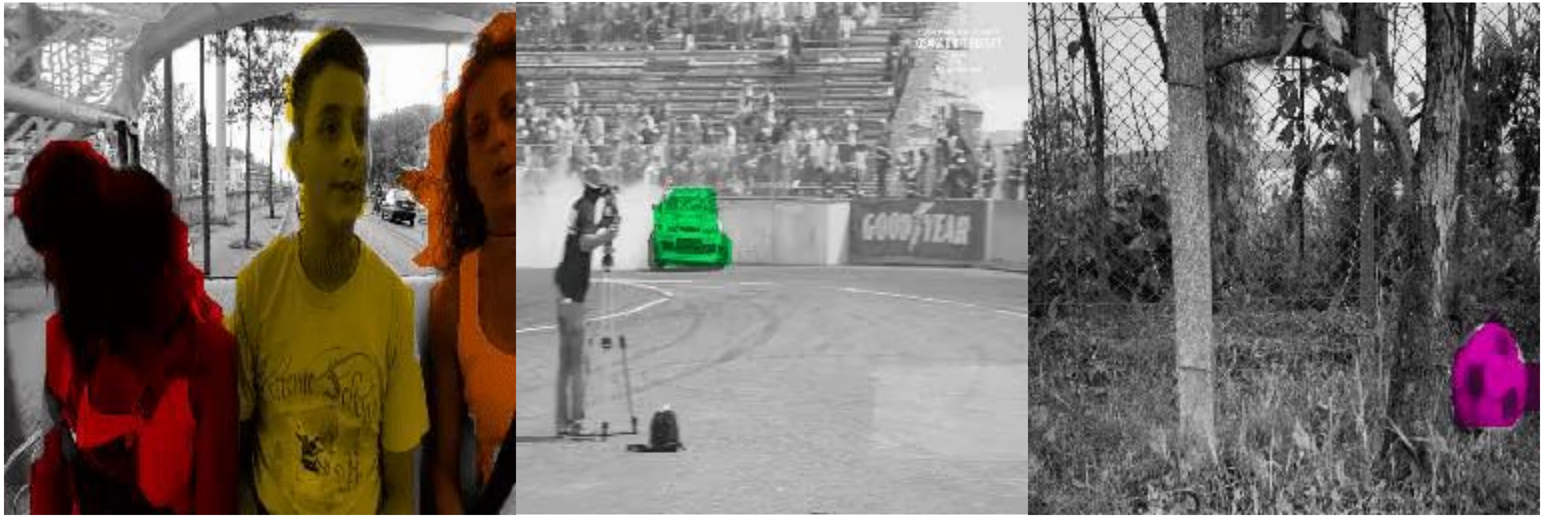


[Human-level performance in first-person multiplayer games with population-based deep reinforcement learning]

Auxiliary task - learned

- Using both previous auxiliary targets
- Learning an additional target function by evolution

Tracking by colorization

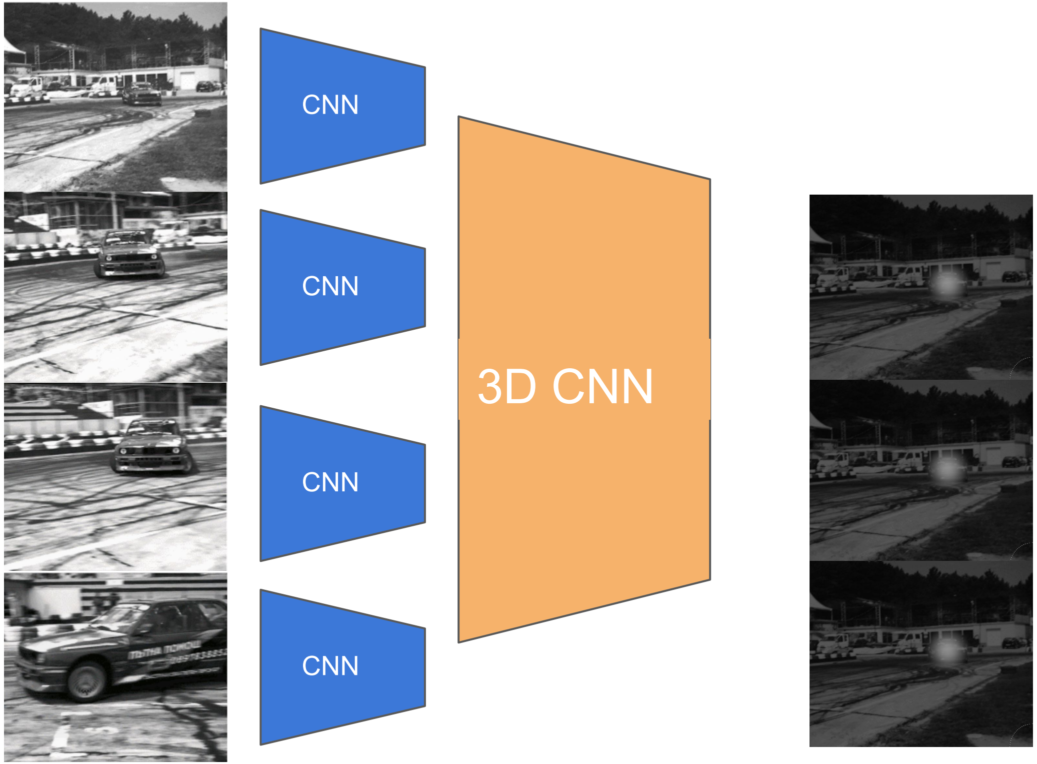


https://ai.googleblog.com/2018/06/self-supervised-tracking-via-video.html

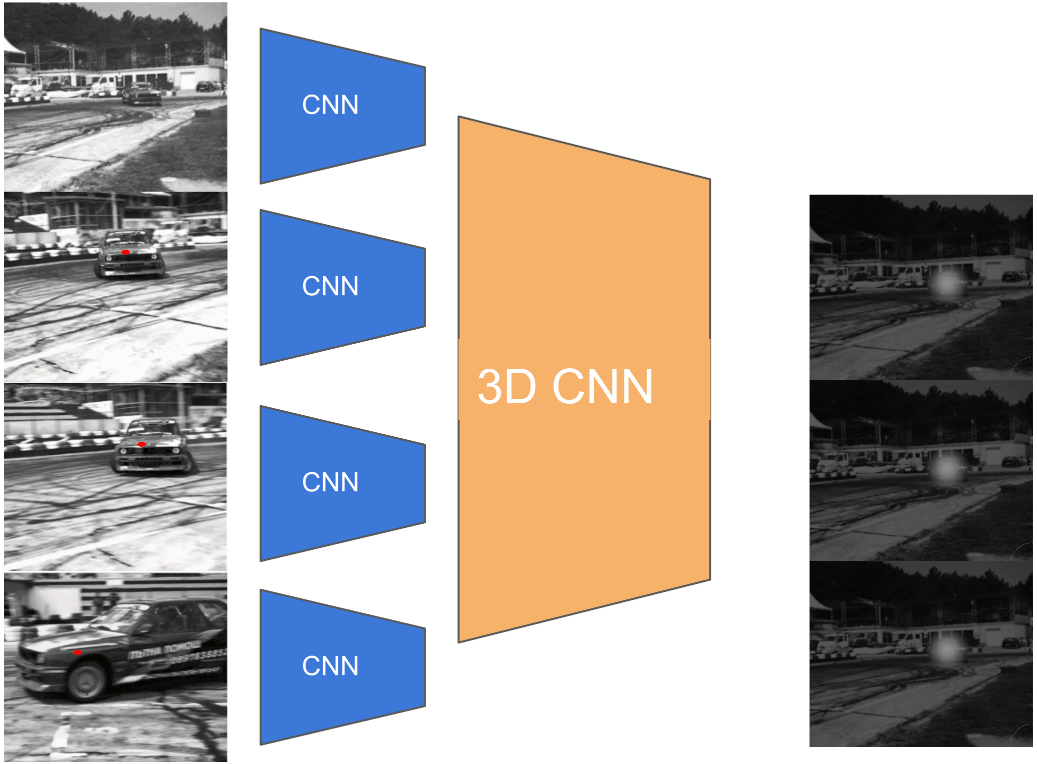
Tracking Emerges by Colorizing Videos

Tracking by colorization

Tracking by colorization



Tracking by colorization



Where to get color from?

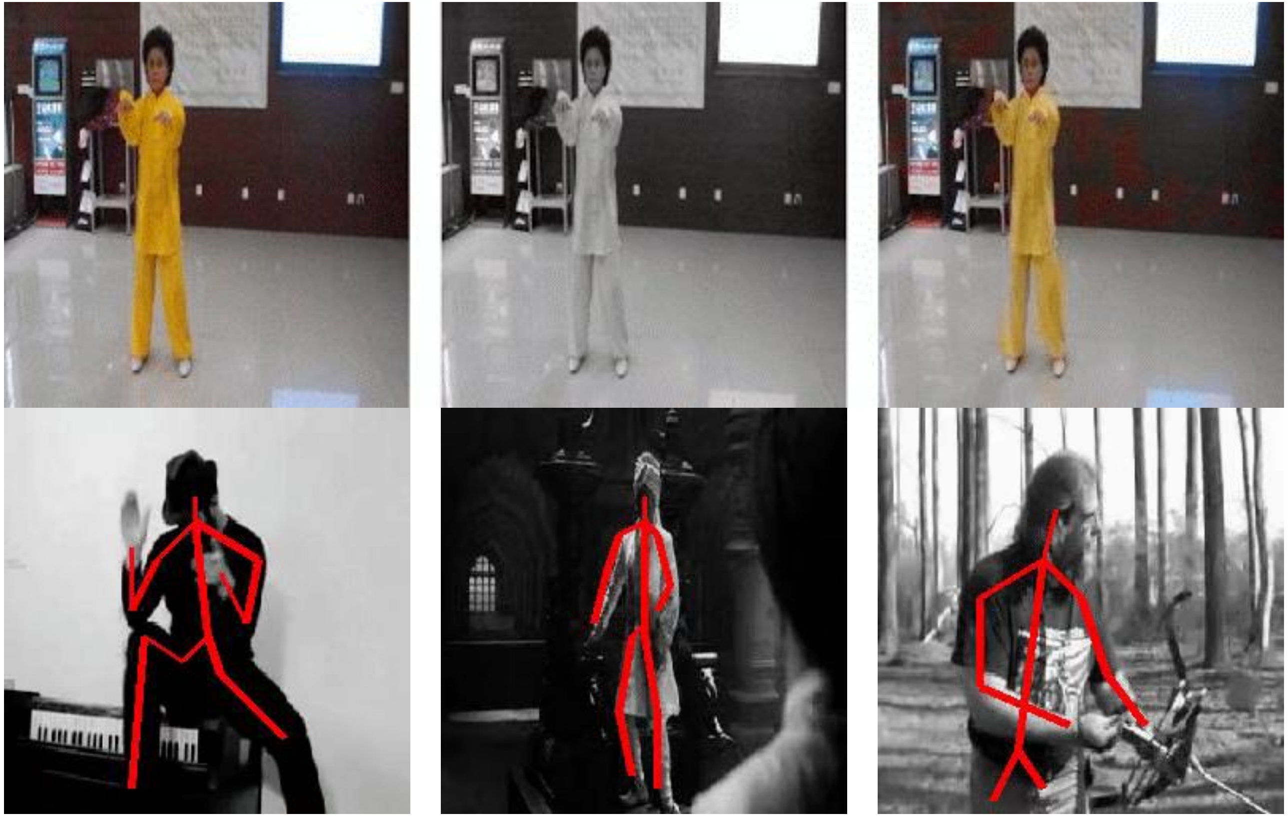
- Weighted average of colors
- For every pixel

Tracking by colorization - Loss

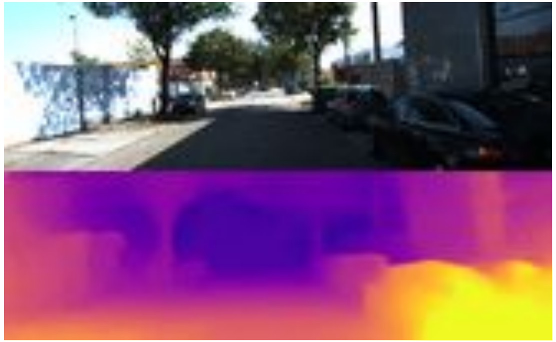
- Simplify/quantize color
- Use softmax cross entropy loss
- Colors are now simple categories

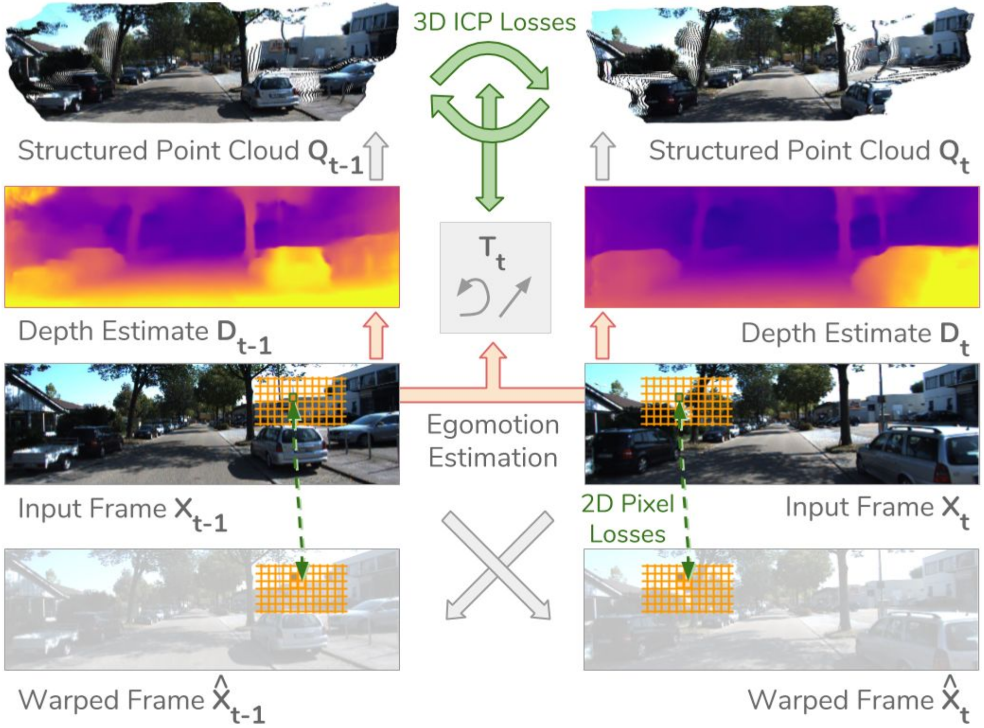
- Why not just just use mean squared loss?

Tracking by colorization - Fun!



Vid2depth - 3D Geometric Constraints



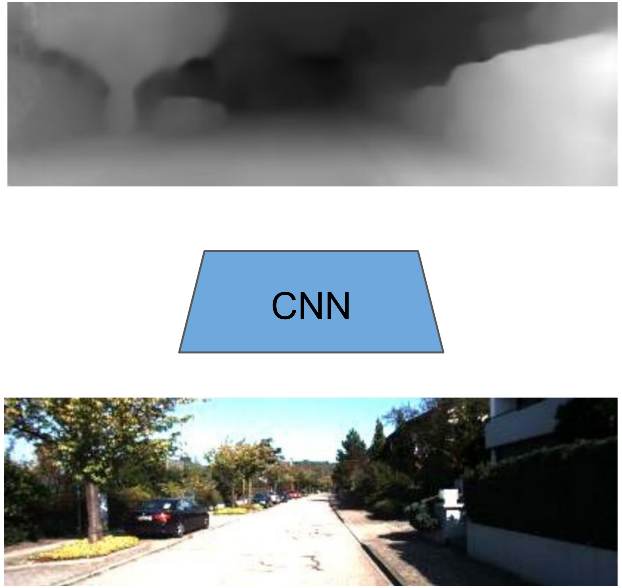


[Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints]

Vid2depth - 3D Geometric Constraints

- You want a 3D map of the world
- First try to estimate depth

D

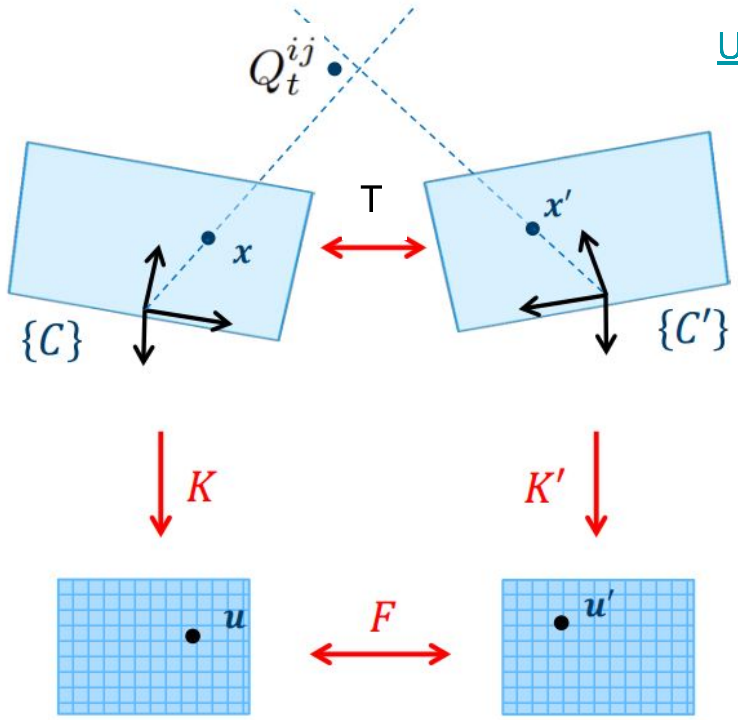


UNIK4690

Vid2depth - 3D Geometric Constraints

$$Q_t^{ij} = D_t^{ij} \cdot K^{-1}[i, j, 1]^T$$

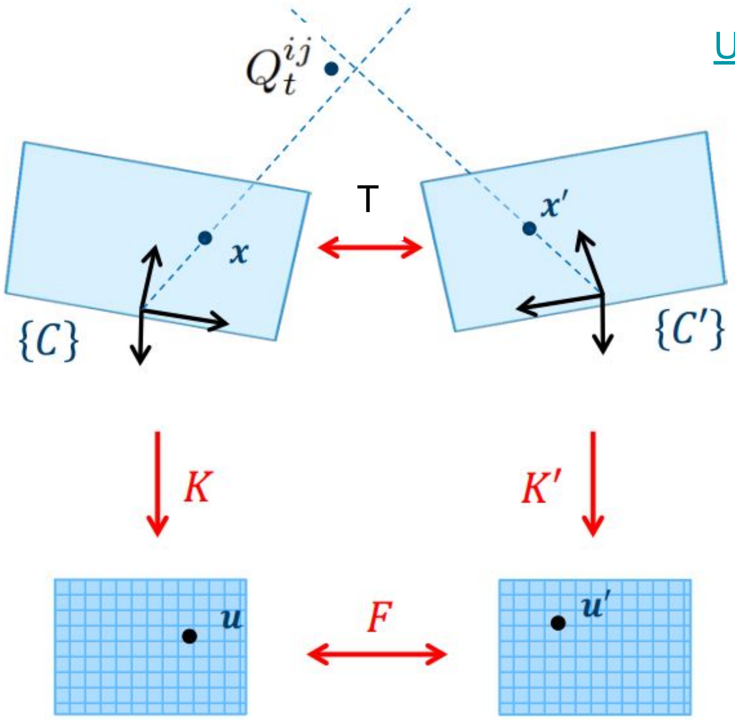
UNIK4690



Vid2depth - 3D Geometric Constraints

$$Q_t^{ij} = D_t^{ij} \cdot K^{-1}[i, j, 1]^T$$

$$[\hat{i}, \hat{j}, 1]^T = KT_t\big(D_t^{ij} \cdot K^{-1}[i, j, 1]^T\big)$$



UNIK4690

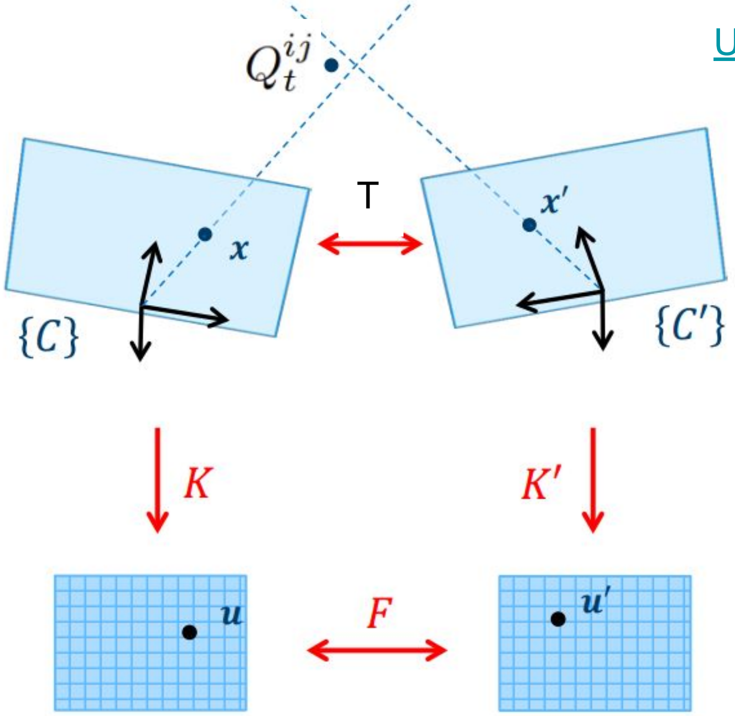
Vid2depth - 3D Geometric Constraints

UNIK4690

$$Q_t^{ij} = D_t^{ij} \cdot K^{-1}[i, j, 1]^T$$

$$[\hat{i}, \hat{j}, 1]^T = KT_t\left(D_t^{ij} \cdot K^{-1}[i, j, 1]^T\right)$$

$$\hat{X}_t^{ij} = X_{t-1}^{\hat{i}\hat{j}}$$



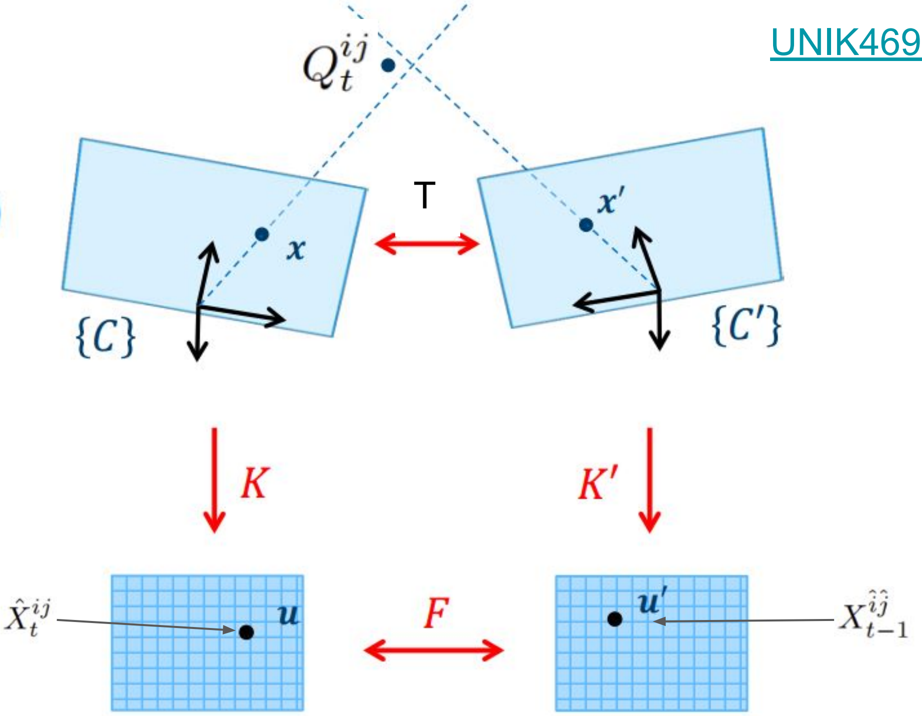
Vid2depth - 3D Geometric Constraints

UNIK4690

$$Q_t^{ij} = D_t^{ij} \cdot K^{-1}[i, j, 1]^T$$

$$[\hat{i}, \hat{j}, 1]^T = KT_t\big(D_t^{ij} \cdot K^{-1}[i, j, 1]^T\big)$$

$$\hat{X}_t^{ij} = X_{t-1}^{\hat{i}\hat{j}}$$



Vid2depth - Image Reconstruction Loss

$$Q_t^{ij} = D_t^{ij} \cdot K^{-1}[i, j, 1]^T$$

$$[\hat{i}, \hat{j}, 1]^T = KT_t(D_t^{ij} \cdot K^{-1}[i, j, 1]^T)$$

$$\hat{X}_t^{ij} = X_{t-1}^{\hat{i}\hat{j}}$$

$$L_{\text{rec}} = \sum_{ij} \|(X_t^{ij} - \hat{X}_t^{ij}) \|$$

Vid2depth - Image Reconstruction Loss

$$Q_t^{ij} = D_t^{ij} \cdot K^{-1}[i, j, 1]^T$$

$$[\hat{i}, \hat{j}, 1]^T = KT_t(D_t^{ij} \cdot K^{-1}[i, j, 1]^T)$$

$$\hat{X}_t^{ij} = X_{t-1}^{\hat{i}\hat{j}}$$

$$L_{\text{rec}} = \sum_{ij} \|(X_t^{ij} - \hat{X}_t^{ij}) \quad \|$$

?!?

Vid2depth - Principled Mask

$$Q_t^{ij} = D_t^{ij} \cdot K^{-1}[i, j, 1]^T$$

$$[\hat{i}, \hat{j}, 1]^T = K T_t (D_t^{ij} \cdot K^{-1}[i, j, 1]^T)$$

$$\hat{X}_t^{ij} = X_{t-1}^{\hat{i}\hat{j}}$$

$$L_{\text{rec}} = \sum_{ij} \|(X_t^{ij} - \hat{X}_t^{ij}) M_t^{ij}\|$$

?!?

Vid2depth - Principled Mask

$$Q_t^{ij} = D_t^{ij} \cdot K^{-1}[i, j, 1]^T$$

$$[\hat{i}, \hat{j}, 1]^T = KT_t\big(D_t^{ij} \cdot K^{-1}[i, j, 1]^T\big)$$

$$\hat{X}_t^{ij} = X_{t-1}^{\hat{i}\hat{j}}$$

$$L_{\text{rec}} = \sum_{ij} \|(X_t^{ij} - \hat{X}_t^{ij})M_t^{ij}\|$$

Vid2depth - Principled Mask

$$Q_t^{ij} = D_t^{ij} \cdot K^{-1}[i, j, 1]^T$$

$$[\hat{i}, \hat{j}, 1]^T = KT_t\big(D_t^{ij} \cdot K^{-1}[i, j, 1]^T\big)$$

$$\hat{X}_t^{ij} = X_{t-1}^{\hat{i}\hat{j}}$$

$$L_{\text{rec}} = \sum_{ij} \|(X_t^{ij} - \hat{X}_t^{ij}) M_t^{ij}\|$$

OBS! Missing depth test

Vid2depth - Image Reconstruction Loss

Not accounted for changes:

- Reflections
- Illumination
- etc.

- Noisy loss
- Artifacts
- Regularization cause blur

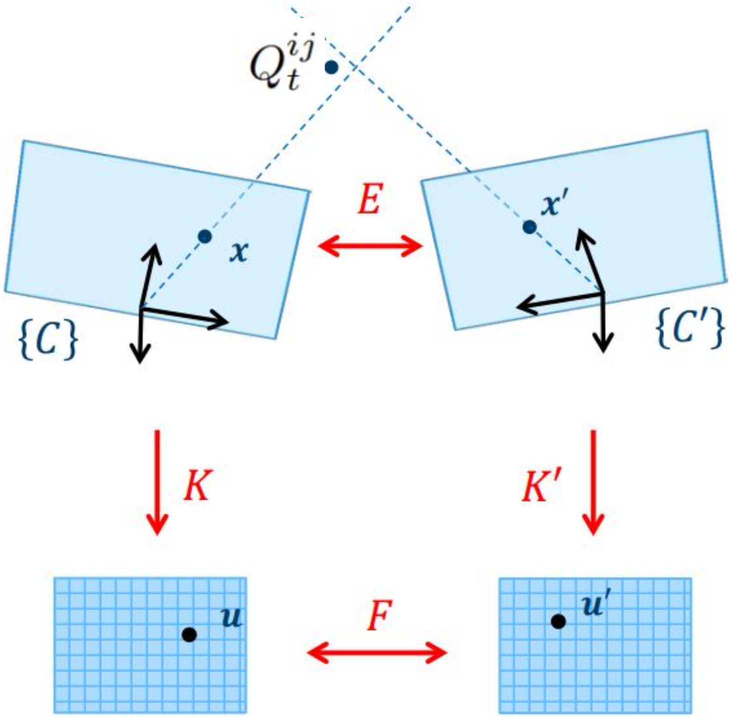
NVIDIA

$$L_{\text{rec}} = \sum_{ij} \|(X_t^{ij} - \hat{X}_t^{ij}) M_t^{ij}\|$$

Vid2depth - 3D Point Cloud Alignment Loss

Remember our point cloud Q

$$Q_t^{ij} = D_t^{ij} \cdot K^{-1}[i, j, 1]^T$$

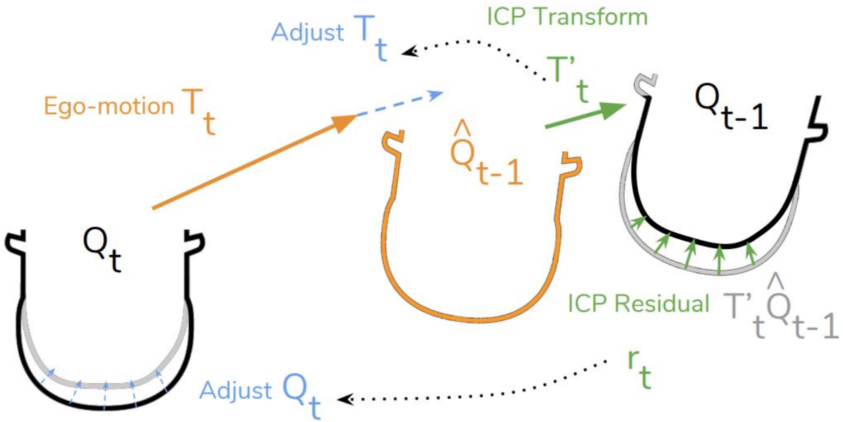


Vid2depth - 3D Point Cloud Alignment Loss

Remember our point cloud Q

1. Finding alignment between point clouds with Iterative Closest Point
   a. Align pairs of points (closest pairs of points)
   b. Find a transform that minimizes point-to-point distances
   c. Apply transform
   d. Realign pairs with transformed point cloud
   e. Outputs “best” transform **T** and residuals **r**

$$\arg\min_{T'} \frac{1}{2} \sum_{ij} \|T' \cdot A^{ij} - B^{c(ij)}\|^2$$

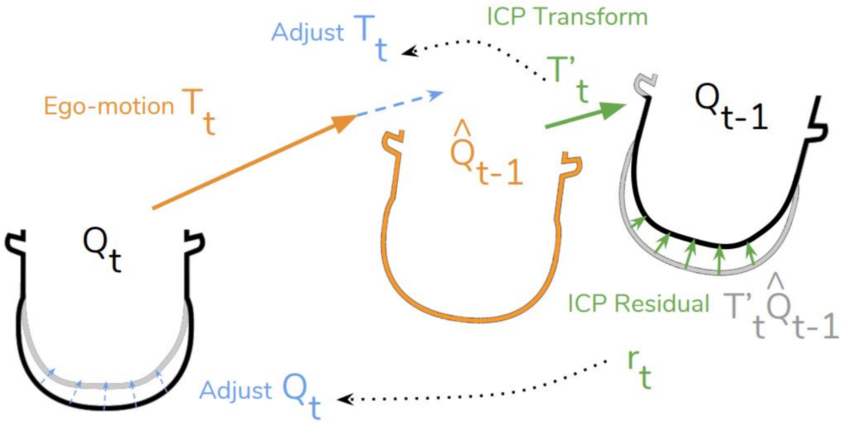


Vid2depth - 3D Point Cloud Alignment Loss

Remember our point cloud Q

1. Finding alignment between point clouds with Iterative Closest Point (ICP)
2. Perfect estimated ego-motion should give identity, transform from ICP

$$\|T'_t - I\|_1$$

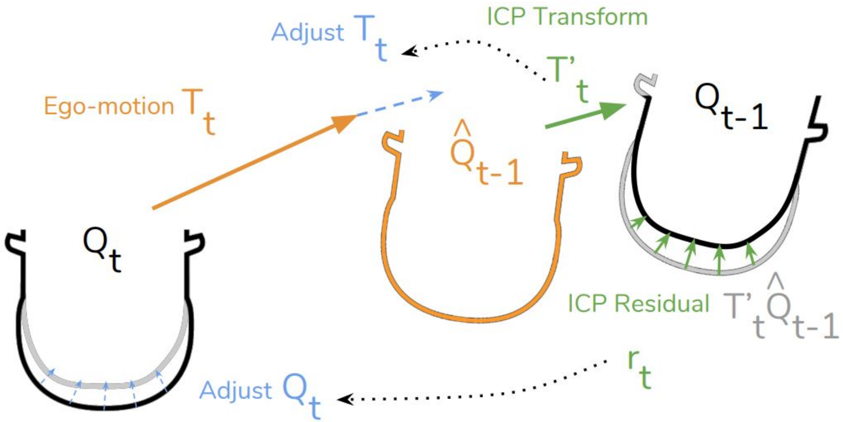


Vid2depth - 3D Point Cloud Alignment Loss

Remember our point cloud Q

1. Finding alignment between point clouds with Iterative Closest Point (ICP)
2. Perfect estimated ego-motion should give identity, transform from ICP
3. Perfect estimated depth image should give zero residuals from ICP

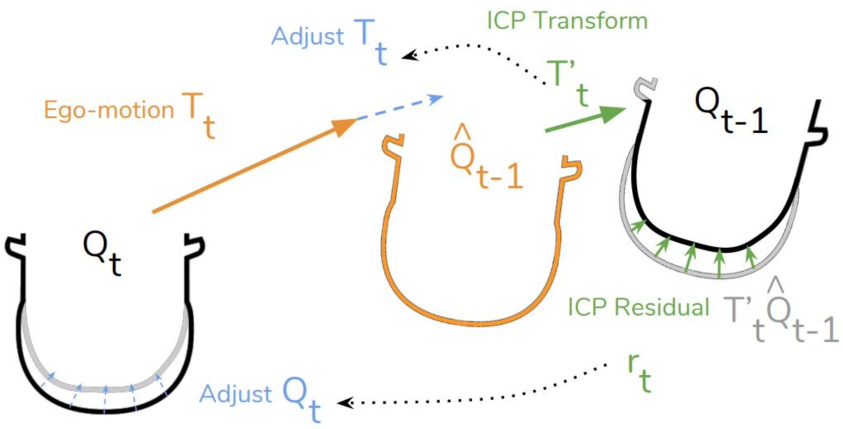
$$\|r_t\|_1$$



Vid2depth - 3D Point Cloud Alignment Loss

Remember our point cloud Q

1. Finding alignment between point clouds with Iterative Closest Point (ICP)
2. Perfect estimated ego-motion should give identity, transform from ICP
3. Perfect estimated depth image should give zero residuals from ICP



$$L_{3D} = ||T'_t - I||_1 + ||r_t||_1,$$

Vid2depth- Structured Similarity

- Quality of image predictions
- Calculated for local patches
- Difference between image and reconstructed image

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x + \sigma_y + c_2)}$$

$$L_{\mathrm{SSIM}} = \sum_{ij} \left[1 - \mathrm{SSIM}(\hat{X}_t^{ij}, X_t^{ij})\right] M_t^{ij}$$

Vid2depth- Depth smoothness loss

- Edges of depth image should correspond to edges in input image
- Often correct, but not always

$$L_{\text{sm}} = \sum_{i,j} \|\partial_x D^{ij}\| e^{-\|\partial_x X^{ij}\|} + \|\partial_y D^{ij}\| e^{-\|\partial_y X^{ij}\|}$$

Vid2depth - results depth

Method	Supervision	Dataset	Cap	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Train set mean	-	K	80m	0.361	4.826	8.102	0.377	0.638	0.804	0.894
Eigen *et al.* [6] Coarse	Depth	K	80m	0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen *et al.* [6] Fine	Depth	K	80m	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu *et al.* [18]	Depth	K	80m	0.201	1.584	6.471	0.273	0.68	0.898	0.967
Zhou *et al.* [32]	-	K	80m	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Zhou *et al.* [32]	-	CS + K	80m	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Ours	-	K	80m	0.163	1.240	6.220	0.250	0.762	0.916	0.968
Ours	-	CS + K	80m	**0.159**	**1.231**	**5.912**	**0.243**	**0.784**	**0.923**	**0.970**
Garg *et al.* [8]	Stereo	K	50m	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Zhou *et al.* [32]	-	K	50m	0.201	1.391	5.181	0.264	0.696	0.900	0.966
Zhou *et al.* [32]	-	CS + K	50m	0.190	1.436	4.975	0.258	0.735	0.915	0.968
Ours	-	K	50m	0.155	0.927	4.549	0.231	0.781	0.931	**0.975**
Ours	-	CS + K	50m	**0.151**	**0.949**	**4.383**	**0.227**	**0.802**	**0.935**	0.974

Table 1. Depth evaluation metrics over the KITTI Eigen [6] test set. Under the Dataset column, K denotes training on KITTI [10] and CS denotes training on Cityscapes [5]. δ denotes the ratio between estimates and ground truth. All results, except [6], use the crop from [8].

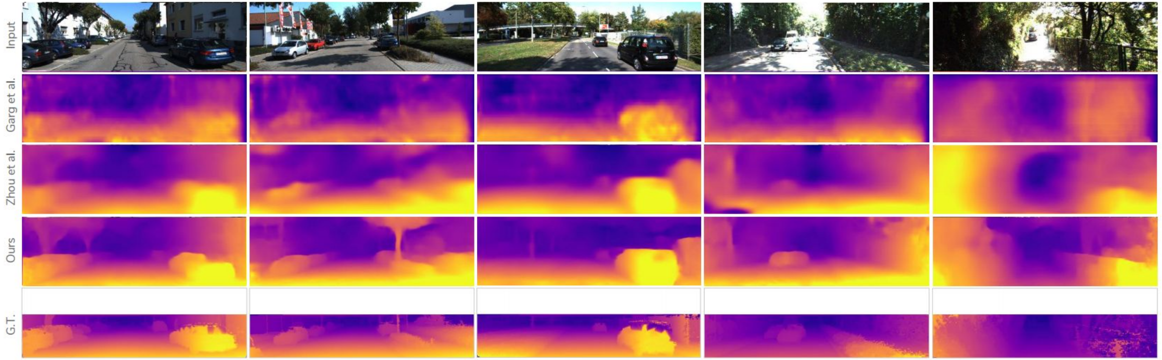
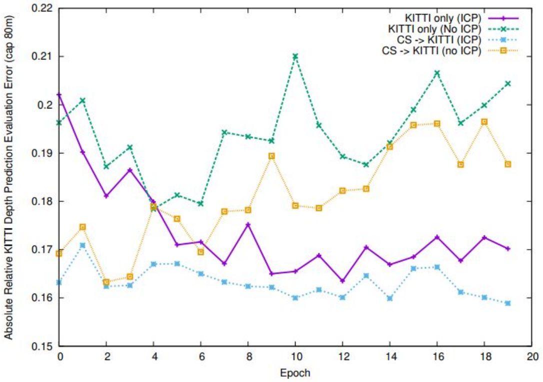
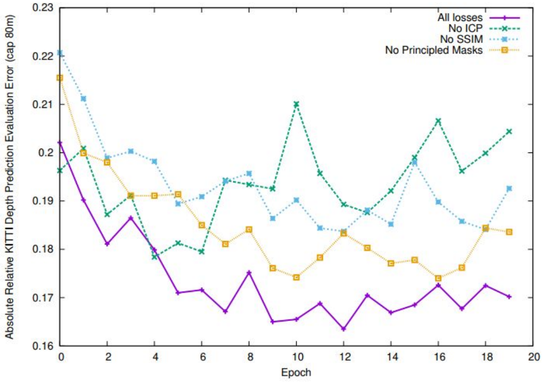


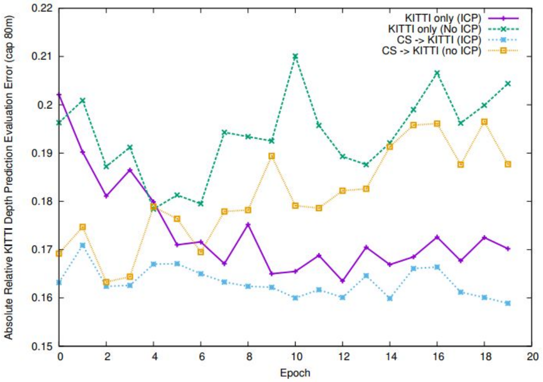
Figure 5. Sample depth estimates from the KITTI Eigen test set, generated by our approach (4th row), compared to Garg *et al.* [8], Zhou *et al.* [32], and ground truth [9]. Best viewed in color.





Vid2depth - results depth

Method	Dataset	Cap	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
All losses	CS + K	80m	**0.159**	**1.231**	**5.912**	**0.243**	**0.784**	**0.923**	**0.970**
All losses	K	80m	0.163	1.240	6.220	0.250	0.762	0.916	0.968
No ICP loss	K	80m	0.175	1.617	6.267	0.252	0.759	0.917	0.967
No SSIM loss	K	80m	0.183	1.410	6.813	0.271	0.716	0.899	0.961
No Principled Masks	K	80m	0.176	1.386	6.529	0.263	0.740	0.907	0.963
Zhou *et al.* [32]	K	80m	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Zhou *et al.* [32]	CS + K	80m	0.198	1.836	6.565	0.275	0.718	0.901	0.960
All losses	Bike	80m	0.211	1.771	7.741	0.309	0.652	0.862	0.942
No ICP loss	Bike	80m	0.226	2.525	7.750	0.305	0.666	0.871	0.946



- Removing artifacts
- Regularizing
- Blurring?

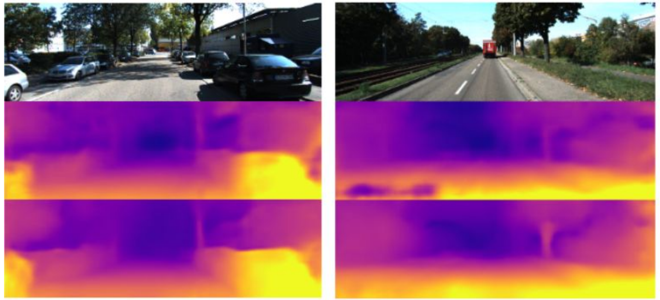
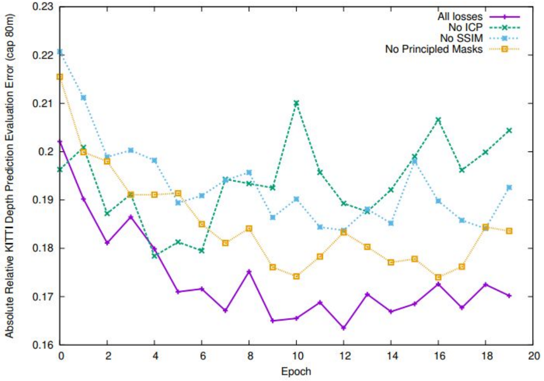


Figure 7. Example depth estimation results from training without the 3D loss (middle), and with the 3D loss (bottom).



Vid2depth - results path

Matches state-of-art on KITTI odometry:
- Without LIDAR
- Only 3 - frames at the time (no loop closure)

Method	**Seq.** 09	**Seq.** 10
ORB-SLAM (full)	0.014 ± 0.008	$\mathbf{0.012 \pm 0.011}$
ORB-SLAM (short)	0.064 ± 0.141	0.064 ± 0.130
Mean Odom.	0.032 ± 0.026	0.028 ± 0.023
Zhou *et al.* [32] (5-frame)	0.021 ± 0.017	0.020 ± 0.015
Ours, no ICP (3-frame)	0.014 ± 0.010	0.013 ± 0.011
Ours, with ICP (3-frame)	$\mathbf{0.013 \pm 0.010}$	$\mathbf{0.012 \pm 0.011}$

Vid2depth - problem

- Assumes static environment
- Too much moving object cause noise in learning and inference