Revisiting Scene Depth Estimation

Dr. Junjie Hu/胡君杰

Shenzhen Institute of Artificial Intelligence and Robotics for Society

List of Related Works

- **Junjie Hu**, Chenyou Fan, Xiyue Guo, Liguang Zhou and Tin Lun Lam. " Self-supervised Single-line LiDAR Depth Completion“, (RAL 2023 submission).
- **Junjie Hu**, Chenyu Bao, Mete Ozay, Chenyou Fan, Qing Gao, Honghia, Liu and Tin Lun Lam. "Deep Depth Completion from Extremely Sparse Data." IEEE Transactions on Pattern Analysis and Machine Intelligence. (TPAMI 2022).
- **Junjie Hu**, Chenyou Fan, Liguang Zhou, Qing Gao, Honghai Liu, Tin Lun Lam. "Lifelong-MonoDepth: Lifelong Learning for Multi-Domain Monocular Metric Depth Estimation." arXiv preprint arXiv:2303.05050, 2023. (TNNLS submission).
- **Junjie Hu**, Chenyou Fan, Mete Ozay, Hualie Jiang, Tin Lun Lam. "Data-free Dense Depth Distillation." arXiv preprint arXiv:2208.12464, 2022. (Neural Network submission).
- **Junjie Hu**, Chenyou Fan, Hualie Jiang, Xiyue Guo, Xiangyong Lu, Tin Lun Lam. "Boosting Light-Weight Depth Estimation Via Knowledge Distillation". The 16th International Conference on Knowledge Science, Engineering and Management. (KSEM 2023).
- **Junjie Hu** and Takayuki Okatani. "Analysis of Deep Networks for Monocular Depth Estimation Through Adversarial Attacks with Proposal of a Defense Method." arXiv preprint arXiv:1911.08790, 2019.
- **Junjie Hu**, Yan Zhang and Takayuki Okatani. "Visualization of Convolutional Neural Networks for Monocular Depth Estimation." 2019 IEEE International Conference on Computer Vision. (ICCV 2019).
- **Junjie Hu**, Mete Ozay, Yan Zhang and Takayuki Okatani. "Revisiting Single Image Depth Estimation: Toward Higher Resolution Maps with Accurate Object Boundaries." 2019 IEEE Winter Conference on Applications of Computer Vision. (WACV 2019).

Outline

- **Backgrounds**
- Prediction of High-Resolution Maps
- Visualization of CNNs
- Defending Adversarial Attacks
- Towards High Generalizability
- Improving Computational Efficiency
- Multi-modality Data Fusion
- Conclusions

Backgrounds

- What is depth map?
  - An image that contains information relating to the distance of the surfaces of scene objects from camera viewpoint.

RGB image

Depth map, distance increases from blue to red

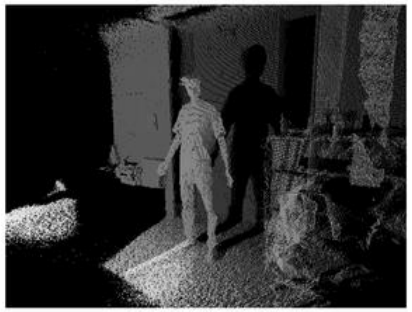
  - It’s also sometimes called 2.5D image, since it depicts 3D information of scene objects from only one viewpoint.

Backgrounds

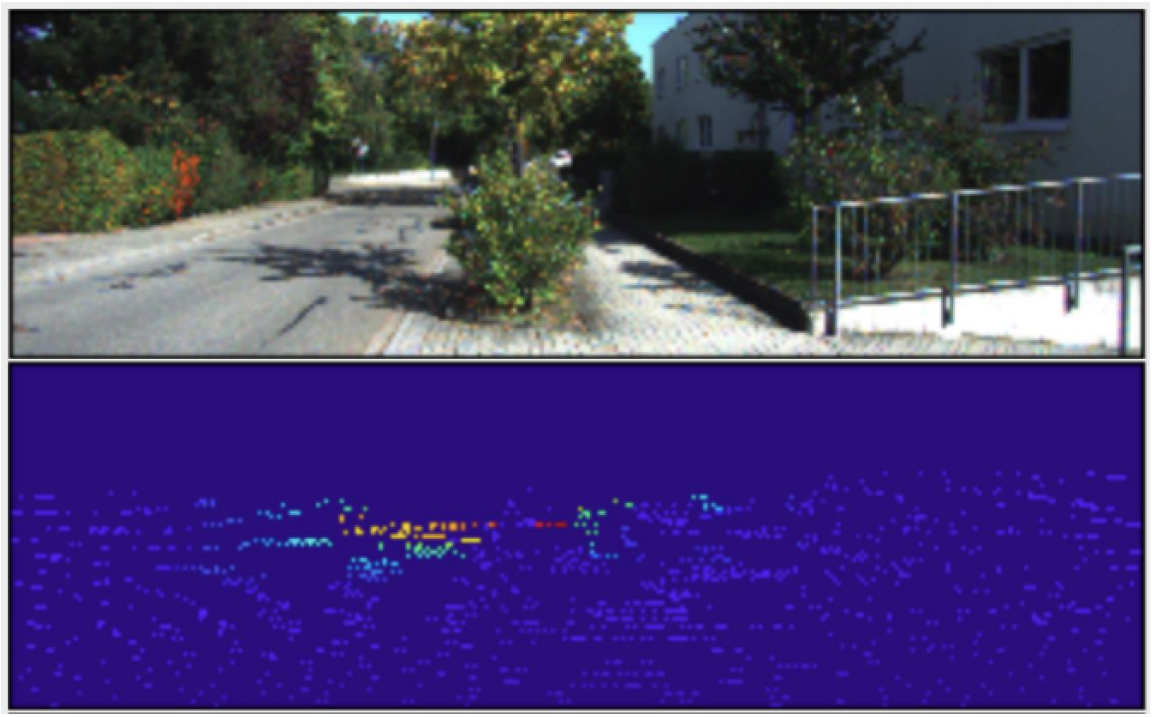
- Why do we need to estimate a depth map from RGB images?
  - Kinect can only be used for indoor scenes.
  - 3D Lidar sensor itself is very expensive and only produces very sparse depth maps.

(a)

(b)



(c)



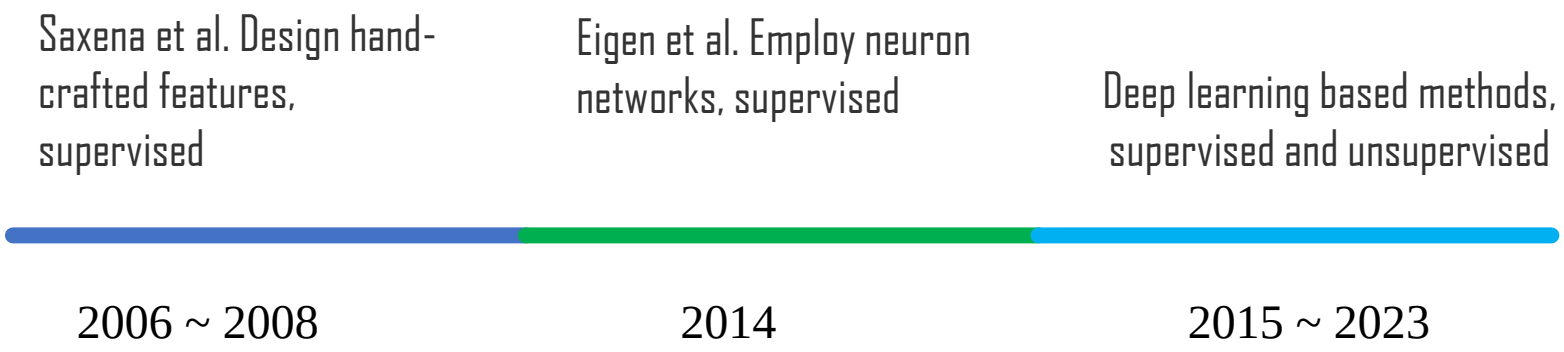
Backgrounds

- Why do we need to estimate a depth map from RGB images?
  - Depth map itself plays a significant role in 3D reconstruction, robot navigation, virtual reality, SLAM [7].
  - Depth map + RGB contributes to various vision tasks, performing better than using RGB images only.
    - RGBD based semantic segmentation [5,6].
    - RGBD based image recognition [1,2,4].
    - RGBD based human action recognition [3].
    - …

1. RGB-D Object Recognition Using Deep Convolutional Neural Networks. ICCV 2017.
2. Accurate and robust face recognition from RGB-D images with a deep learning approach. BMVC 2016.
3. Jointly learning heterogeneous features for RGB-D activity recognition. CVPR 2015.
4. Depth CNNs for RGB-D scene recognition: learning from scratch better than transferring from RGB-CNNs. AAAI 2017.
5. Depth-aware CNN for RGB-D Segmentation. ECCV 2018.
6. Cascaded Feature Network for Semantic Segmentation of RGB-D Images. ICCV 2017.
7. CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction. CVPR 2017.

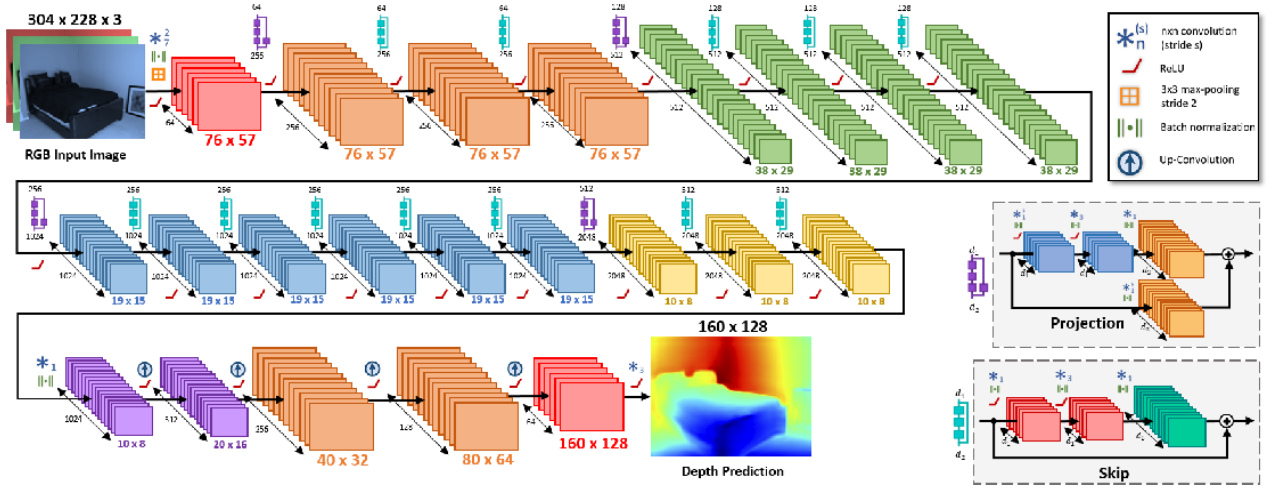
Backgrounds

- History of monocular depth estimation



Backgrounds

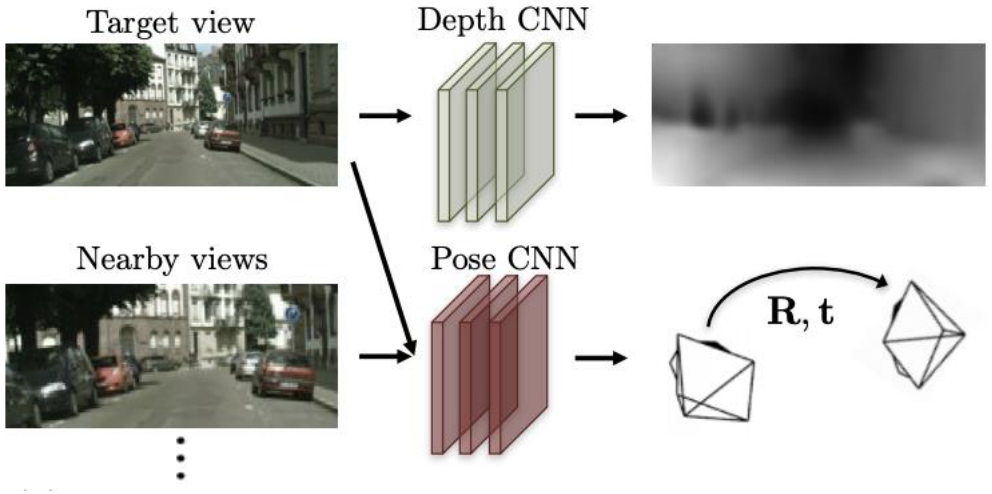
- How do we estimate a depth map from RGB images?
  - Supervised learning
    - Data-driven approach, needs a large amount of RGB-D data.
    - Networks: encoder-decoder network [1,2], dilated network [3], etc.
    - Loss: l_1 of depth [1], loss of depth, gradients and normals [2], ordinal loss [3].

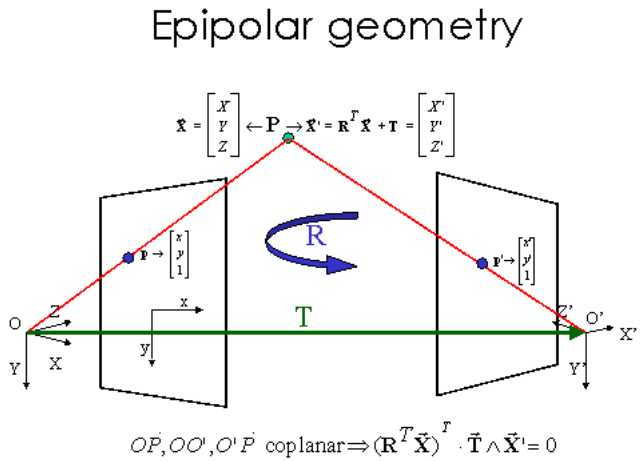


1. Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image. ICRA 2017.
2. Revisiting Single Image Depth Estimation: Toward Higher Resolution Maps With Accurate Object Boundaries. WACV 2019.
3. Deep Ordinal Regression Network for Monocular Depth Estimation. CVPR 2018.

Backgrounds

- How do we estimate a depth map from RGB images?
  - Unsupervised learning (Hot topic now [1,2,3,4])
    - Data-driven approach, needs a large amount of multi-view/stereo images.
    - Epipolar geometry or stereo matching for RGB→D transformation.





1. Unsupervised Monocular Depth Estimation with Left-Right Consistency. CVPR 2017.
2. Unsupervised Learning of Depth and Ego-Motion from Video. CVPR 2017.
3. Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction. CVPR 2018.
4. Unsupervised Learning of Stereo Matching. ICCV 2017.

Backgrounds

- Summary
  - Data driven approaches.
  - Supervised or unsupervised learning.
  - Promising results on various datasets (NYU-v2, KITTI, Make3D)
  - Facing several challenges that hinder its real-world deployment.
    - Accuracy (low perceptual quality).
    - Interpretability (Black-box that is not explainable to human).
    - Vulnerability (not robust to adversarial attacks).
    - Low generalizability (cannot generalize across domains).
    - Time-consuming (especially for robot applications).
    - Cannot infer metric depth in practice (many methods can only infer relative depth maps).

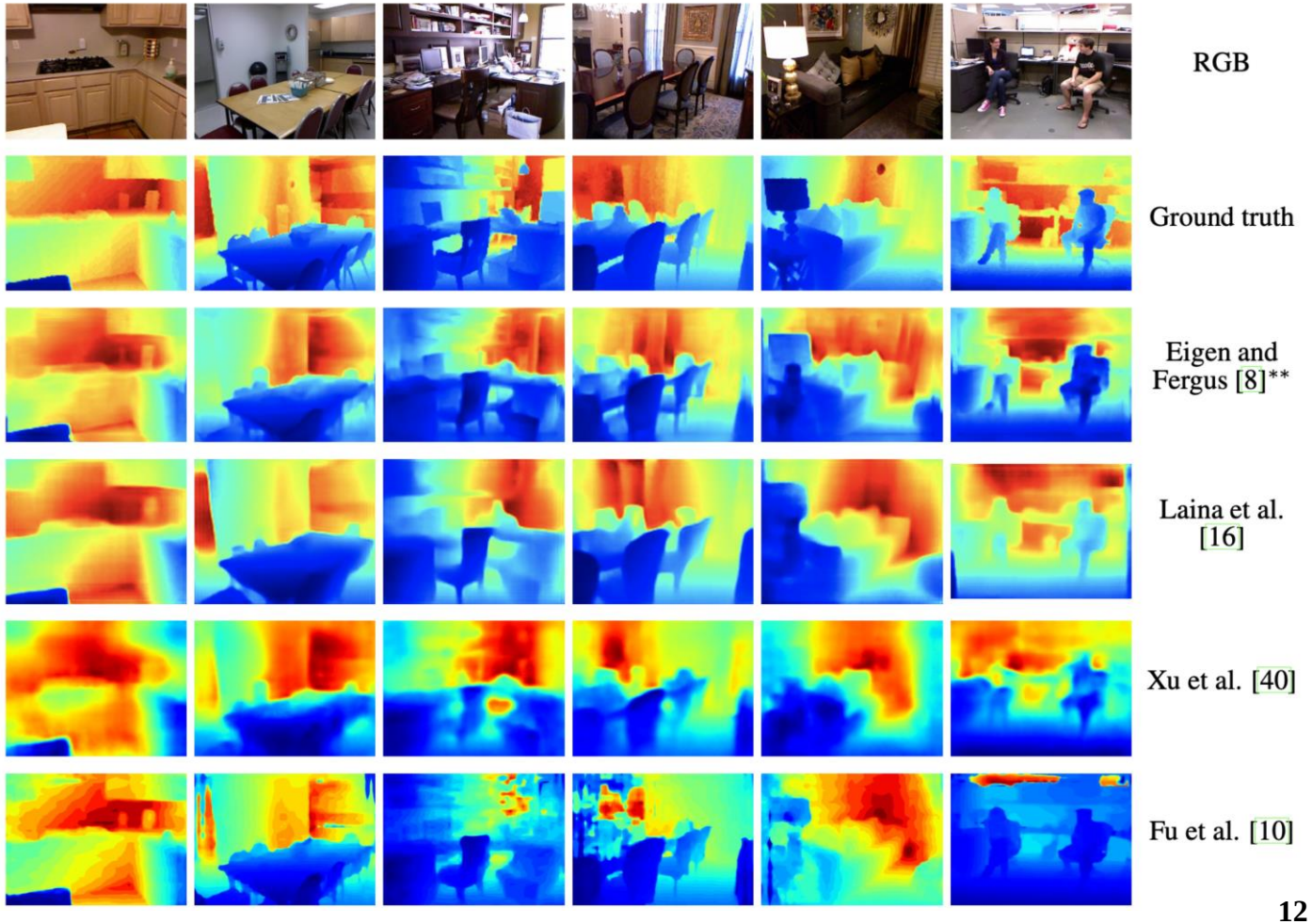
Outline

- Backgrounds
- **Prediction of High-Resolution Maps**
- Visualization of CNNs
- Defending Adversarial Attacks
- Generalizability on Multiple Domains
- Improving Computational Efficiency
- Multi-modality Data Fusion
- Conclusions

Motivation

- The depth maps estimated by previous methods suffer from

  - High structure distortion
  - Strong blurring effect
  - Missing small objects
  - Mosaic patterns



Solution

- We consider to improve it from two aspects
  - Improved network design
    - How to leverage multi-scale features?

Our network is also employed by several papers, including SLAM system and other depth estimation frameworks.

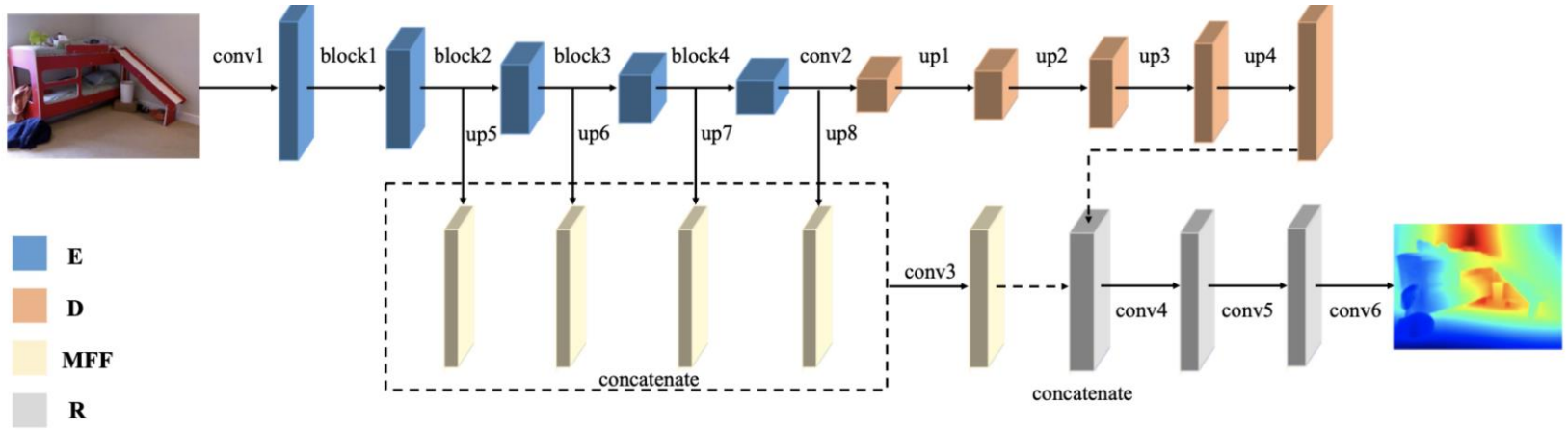
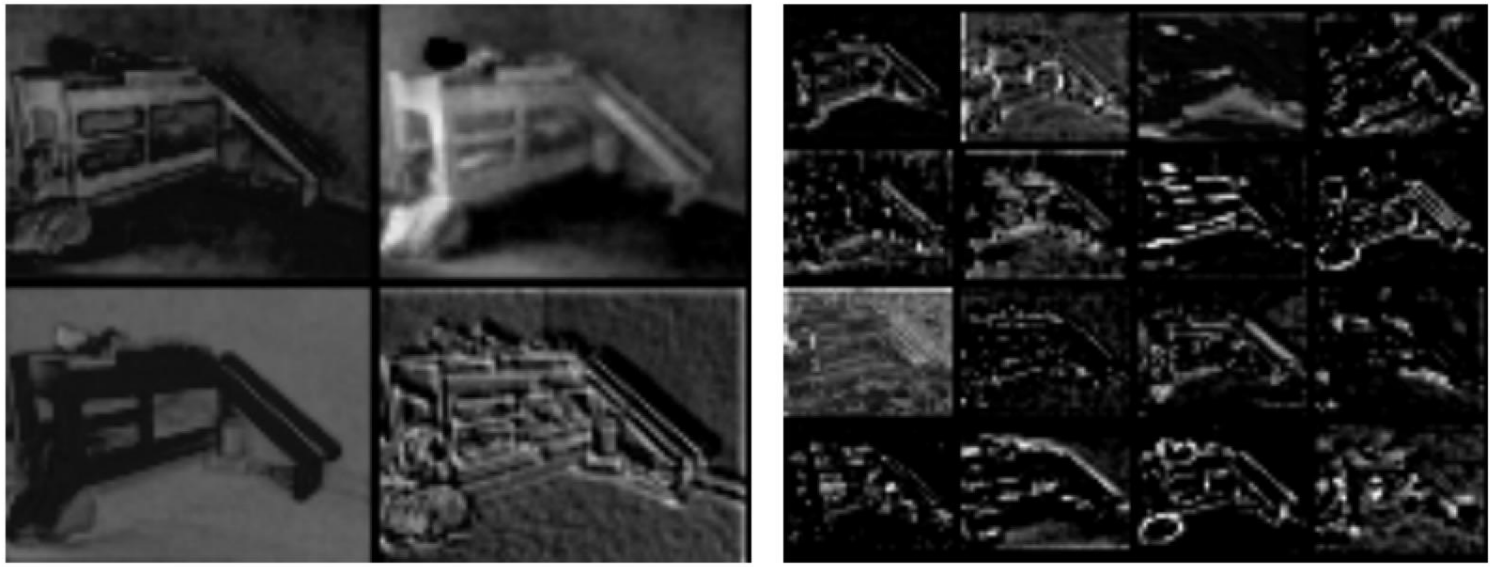


Figure 2. A diagram of the proposed network architecture. Given an input image, the encoder (E) extracts multi-scale features (1/4, 1/8, 1/16, and 1/32). The decoder (D) converts the last 1/32 scale feature to get a 1/2 scale feature. Each of the multi-scale features is up-scaled to 1/2 scale, and fused by the multi-scale feature fusion module (MFF). The outputs of D and MFF and are refined by the refinement module (R) to obtain the final depth map. Each box named "block$n$" denotes a block of multiple convolutional layers, such as residual block of ResNet; each box named "up$n$" denotes a up-projection layer introduced in [16]. Batch normalization and ReLU nonlinearity are applied to the output of each convolutional layer except conv6.

Junjie Hu, et al. "Revisiting Single Image Depth Estimation: Toward Higher Resolution Maps with Accurate Object Boundaries." WACV 2019

Solution



(a) (b)

Figure 4. Visualization of outputs of different layers of the encoder network for the input image shown in Fig. 2. Selected channels of (a) block1, and (b) block2.

Solution

- We consider to improve it from two aspects
  - Comprehensive loss function
    - How to leverage geometry properties?

$$\mathrm{L} = l_{depth} + l_{grad} + l_{normal}$$

    - Where l_{depth} l_{grad} are l_1 loss of depth and gradient in log space.
    - l_{normal} penalizes the cosine difference of two normals.

Solution

- We consider to improve it from two aspects
  - Comprehensive loss function
    - How to leverage geometry properties?

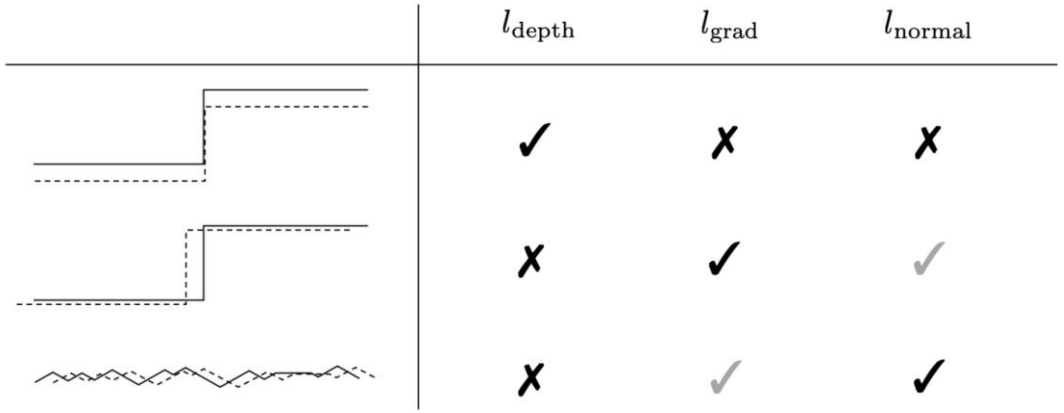


Figure 5. The three loss functions have orthogonal sensitivities to different types of errors of estimated depth maps. The solid and dotted lines depicted in the first column indicate two depth maps under comparison, where they are represented by one-dimensional depth images for the sake of explanation, and the vertical axis is depth and the horizontal axis is, say, the x axis of the images.

Experimental results

Table 2. Comparisons of different methods on the NYU-Depth V2 dataset. The methods marked by * use partially known depths, and those with ** employ joint task learning.

Method	RMS	REL	log 10	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen et al. [9]	0.907	0.215	-	0.611	0.887	0.971
Liu et al. [25]	0.824	0.230	0.095	0.614	0.883	0.971
Chakrabarti et al. [3]	0.620	0.149	-	0.806	0.958	0.987
Cao et al. [2]	0.819	0.232	0.091	0.646	0.892	0.968
Li et al. [24]	0.635	0.143	0.063	0.788	0.958	0.991
Ma and Karaman [28]	(-)	0.143	-	0.810	0.959	0.989
Laina et al. [16]	0.573	0.127	0.055	0.811	0.953	0.988
Xu et al. [40]	0.586	0.121	0.052	0.811	0.954	0.987
Lee et al. [21]	0.572	0.139	-	0.815	0.963	0.991
Fu et al. [10]	**0.509**	**0.115**	0.051	0.828	0.965	0.992
Qi et al. [32]	0.569	0.128	0.057	0.834	0.960	0.990
Ours (ResNet-50)	0.555	0.126	0.054	0.843	0.968	0.991
Ours (DenseNet-161)	0.544	0.123	0.053	0.855	0.972	0.993
Ours (SENet-154)	0.530	**0.115**	**0.050**	**0.866**	**0.975**	**0.993**
Ma and Karaman [28]*	(-)	0.044	-	0.971	0.994	0.998
Li et al. [23]	0.821	0.232	0.094	0.621	0.886	0.968
Eigen and Fergus [8]**	0.641	0.158	-	0.769	0.950	0.988
Dharmasiri et al. [6]**	0.624	0.156	-	0.776	0.953	0.989
Xu et al. [39]**	0.582	0.120	0.055	0.817	0.954	0.987

Experimental results

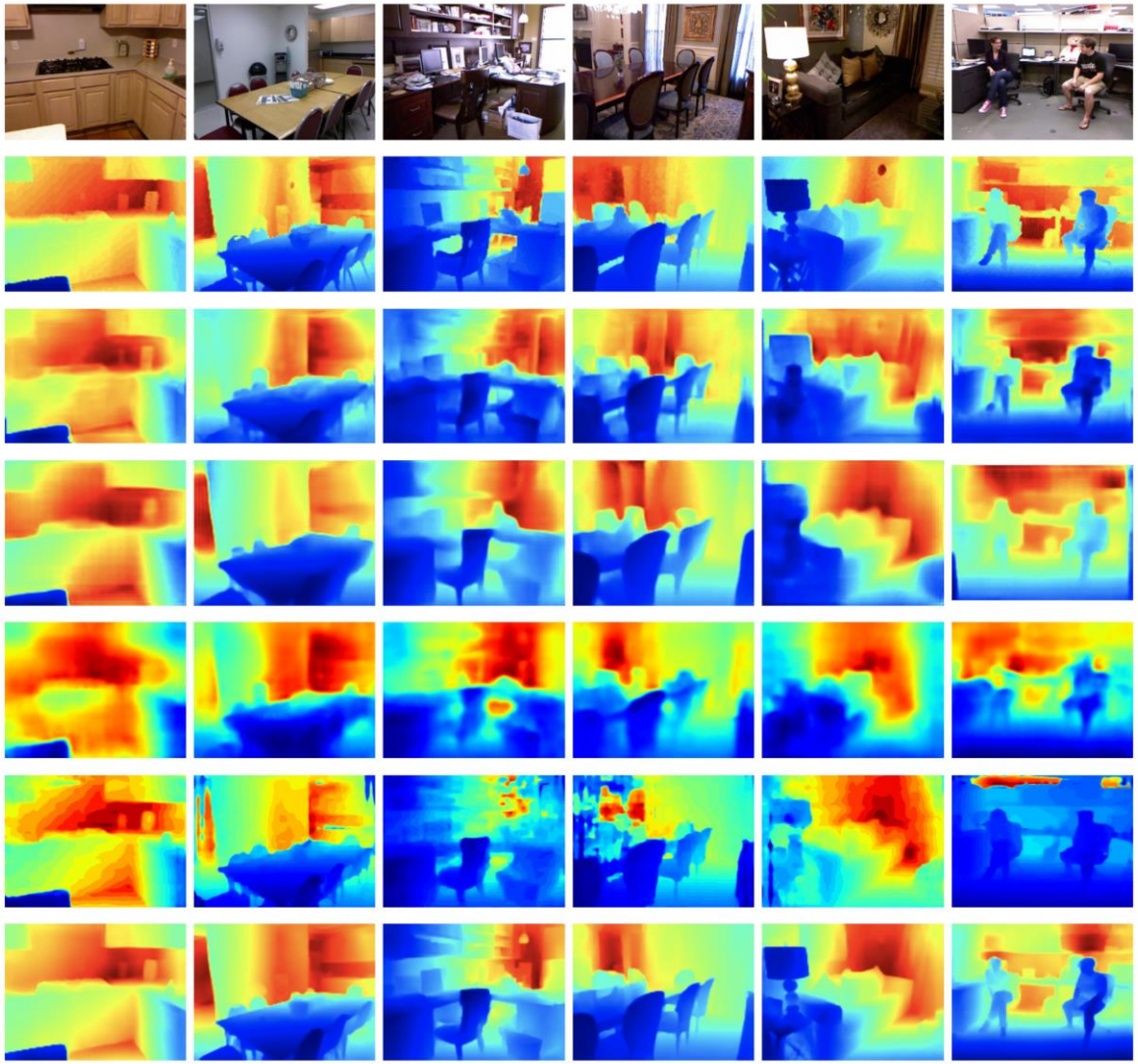


Figure 6. Results of different methods for six images. From the first to the last row; input RGB images, ground truth depth map, a multi-task learning method [8], encoder-decoder network [16], CRF-based method [40], dilated ordinary regression network [10], and our proposed network trained with the full loss function. We show them in the ascending order of quality in traditional measures.

Experimental results

- Ablation study of different loss functions

Table 4. Results of our method that is built on ResNet-50 trained with different loss functions on the NYU-Depth V2 dataset. For edge accuracy, we report the results for >0.5.

	RMS	REL	$\delta < 1.25$	F1
w/ l_{depth}	0.580	0.133	0.830	0.525
w/ $l_{\text{depth}} + \lambda l_{\text{grad}}$	0.563	0.128	0.841	0.543
w/ full loss	**0.555**	**0.126**	**0.843**	**0.548**

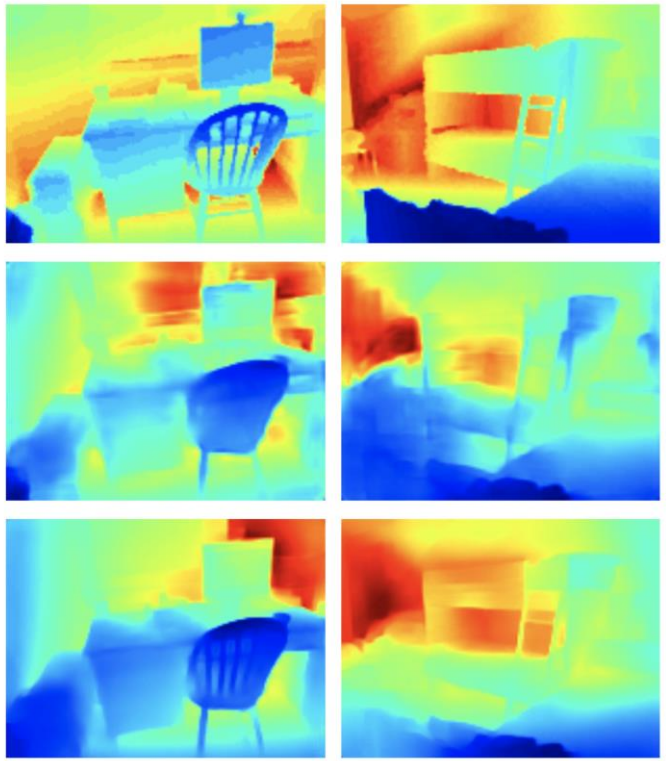


Figure 7. Visual comparison of our method trained using ResNet-50 with different loss functions on the NYU-Depth V2 dataset. From top to bottom: Ground Truth, trained with l_{depth} and the full loss, respectively.

Outline

- Backgrounds
- Prediction of High-Resolution Maps
- **Visualization of CNNs**
- Defending Adversarial Attacks
- Generalizability on Multiple Domains
- Improving Computational Efficiency
- Multi-modality Data Fusion
- Conclusions

Motivation

- Why CNNs can accurately predict a depth map only from single image? (we only consider the supervised methods.)
  - It's black box as in other tasks!
  - A fundamental, yet hasn't been explored/answered question.
  - It captures lots of attentions due to safety concerns, *e.g. self-driving cars, robots' navigation.*

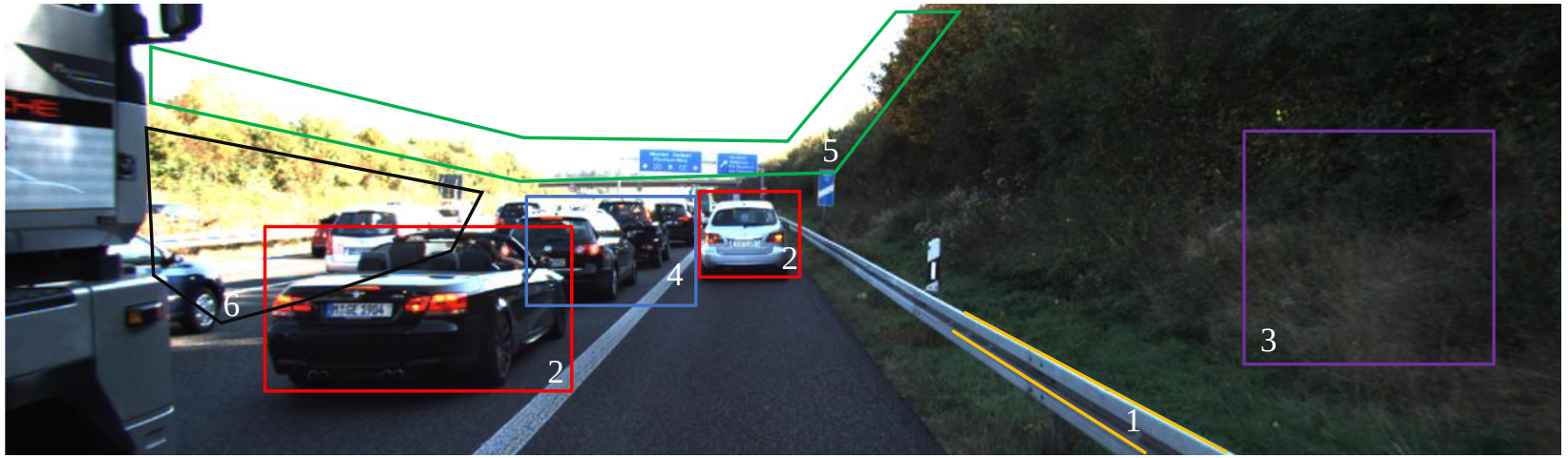
Motivation

- Why CNNs can accurately predict a depth map only from single image?
  - Interpretability
    - Find/visualize the evidence for supporting CNN’s decision.
    - The evidence should be explainable and convincible to human.

  - We attempt to answer the question from the following aspect:
    - Which pixels of an image are most relevant to depth inference?

Motivation

- Why CNNs can accurately predict depth map only from single image?
  - Which pixels of an image are most relevant to depth inference?
    - Human vision (monocular) only use several cues to perceive depth.



1. Linear perspective
2. Relative size
3. Texture gradient
4. Interposition
5. Aerial perspective
6. Light and shades

Related Work

- Visualization of neuron networks.

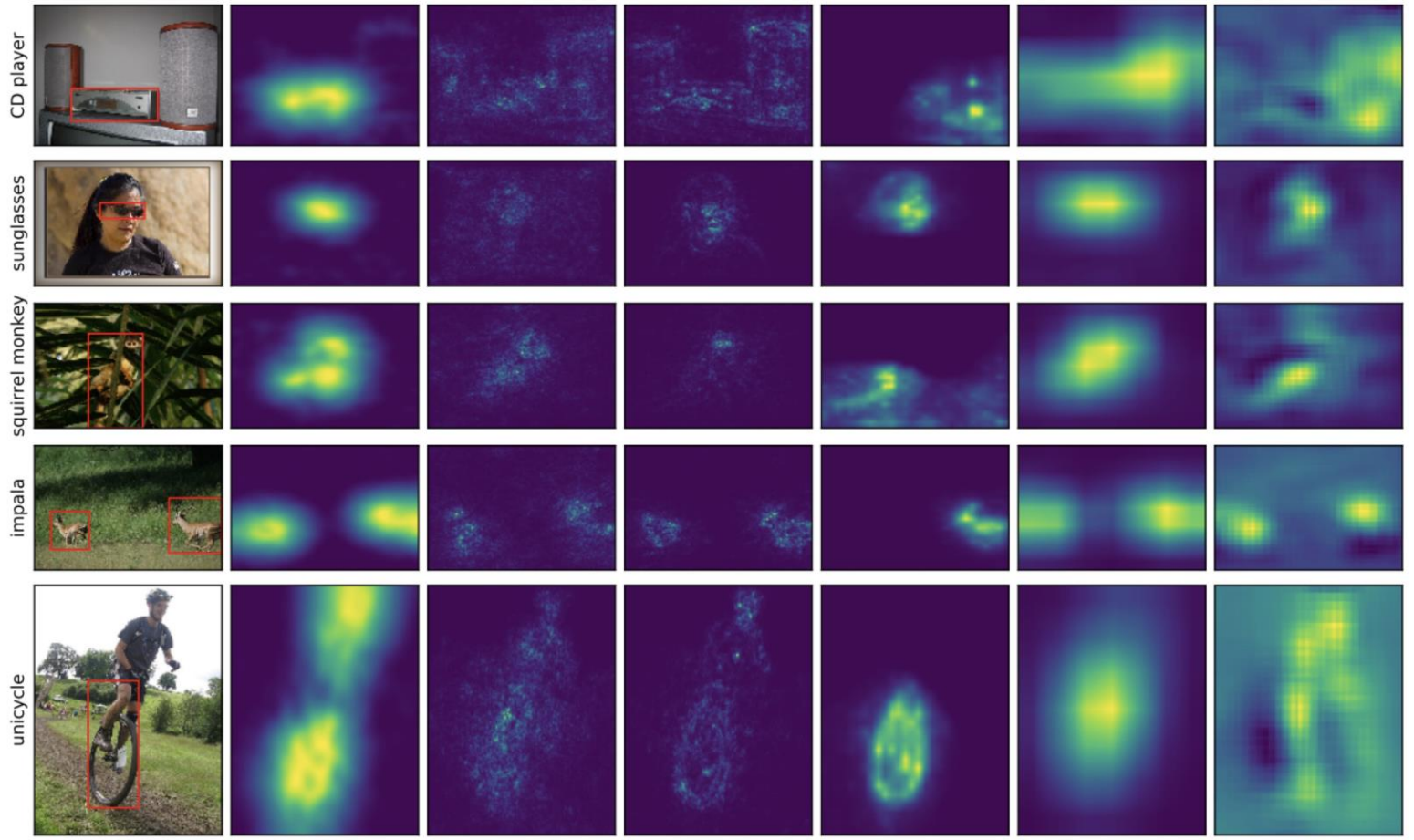
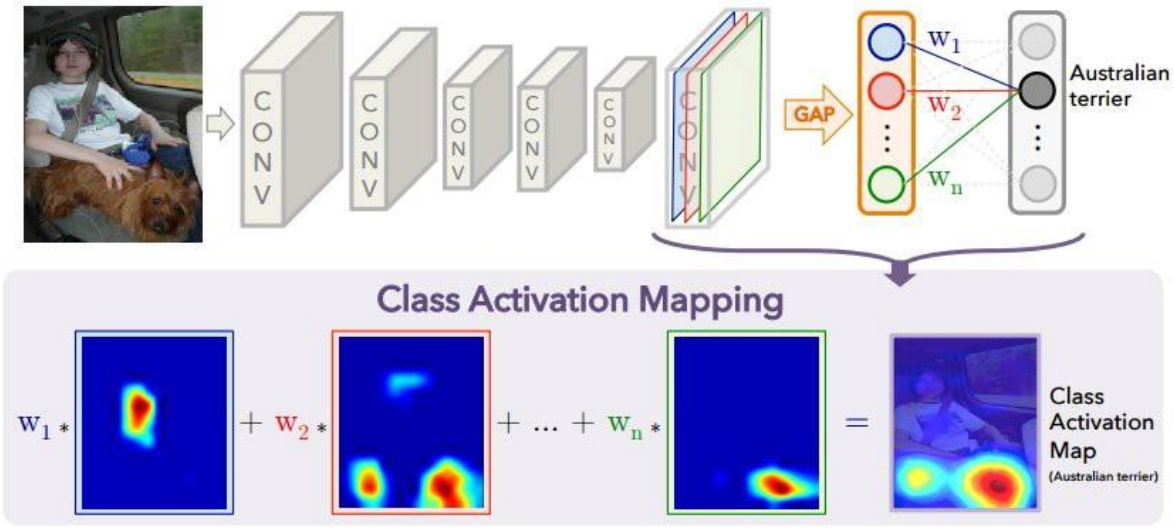


Figure 2. Comparison with other saliency methods. From left to right: original image with ground truth bounding box, learned mask subtracted from 1 (our method), gradient-based saliency [15], guided backprop [16, 8], contrastive excitation backprop [20], Grad-CAM [14], and occlusion [19].

Related Work

- Mask/Saliency visualization is a common approach for understanding useful features for neuron networks.
  - Usually based on backpropagation

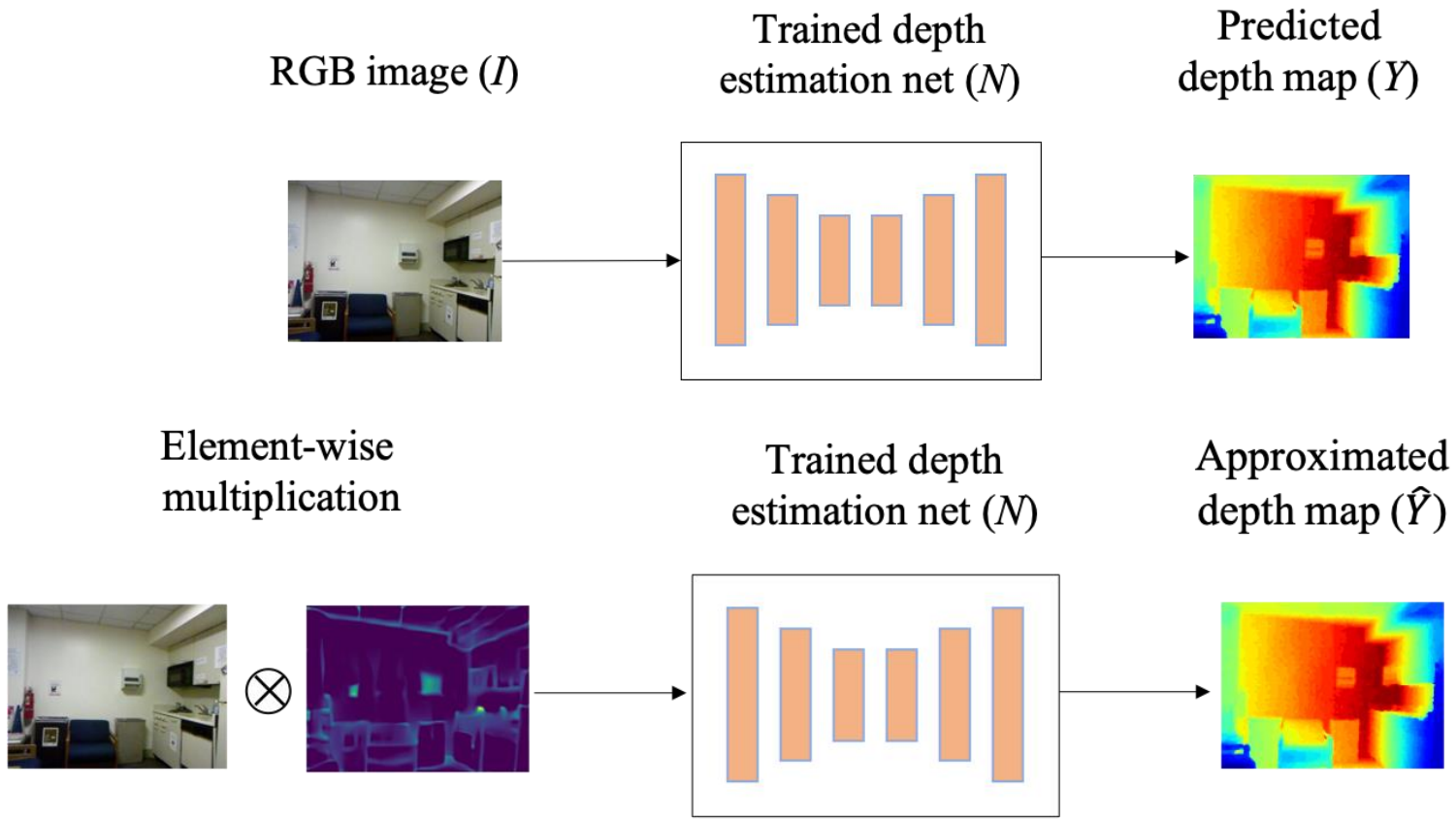
- However, previous methods cannot be applied to depth estimation task **as the output is a depth map not a score.**



Zhou eta al. Learning Deep Features for Discriminative Localization. CVPR 2015.

Solution

- We assume CNNs can infer depth map equally well from a selected set of sparse pixels of an image, as long as they are relevant to depth estimation.



Junjie Hu, et al. "Visualization of Convolutional Neural Networks for Monocular Depth Estimation." 2019 IEEE International Conference on Computer Vision. (ICCV 2019).

Solution

- We assume CNNs *can* ***infer depth map equally*** well from a selected ***set of sparse pixels*** of an image, as long as they are relevant to depth estimation.
  - Then the problem is formulated as:
    - l_{dif} is error between two depth images.

$$\min_{M} l_{dif}\left(Y, \hat{Y}\right) + \lambda \|M\|_1$$

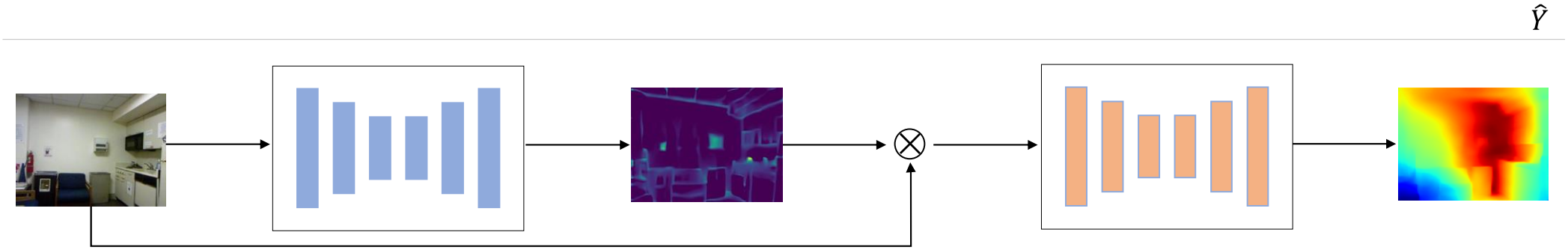
Solution

- Solving this optimization problem will yield unpredictable results, typical examples are the adversarial examples.

(a) (b)

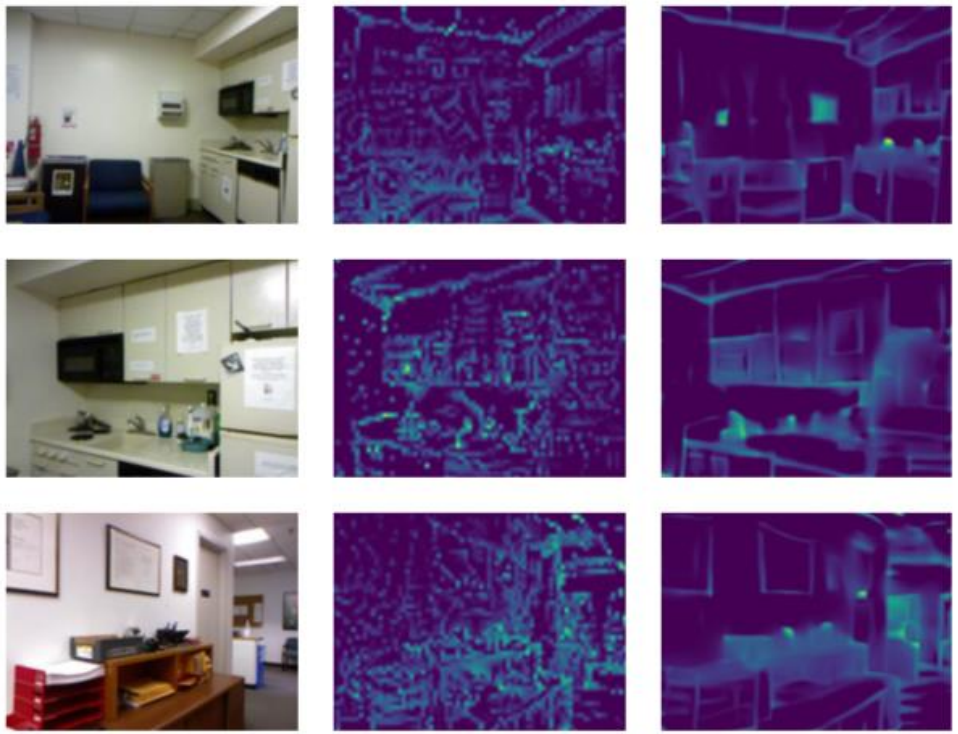
Solution

- we use a network G to predict $M = G(I)$.
  - It can predict more interpretable mask with less airfacts.
  - It can efficiently predict masks for a whole dataset.



Solution

- Masks predicted by solving the optimization function and the network G are shown in (b) and (c), respectively.



(a) (b) (c)

Solution

- Algorithm for learning masks.

Algorithm 1 Algorithm for training the network G for prediction of M.

Input: N: a target, fully-trained network for depth estimation; ψ: a training set, *i.e.*, pairs of the RGB image and depth map of a scene; λ: a parameter controlling the sparseness of M.

Hyperparameters: Adam optimizer, learning rate: $1e^{-4}$, weight decay: $1e^{-4}$, training epochs: K.

Output: G: a network for predicting M.

Freeze N;
for $j = 1$ to K **do**
  for $i = 1$ to T **do**
    Select RGB batch ψ_i from ψ;
    Set gradients of G to 0;
    Calculate depth maps for ψ_i:
      $Y_{\psi_i} = N(\psi_i)$;
    Calculate the value (L) of objective function:
    $L = l_{\text{dif}}(Y_{\psi_i}, N(\psi_i \otimes G(\psi_i)) + \lambda \frac{1}{n} \|G(\psi_i)\|_1$;
    Backpropagate L;
    Update G;
  end for
end for

Experimental results

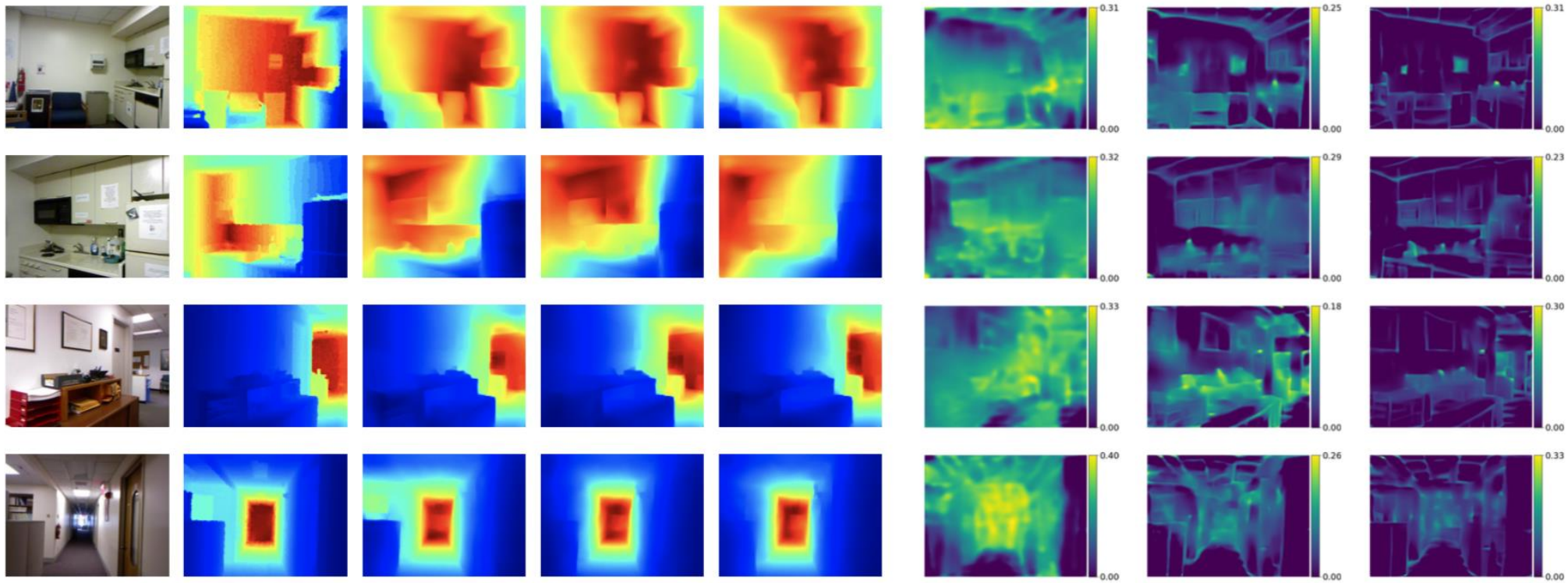
- Are the predicted masks correct?
  - 33% increase of RMSE for $\lambda = 5$, which is acceptable considering the accuracy-interpretability trade-off that is also seen in many visualization studies.

Table 1. Accuracy of depth estimation for different values of the sparseness parameter λ. Results on the NYU-v2 dataset by the ResNet-50 model of [14]. Sparseness in the table indicates the average number of non-zero pixels in M'.

λ	RMSE (M)	RMSE (M')	Sparseness
original	0.555	0.555	1.0
$\lambda = 1$	0.605	0.568	0.920
$\lambda = 2$	0.668	0.617	0.746
$\lambda = 3$	0.699	0.668	0.589
$\lambda = 4$	0.731	0.733	0.425
$\lambda = 5$	0.740	0.758	0.361
$\lambda = 6$	0.772	0.882	0.215

Experimental results

- Predicted masks while varying the sparseness parameter λ.



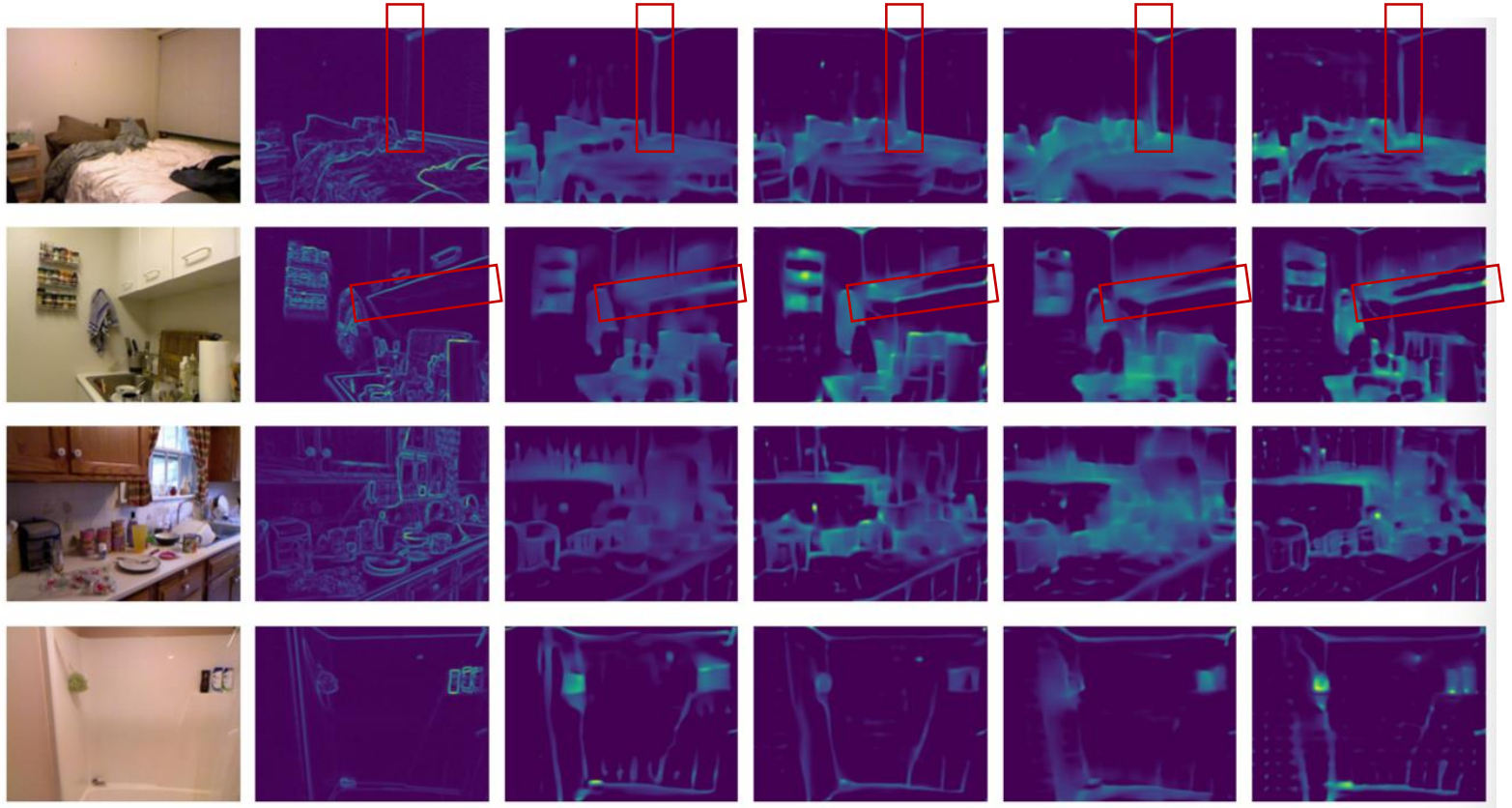
(a) RGB images | (b) Ground truth | (c) $\hat{Y}$ when λ=1 | (d) $\hat{Y}$ when λ=3 | (e) $\hat{Y}$ when λ=5 | (f) M when λ=1 | (g) M when λ=3 | (h) M when λ=5

Figure 4. Visual comparison of approximated depth maps and estimated masks (M's) for different values of the sparseness parameter λ.

Experimental results

- Predicted masks on the NYU-v2 dataset for different depth estimation networks.

Not all edges are important

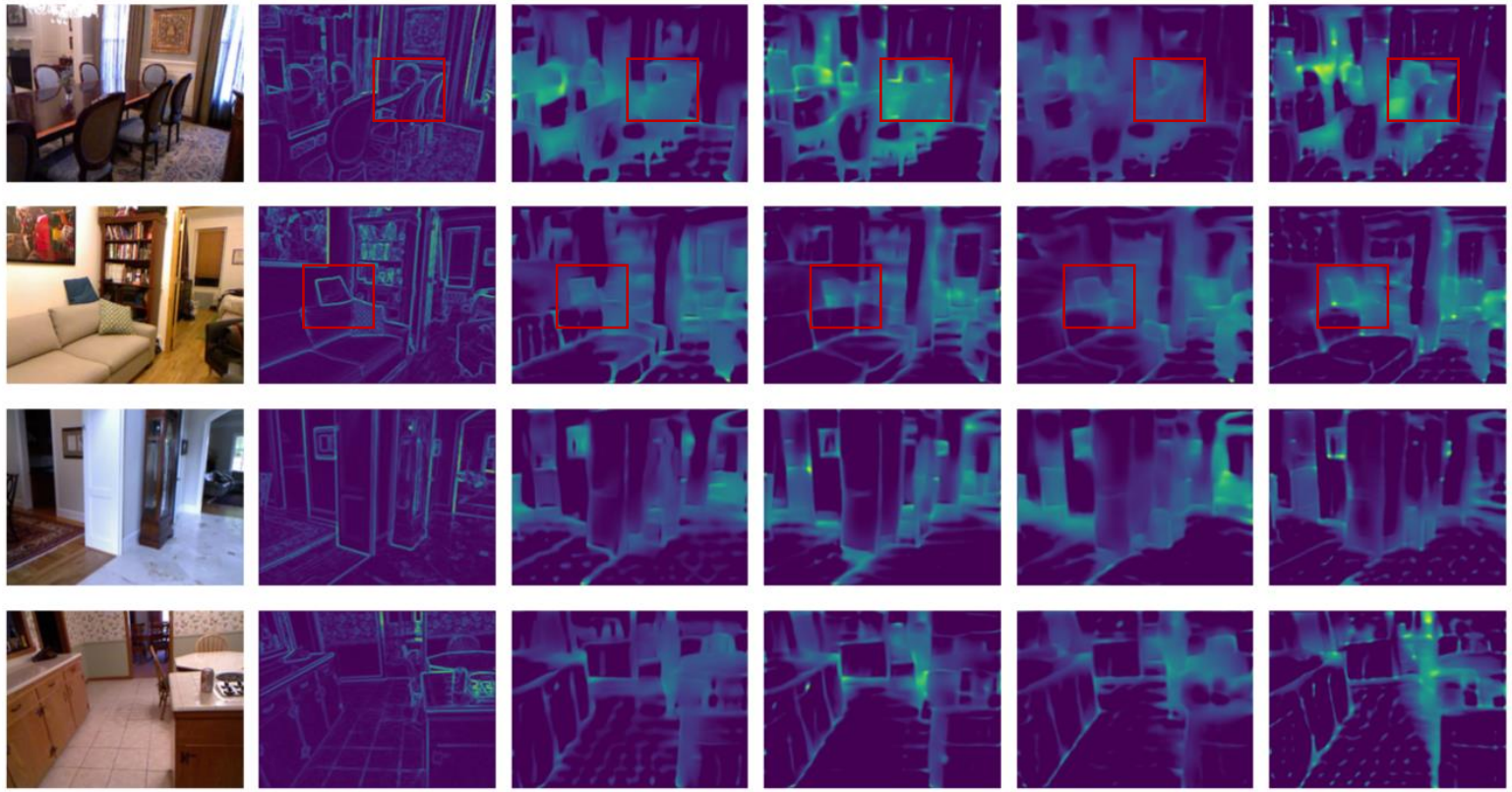


(a) RGB images | (b) Edge maps | (c) M for [16] (ResNet-50) | (d) M for [14] (ResNet-50) | (e) M for [14] (DenseNet-161) | (f) M for [14] (SENet-154)

Experimental results

- Predicted masks on the NYU-v2 dataset for different depth estimation networks.

Inside of regions are also important

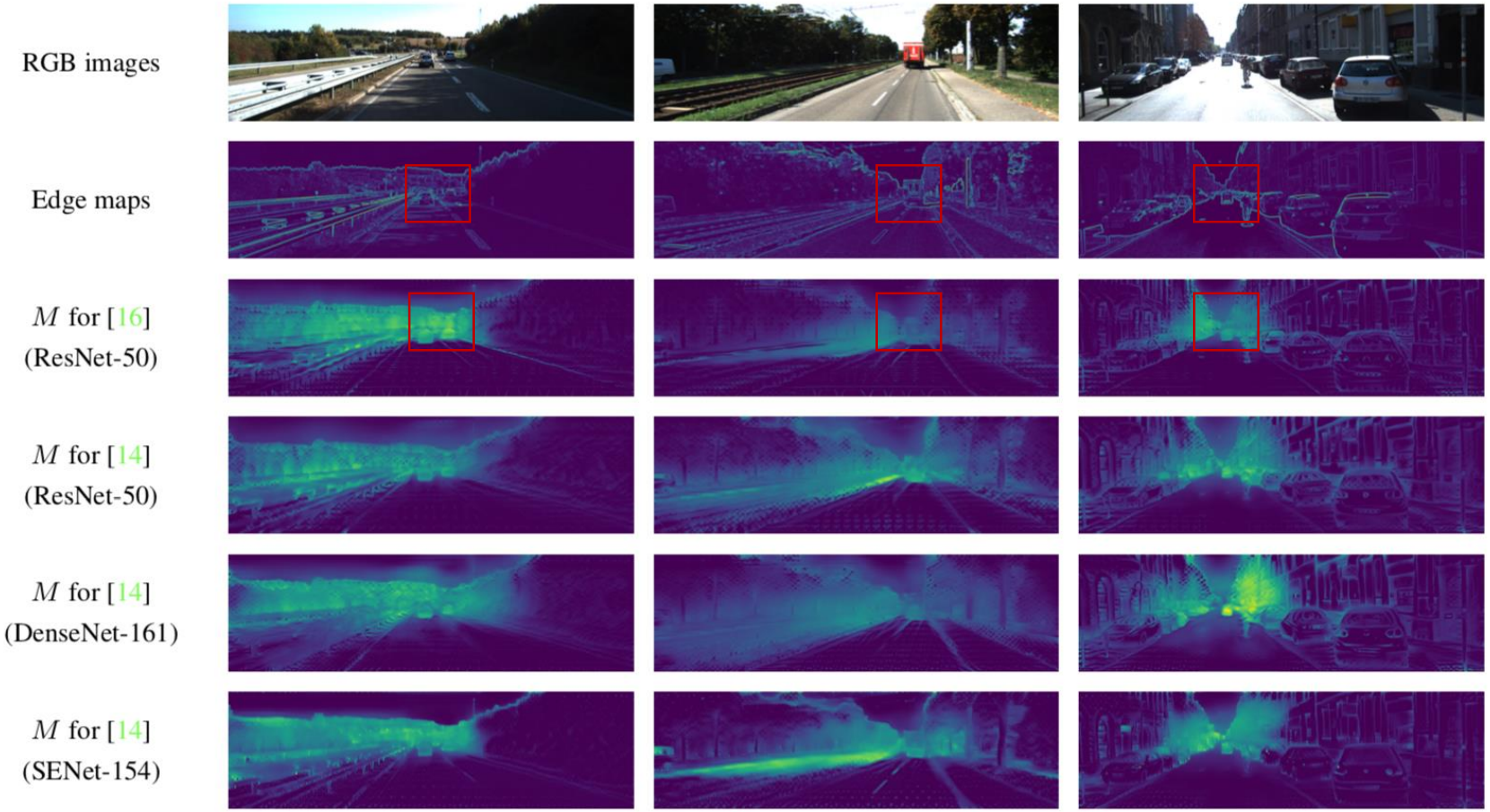


(a) RGB images | (b) Edge maps | (c) M for [16] (ResNet-50) | (d) M for [14] (ResNet-50) | (e) M for [14] (DenseNet-161) | (f) M for [14] (SENet-154)

Experimental results

- Predicted masks on the KITTI dataset for different depth estimation networks.

Vanishing points are important

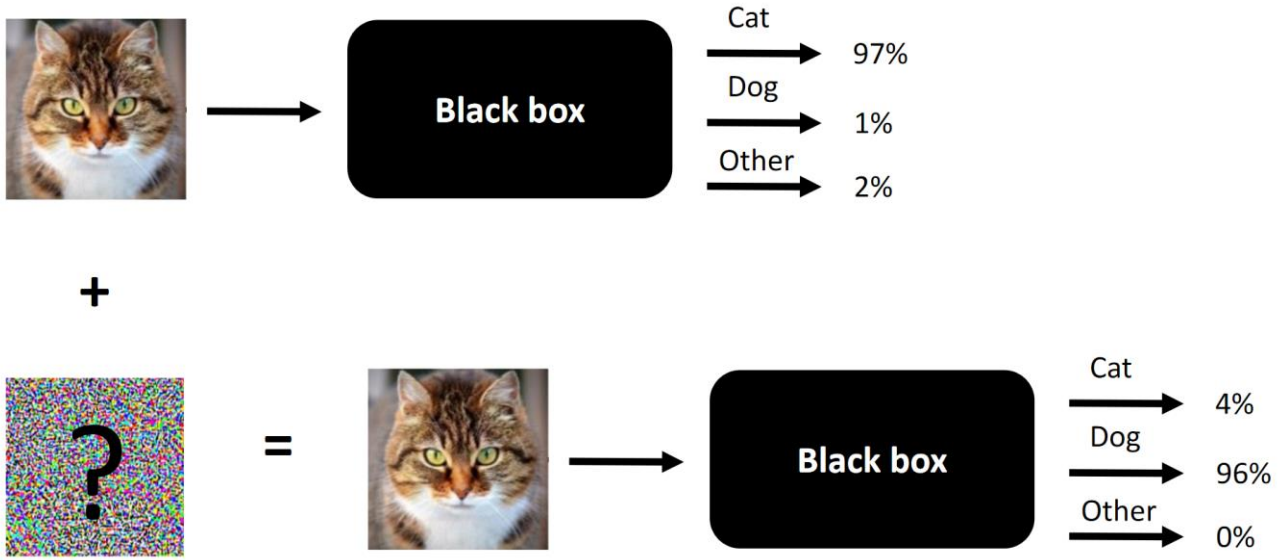


Outline

- Backgrounds
- Prediction of High-Resolution Maps
- Visualization of CNNs
- **Defending Adversarial Attacks**
- Generalizability on Multiple Domains
- Improving Computational Efficiency
- Multi-modality Data Fusion
- Conclusions

Motivation

- Neuron networks are known to be vulnerable to adversarial attacks.



Motivation

- Vulnerability of CNNs based depth estimators to adversarial attacks.

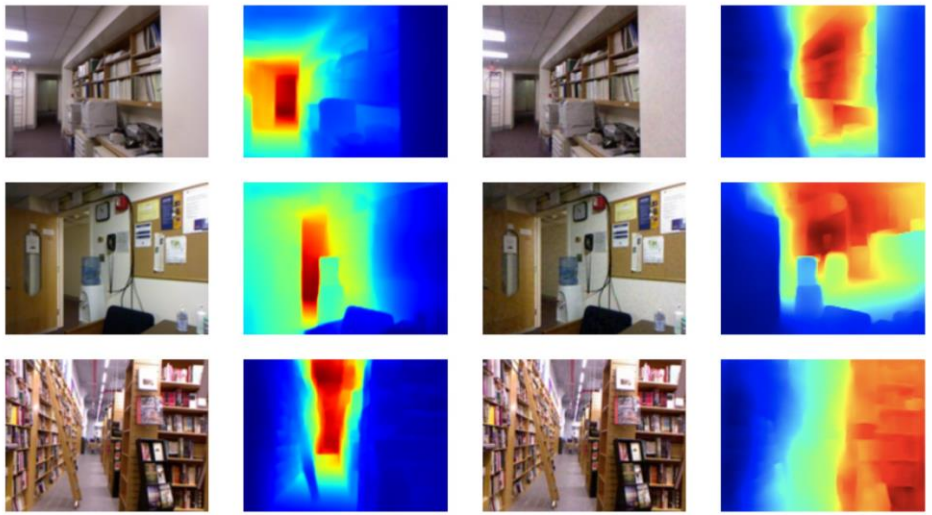


Figure 1. Vulnerability of CNNs to adversarial examples on the monocular depth estimation task. (a) Input. (b) Estimated depth from (a). (c) Adversarial input created by IFGSM with $\epsilon = 0.1$ and 10 iterations; see Eq.(1) for details. (d) Estimate from (c).

Motivation

- As our previous work show CNNs use similar cues as human vision, then
  - Why CNNs can be easily fooled with imperceptible noise for human vision?
  - Human vision can still correctly perceive depth even from adversarial images.
  - **We conjecture that adversarial attacks are made possible because of non-salient pixels.**

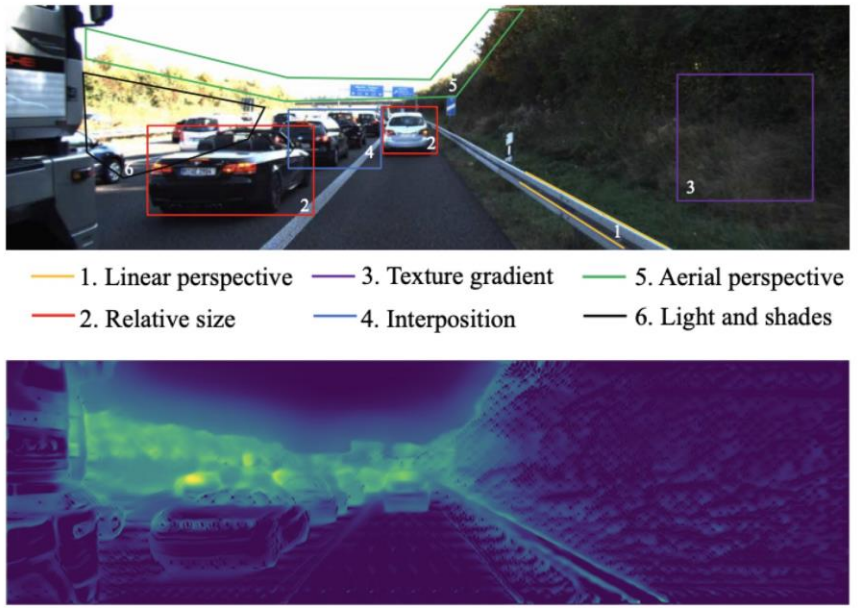


Figure 1. The upper row shows six monocular cues that are considered to be used for depth perception in human vision. The lower row shows the mask predicted by our method.

Solution

- **We conjecture that adversarial attacks are made possible because of non-salient pixels.**
  - As seen in Table1, the increase of RMSE is large for adversarial attack , i.e. $N(x^*)$..
  - The increase of RMSE is small if we use clean images to estimate saliency maps, i.e. $N(x^* \otimes G(x))$.

Table 1. Results of FGSM attack on several configurations of the depth estimation network N of [10]. The numbers are RMSE values over the test set of NYU-v2; x^* indicates the adversarial input given by $x^* = Adv(x; N)$; G is the saliency estimation network. N and G are trained on clean images. $\epsilon = 0$ means no attack.

	$\epsilon = 0$	$\epsilon = 0.05$	$\epsilon = 0.1$
$N(x^*)$	0.555	1.055	1.139
$N(x^* \otimes G(x^*))$	0.683	0.813	0.943
$N(x^* \otimes G(x))$	0.683	0.696	0.712

Solution

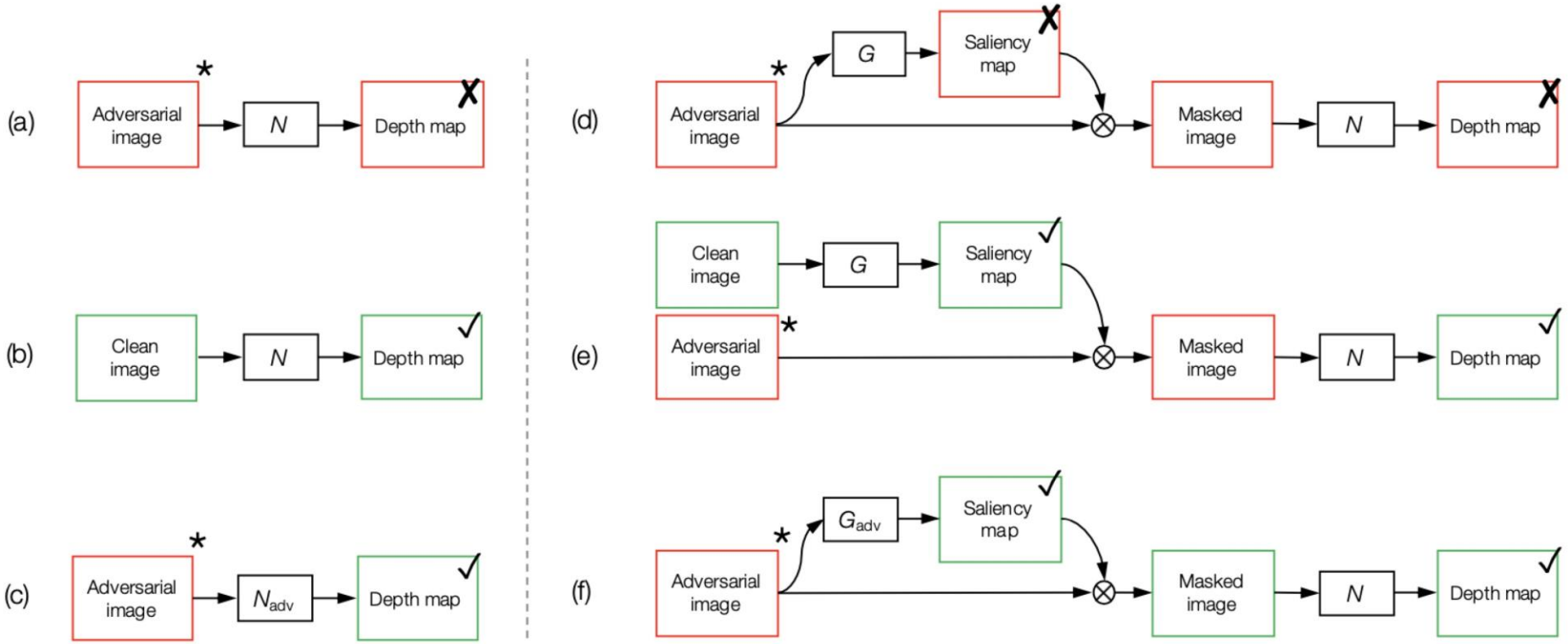


Figure 3. Configurations of depth estimation network N and saliency estimation network G considered in this paper. (a) $N(x^*)$: N is vulnerable to an adversarial input x^*. (b) $N(x)$: N yields an accurate estimate from a clean image x. (c) $N_{\mathrm{adv}}(x^*)$: An estimator N_{adv} robust to x^*. (d) $N(x^* \otimes G(x^*))$: N combined with G remains to be vulnerable. (e) $N(x^* \otimes G(x))$: An accurate estimate is obtained when the clean image is input to G. (f) $N(x^* \otimes G_{\mathrm{adv}}(x^*))$: A robust estimator G_{adv} trained by using adversarial inputs can be used to defend the attack. All the adversarial images (with *) are identical; it is tailored for N in setting (a).

Junjie Hu and Takayuki Okatani. "Analysis of Deep Networks for Monocular Depth Estimation Through Adversarial Attacks with Proposal of a Defense Method." arXiv preprint arXiv:1911.08790, 2019.

Solution

- Our problem is formulated as

$$G_{\text{adv}} = \underset{G}{\text{argmin}}\ \ell_{\text{dif}}(\bar{y}, N(x' \otimes G(x'))) + \lambda \frac{1}{n} \|G(x')\|_1,$$

where x' is either a clean image x or adversarial image x^*

- Algorithm for learning G is shown at right.

Algorithm 1 Algorithm for training the saliency prediction network G. The batch size is set to 1 for simplicity.

Input: N: a target, fully-trained network for depth estimation; χ: a training set, *i.e.*, pairs of an RGB image of a scene and its depth map; ϵ: ℓ_∞ bound for IFGSM; K: training epochs; and J: iterations per epoch.

Output: G_{adv}: a network for predicting a saliency map.

for $k = 1$ to K **do**
  for $j = 1$ to J **do**
    Select an RGBD pair $\{x, \bar{y}\}$ from χ
    $p = \text{Uniform}(0, 1)$
    if $p > 0.5$ **then**
      $\epsilon = \text{Uniform}(0.01, 0.3)$
      $T = \lfloor \text{Uniform}(1, 10) \rfloor$
      $x_0^* = x$
      $t = 0$
      for $t = 1$ to T **do**
        $x_{t+1}^* = \text{IFGSM}(x_t^*, \epsilon)$
      end for
      $x' = x_t^*$
    else
      $x' = x$
    end if
    $L = \ell_{\text{dif}}(\bar{y}, N(x' \otimes G(x'))) + \lambda \frac{1}{n} \|G(x')\|_1$
    Backpropagate L
    Update G
  end for
end for
$G_{\text{adv}} \leftarrow G$

Experimental Results

- Visual results of estimated depth maps.

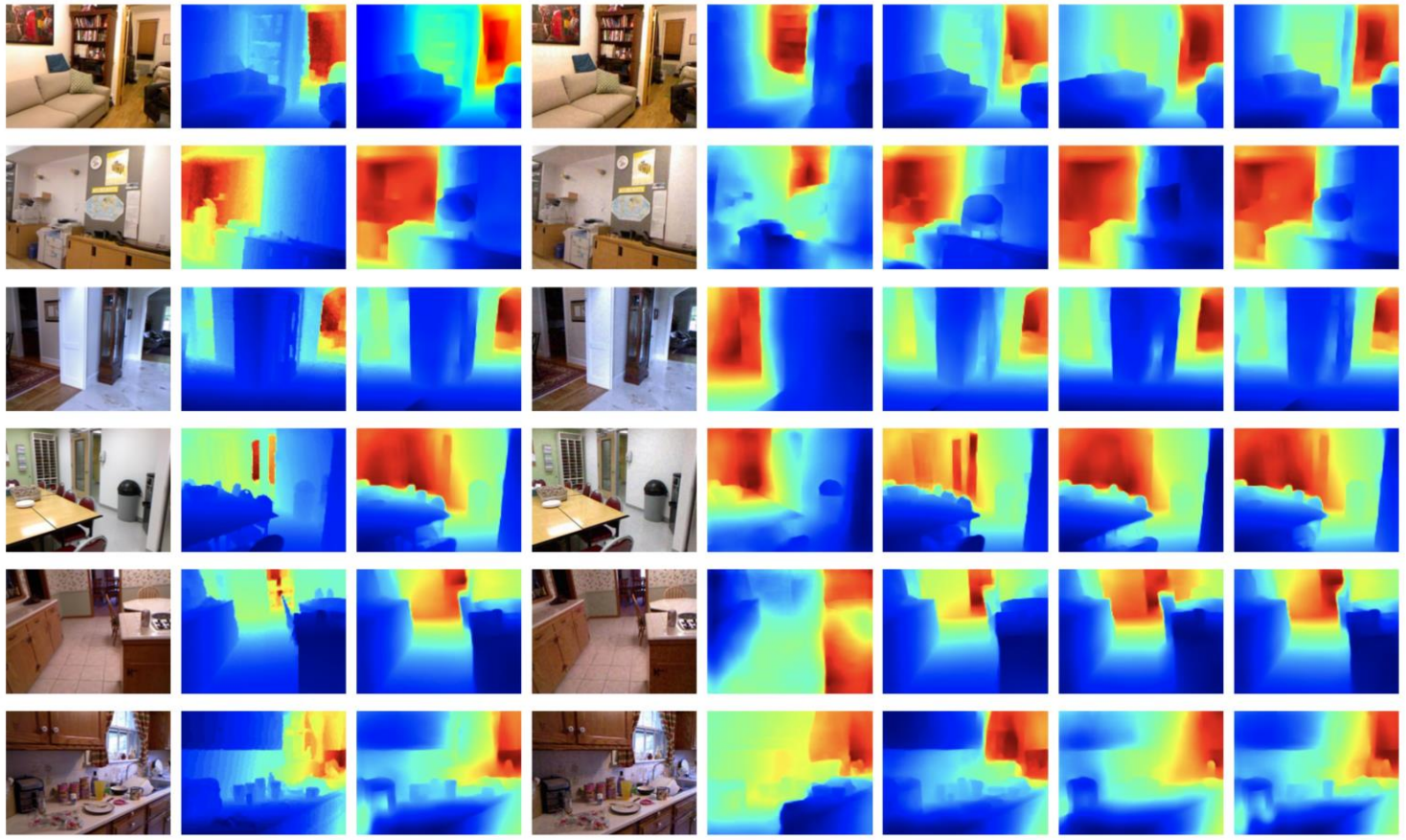


Figure 6. Visual comparisons of depth maps estimated from adversarial inputs x^*'s generated by IFGSM with $\epsilon = 0.1$ and 10 iterations; x's are clean inputs.

Experimental Results

- Performance against FGSM attacks.

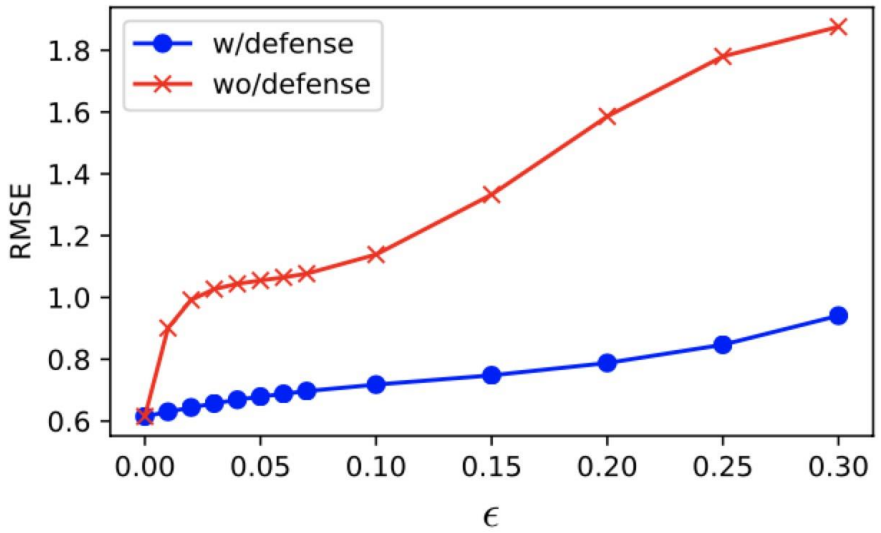


Figure 4. The RMSE of the depth maps estimated by $N(x^*)$ ('w/o defense') and $N(x^* \otimes G_{\text{adv}}(x^*))$ ('w/defense') from adversarial inputs generated by FGSM with different perturbation strength ϵ.

Experimental Results

- Performance against IFGSM (PGD) attacks.

Table 2. Quantitative comparisons of four depth estimation methods. $N(x)$ is the plain network trained only on clean images. $N_{\text{adv}}(x)$ is the same net but trained also on adversarial inputs. $G(x)$ is the saliency predictor trained only on clean images. $G_{\text{adv}}(x)$ is the same net but trained on adversarial inputs by Algorithm 1. The adversarial inputs x^*'s are generated targeting $N(x)$ by IFGSM with 10 iterations.

Attack	Prediction Method	RMSE ↓	REL↓	log 10 ↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
No attack (clean)	$N(x)$	**0.555**	**0.126**	**0.054**	**0.843**	**0.968**	**0.991**
	$N_{\text{adv}}(x)$	0.682	0.167	0.071	0.757	0.935	0.979
	$N(x \otimes G(x))$	0.683	0.154	0.068	0.773	0.939	0.982
	$N(x \otimes G_{\text{adv}}(x))$	0.615	0.148	0.063	0.792	0.952	0.987
IFGSM ($\epsilon = 0.05$)	$N(x^*)$	1.465	0.419	0.200	0.249	0.568	0.774
	$N_{\text{adv}}(x^*)$	0.666	0.160	0.067	**0.774**	0.943	0.982
	$N(x^* \otimes G(x))$	0.692	**0.156**	0.069	0.768	0.937	0.981
	$N(x^* \otimes G_{\text{adv}}(x^*))$	**0.644**	0.158	**0.067**	0.771	**0.945**	**0.984**
IFGSM ($\epsilon = 0.1$)	$N(x^*)$	1.792	0.373	0.273	0.161	0.373	0.571
	$N_{\text{adv}}(x^*)$	0.677	0.159	0.068	0.769	0.942	0.981
	$N(x^* \otimes G(x))$	0.706	**0.158**	0.070	0.763	0.934	0.980
	$N(x^* \otimes G_{\text{adv}}(x^*))$	**0.655**	0.160	**0.067**	**0.770**	**0.942**	**0.983**
IFGSM ($\epsilon = 0.15$)	$N(x^*)$	1.988	0.516	0.325	0.109	0.263	0.442
	$N_{\text{adv}}(x^*)$	0.724	0.167	0.073	0.741	0.931	0.976
	$N(x^* \otimes G(x))$	0.720	**0.159**	0.071	0.759	0.931	0.978
	$N(x^* \otimes G_{\text{adv}}(x^*))$	**0.677**	0.162	**0.068**	**0.767**	**0.939**	**0.981**
IFGSM ($\epsilon = 0.2$)	$N(x^*)$	2.107	0.541	0.360	0.075	0.201	0.370
	$N_{\text{adv}}(x^*)$	0.798	0.180	0.081	0.701	0.911	0.970
	$N(x^* \otimes G(x))$	0.743	**0.161**	0.074	0.751	0.927	0.977
	$N(x^* \otimes G_{\text{adv}}(x^*))$	**0.703**	0.165	**0.071**	**0.754**	**0.933**	**0.979**

Experimental Results

- Visualization of clean/adversarial inputs and saliency maps.

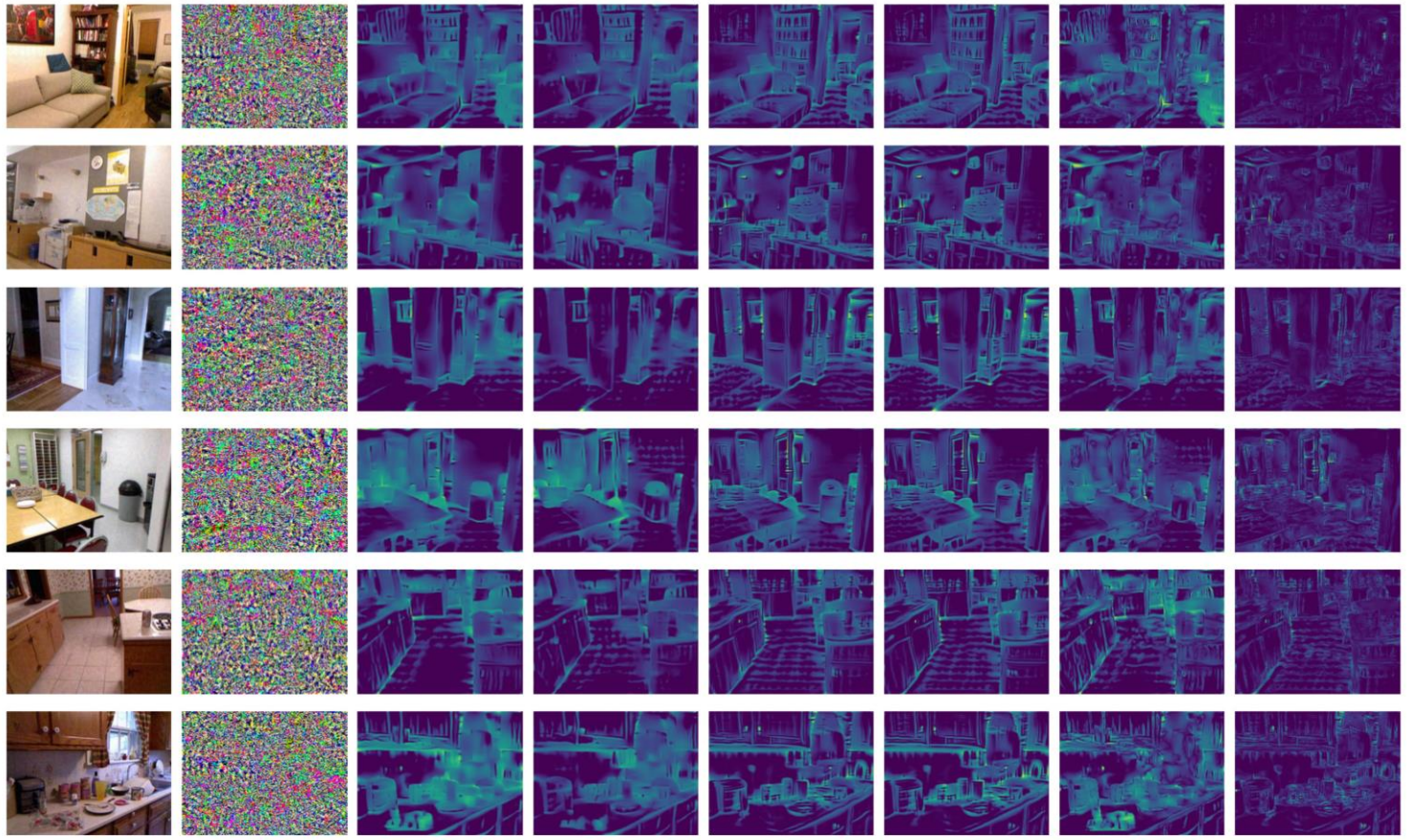


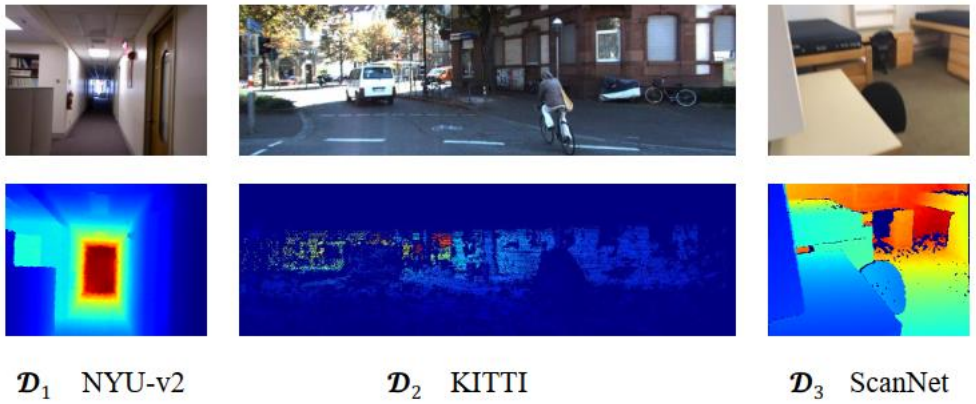
Figure 7. Visualization of clean and adversarial inputs and the saliency maps predicted from them. An identical color map created on a certain range is used for (c)-(f) and for (g) and (h), respectively.

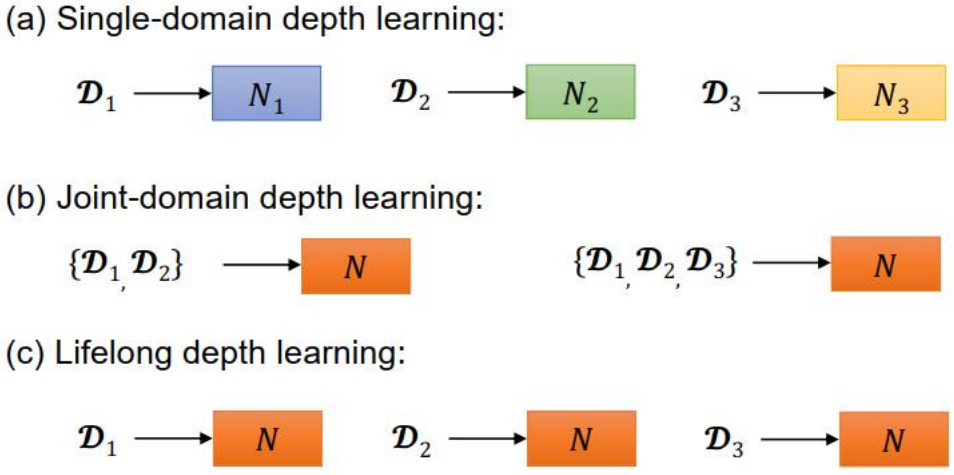
Outline

- Backgrounds
- Prediction of High-Resolution Maps
- Visualization of CNNs
- Defending Adversarial Attacks
- **Generalizability on Multiple Domains**
- Improving Computational Efficiency
- Multi-modality Data Fusion
- Conclusions

Motivation

- Visual methods for scene depth estimation suffer from low generalizability across different domains.
- Current attempts mix data from different domains and train domain-invariant model.
- However, these methods can only estimate relative depth maps without absolute scene scale.
- We propose lifelong-monodepth which takes a lifelong learning strategy and can estimate metric depth maps.





Motivation

COMPARISONS BETWEEN SEVERAL REPRESENTATIVE EXISTING WORKS AND OUR WORK.

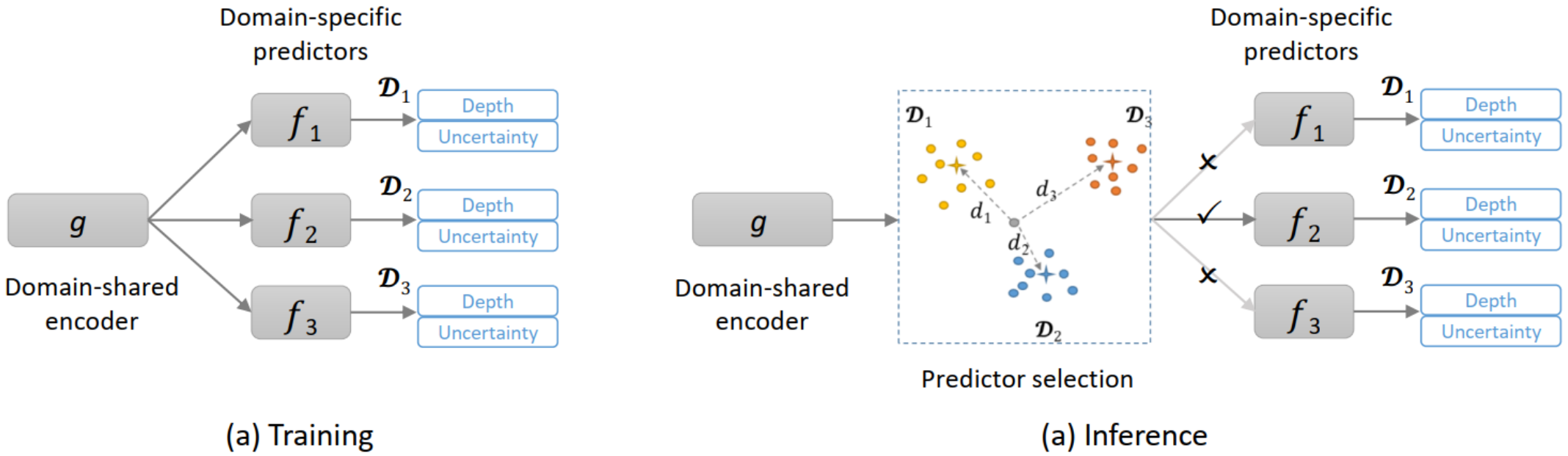
Methods	Lifelong learning	Scare-aware	Cross-domain learning strategy	(Un)supervised learning
Virtual Normal v1 [42]	✗	✓	✗	Supervised
Virtual Normal v2 [43]	✗	✗	Mixed data	Supervised
DABC [23]	✗	✓	Mixed data	Supervised
MiDas [29]	✗	✗	Mixed data	Supervised
CoSelfDepth [20]	✓	✗	Mixed data	Unsupervised
Ours	✓	✓	Sequential learning	Supervised

Challenges

- Significant domain gap: both visual images and depth images are significantly different across different domains. Thus, a trained model transfers poorly between two domains with significant differences in both visual and depth images.
- Depth scale imbalance: scene depth scales are usually domain-dependent and dominated by a specific range such that model transfer between two domains of different .scales is ineffective

Solution

- An efficient multi-head framework that enables lifelong, cross-domain, and scale-aware monocular depth learning.



Junjie Hu, Chenyou Fan, Liguang Zhou, Qing Gao, Honghai Liu, Tin Lun Lam. "Lifelong-MonoDepth: Lifelong Learning for Multi-Domain Monocular Metric Depth Estimation." arXiv preprint arXiv:2303.05050, 2023. Submitted to TNNLS.

Solution

- We propose an uncertainty-aware knowledge preservation solution by incorporating uncertainty estimation and uncertainty consistency into our lifelong learning framework.

- The former strikes a balance between different domains since the quality of depth is domain-dependent, and the latter provides a strong regularization for better preserving the model's knowledge on original domains.

Algorithm 1 Lifelong-MonoDepth: Training

Input: $\mathcal{D}^{t+1}$: new target domain;
$N^t = \{g', f'_1, ..., f'_t\}$: old model;
λ^t: weight coefficients;
$\mathcal{P} = \{\mathcal{P}_1, ..., \mathcal{P}_t\}$: replay sets;

Output: $N^{t+1} = \{g, f_1, ..., f_{t+1}\}$: new model;

Freeze N^t;
for j = 1 to *iterations* **do**
▷ % knowledge acquisition from new domain %
  Set gradients of N^{t+1} to 0;
  Select a batch (x^{t+1}, y^{t+1}) from $\mathcal{D}^{t+1}$;
  Get predictions $\hat{y}^{t=1}, s^{t+1} \leftarrow f_{t+1}(g(x^{t+1}))$;
  Compute uncertainty-aware depth loss ℓ_{ud} by Eq.(1);
  ▷ % knowledge preservation for old domains %
  for i = 1 to t **do**
    Get consistency loss ℓ_{cons} by Eq.(3);
    Select a batch (x^i, y^i) from $\mathcal{P}_i$;
    Compute replay loss ℓ_{replay} by Eq.(4);
  end for
  Get the total loss $\mathcal{L} = \ell_{ud} + \lambda^i \sum_{i=1}^{t} (\ell_{cons} + \ell_{replay})$;
  Backpropagate $\mathcal{L}$;
  Update N^{t+1};
end for

Solution

- Automatic identification of the domain-specific predictor for an input image during inference based on the minimum distance to mean features of each domain.

Algorithm 2 Lifelong-MonoDepth: Inference

Input: $N^t = \{g, f_1, ..., f_t\}$: learned model on $\mathcal{D}^1$ to $\mathcal{D}^t$;
$\mu = \{\mu_1, ..., \mu_t\}$: domain-specific mean features;
x: an image from any domain $\mathcal{D}^i, i \in \{1, ..., t\}$;

Output: $\hat{y}$: a depth map;

1: Compute intermediate features by $g(x)$;
2: **for** i = 1 to t **do**
3: Compute the distance d_i between $g(x)$ and μ_i;
4: **end for**
5: Select predictor $f_i \leftarrow \arg\min d_i$;
6: Output depth map $\hat{y} \leftarrow f_i(g(x))$;

Experimental Results

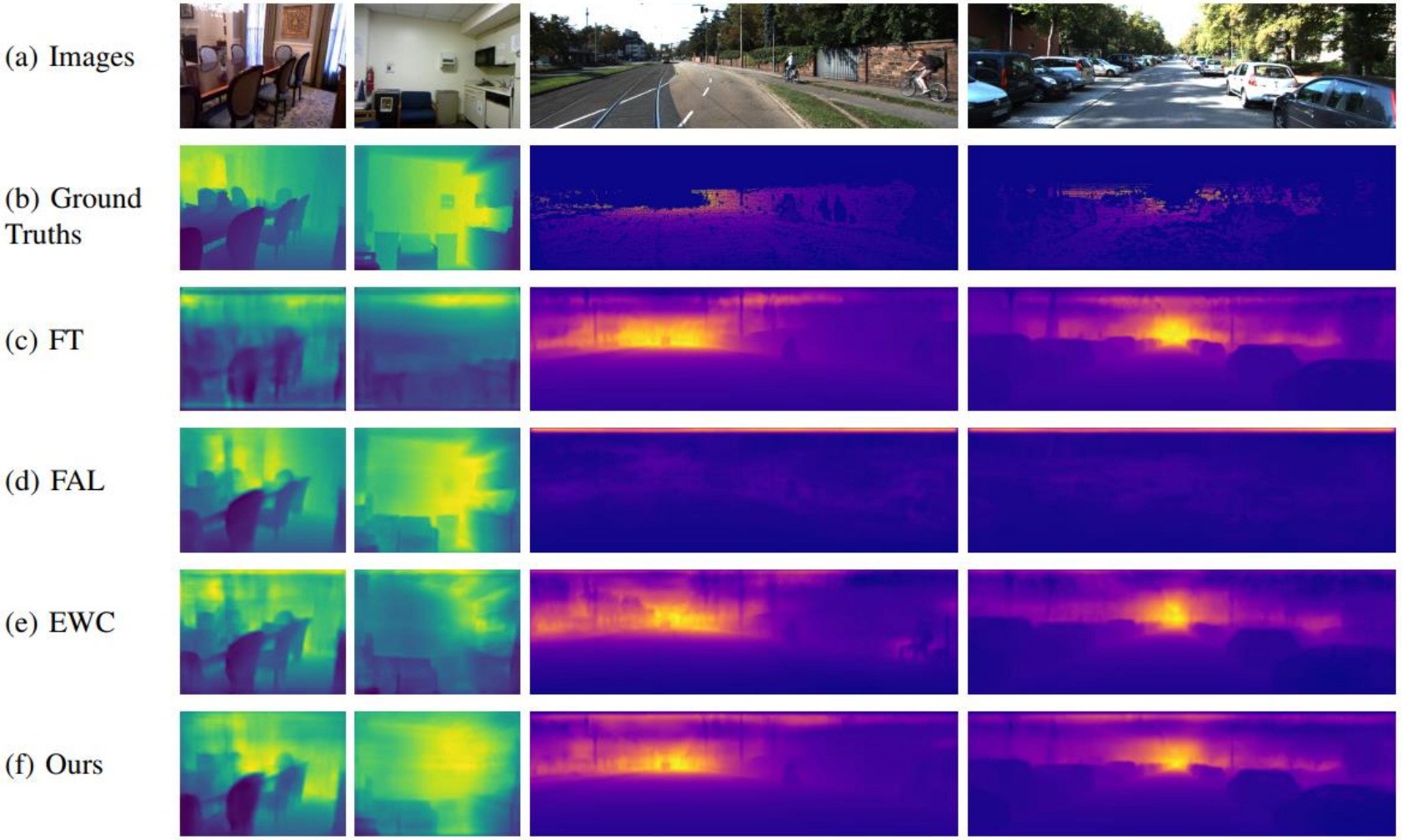
- Quantitative comparisons

QUANTITATIVE COMPARISONS BETWEEN EXISTING METHODS AND THE PROPOSED METHOD IN WHICH & DENOTES DATA MIXING AND → DENOTES SEQUENTIAL ORDER FOR LIFELONG LEARNING. NOTE THAT WE SPECIFY THE CORRECT DOMAIN-SPECIFIC PREDICTOR FOR EACH INPUT IMAGE. ∗ DENOTES RESULTS TAKEN FROM [20].

Method	NYU-v2			KITTI			Average		
	RMSE	REL	δ_1	RMSE	REL	δ_1	RMSE	REL	δ_1
SDT	0.532	0.130	0.836	3.286	0.070	0.939	1.909	0.100	0.888
JDT (NYU-v2 & KITTI)	0.581	0.151	0.803	3.658	0.086	0.911	2.120	0.119	0.857
Comoda* [22] (NYU-v2 & KITTI → KITTI)	0.673	0.191	0.706	6.249	0.158	0.769	3.461	0.175	0.738
CoSelfDepth* [20] (NYU-v2 & KITTI → KITTI)	0.626	0.187	0.728	5.809	0.154	0.784	3.218	0.171	0.756
FT (NYU-v2 → KITTI)	1.133	0.328	0.451	3.655	0.079	0.918	2.394	0.204	0.685
FAL (NYU-v2 → KITTI)	0.532	0.130	0.836	8.946	0.252	0.600	4.739	0.191	0.718
EWC (NYU-v2 → KITTI)	1.007	0.251	0.475	4.550	0.100	0.876	2.779	0.176	0.676
Ours (NYU-v2 → KITTI)	0.622	0.162	0.768	3.829	0.081	0.910	**2.226**	**0.122**	**0.839**
FT (KITTI → NYU-v2)	0.555	0.137	0.820	13.22	0.450	0.179	6.888	0.294	0.500
FAL (KITTI → NYU-v2)	0.991	0.318	0.523	3.286	0.070	0.939	2.139	0.194	0.731
EWC (KITTI → NYU-v2)	0.650	0.173	0.755	7.178	0.243	0.573	3.914	0.208	0.664
Ours (KITTI → NYU-v2)	0.567	0.142	0.812	5.060	0.136	0.813	**2.814**	**0.139**	**0.813**

Experimental Results

- Qualitative comparisons



Outline

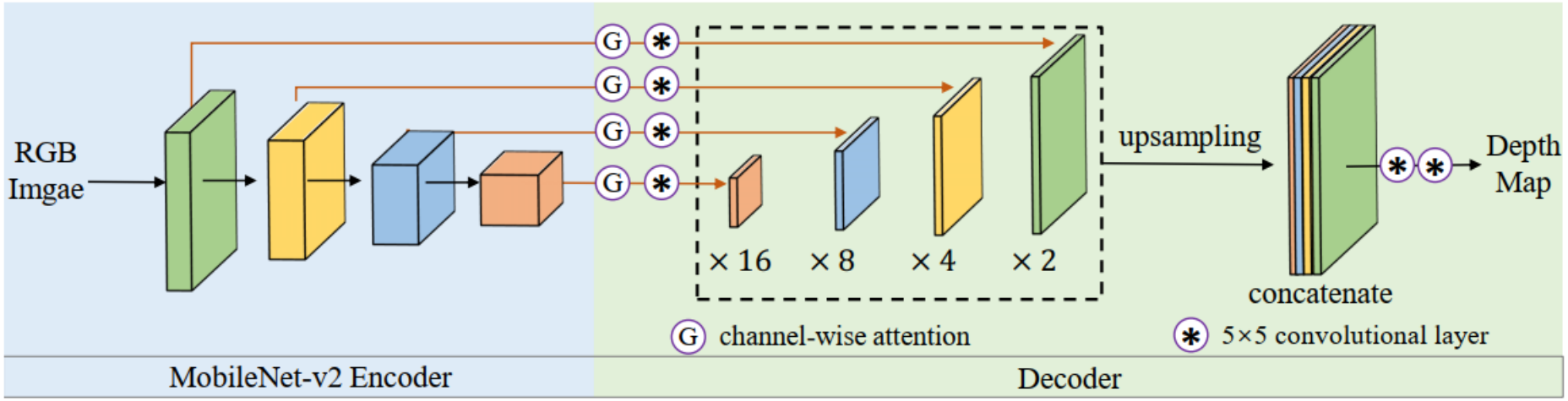
- Backgrounds
- Prediction of High-Resolution Maps
- Visualization of CNNs
- Defending Adversarial Attacks
- Generalizability on Multiple Domains
- **Improving Computational Efficiency**
- Multi-modality Data Fusion
- Conclusions

Motivation

- Many practical applications, *e.g.*, robot navigation, demand a lightweight model due to the hardware limitations and requirement for computationally efficient inference.
- We can either perform model compression on a well-trained large network or apply supervised learning to directly train a compact network.

Solution

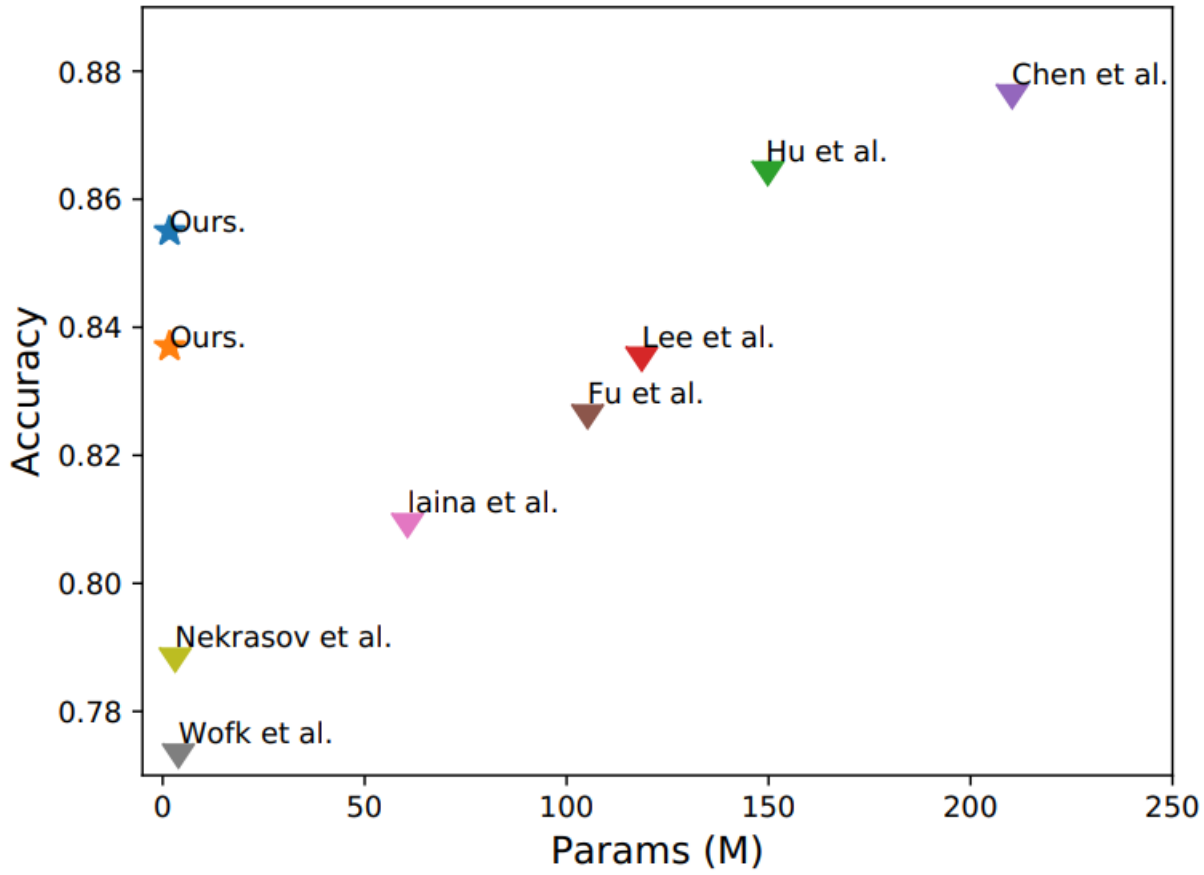
- Lightweight network for robot depth estimation.



Junjie Hu, Chenyou Fan, Hualie Jiang, Xiyue Guo, Xiangyong Lu, Tin Lun Lam. "Boosting Light-Weight Depth Estimation Via Knowledge Distillation". The 16th International Conference on Knowledge Science, Engineering and Management. (KSEM 2023).

Experimental Results

- Learning the lightweight network with knowledge distillation.



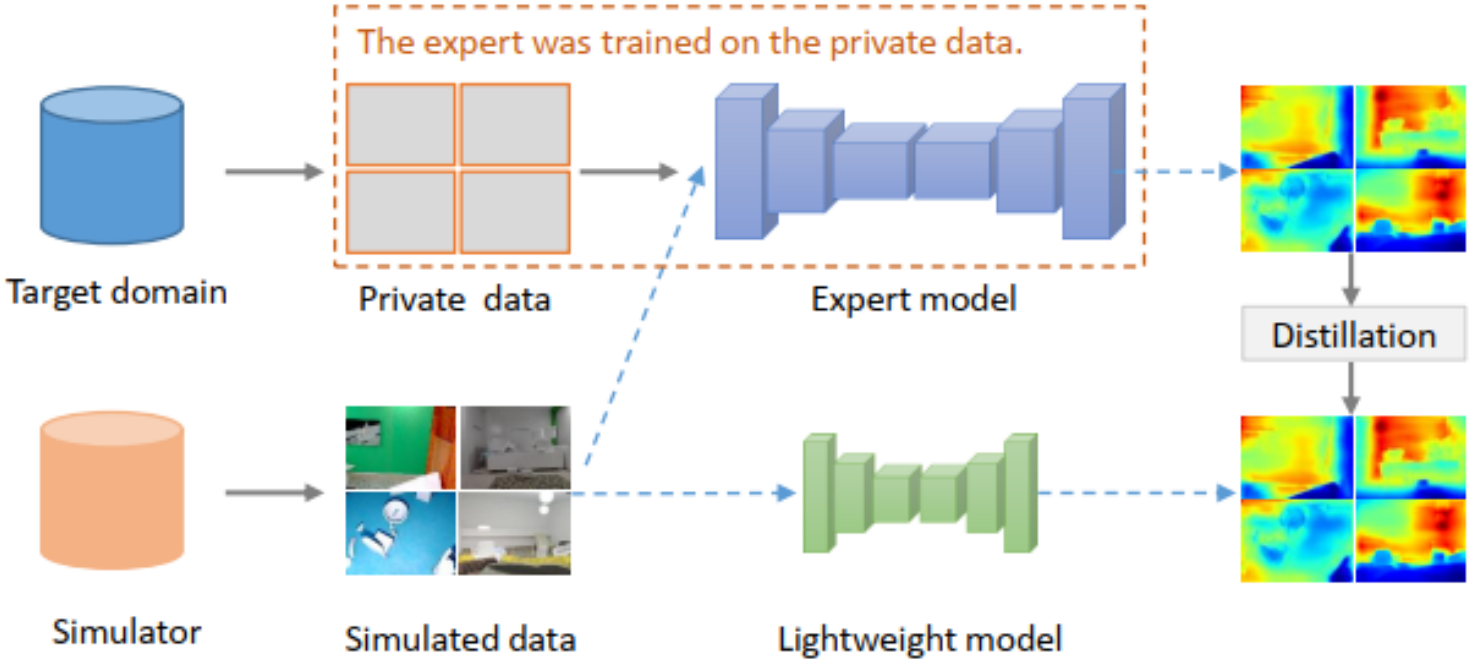
Motivation

- These solutions assume that the original training data of the target domain is known and can be freely accessed.

- However, since data privacy and security are invariably a severe concern in the real world, the training data is routinely unknown in practice, especially for industrial applications.

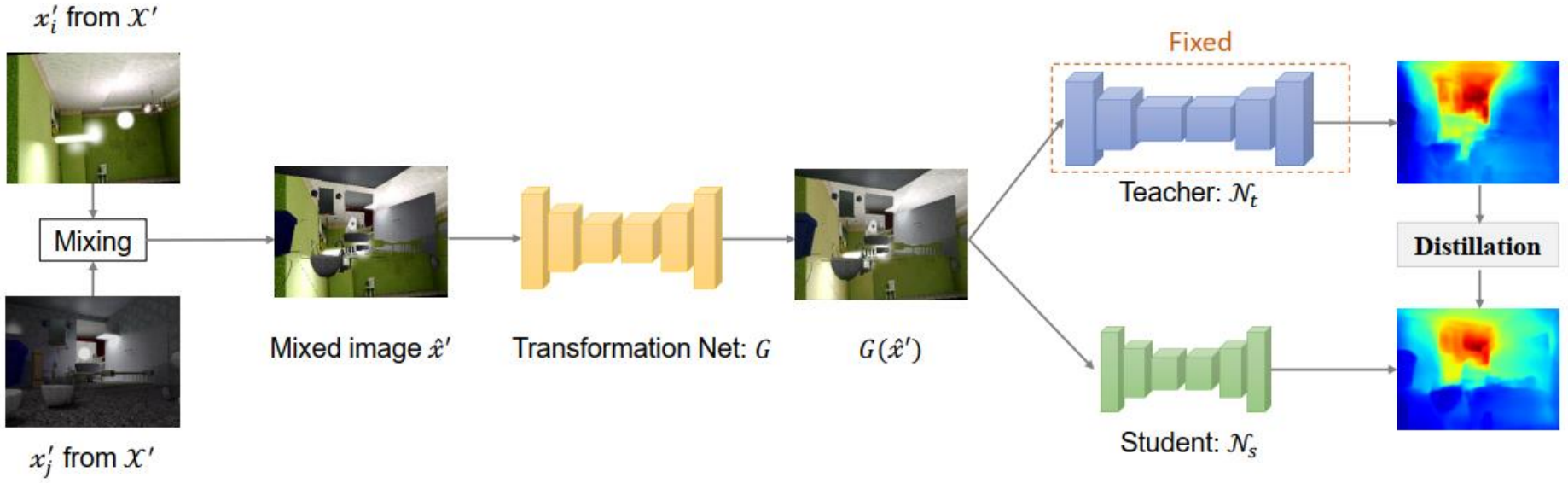
Solution

- Using RGB images collected from simulator to perform data-free knowledge distillation.
- We consider three critical elements for choosing the alternative set: i) scene similarity between the original scenarios and simulated environments, ii) the number of training images, and iii) domain gap between the real world and the simulation.



Solution

- We generate mixed images aiming to cover the distributed patterns of objects in the target domain by applying random object-wise mixing between two simulated images.
- Then, we propose to regularize the mixed images to fit the target domain by tackling an efficient image-to-feature adaption problem with a transformation network.



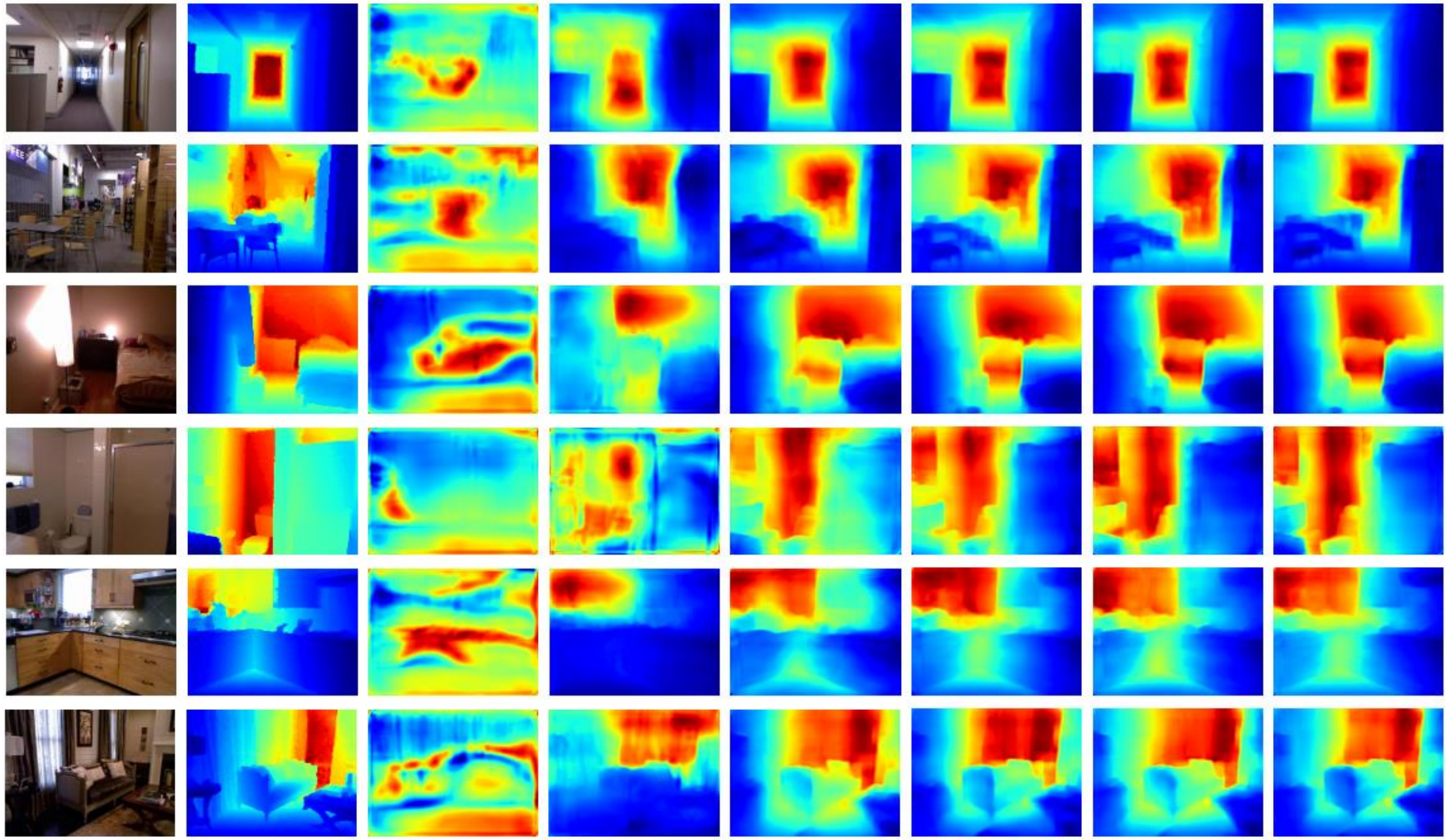
Junjie Hu, Chenyou Fan, Mete Ozay, Hualie Jiang, Tin Lun Lam. "Data-free Dense Depth Distillation." arXiv preprint arXiv:2208.12464, 2022. submitted to Neural Network

Experimental Results

Table 3
Quantitative results on the NYU-v2 dataset.

Teacher (Backbone) → Student (Backbone)		ResNet-34 → ResNet-34		ResNet-34 → MobileNet-v2		ResNet-50 → ResNet-18		ResNet-50 → ResNet-18		SeNet-154 → ResNet-34	
Parameter Reduction		None		21.9M → 1.7M		63.6M → 13.7M		67.6M → 14.9M		258.4M → 38.7M	
Method	Data	REL	δ_1	REL	δ_1	REL	δ_1	REL	δ_1	REL	δ_1
Teacher	NYU-v2	0.133	0.829	0.133	0.829	0.134	0.824	0.126	0.843	0.111	0.878
Student		0.133	0.829	0.145	0.802	0.145	0.805	0.137	0.826	0.125	0.843
Random noises	None	0.426	0.193	0.431	0.194	0.517	0.102	0.511	0.112	0.514	0.107
DFAD		0.285	0.402	0.306	0.329	0.300	0.382	0.341	0.338	0.347	0.278
KD-OOD	SceneNet $\mathcal{X}_1'$	0.164	0.753	0.175	0.712	0.188	0.660	0.175	0.710	0.174	0.695
Ours		**0.155**	**0.774**	**0.168**	**0.742**	**0.173**	**0.701**	**0.167**	**0.722**	**0.156**	**0.759**
KD-OOD	SceneNet $\mathcal{X}_2'$	0.158	0.761	0.165	0.742	0.180	0.676	0.172	0.713	0.161	0.738
Ours		**0.151**	**0.789**	**0.157**	**0.778**	**0.165**	**0.726**	**0.157**	**0.760**	**0.151**	**0.776**

Experimental Results



(a) RGB images. (b) Ground truth. (c) Random noise. (d) DFAD. (e) KD-OOD with $\mathcal{X}'_1$. (f) Our results with $\mathcal{X}'_1$. (g) KD-OOD with $\mathcal{X}'_2$. (h) Our results with $\mathcal{X}'_2$.

Outline

- Backgrounds
- Prediction of High-Resolution Maps
- Visualization of CNNs
- Defending Adversarial Attacks
- Towards High Generalizability
- Improving Computational Efficiency
- **Multi-modality Data Fusion**
- Conclusions

Motivation

- Visual methods often yield a low inference accuracy and poor generalizability and thus are vulnerable to real-world deployment.

- RMSE is around 4 meters for visual methods and 0.7 meters for multi-modality data fusion methods on the KITTI dataset.

- A practical way is to predict scene depth from both RGB images and sparse depth maps (depth sensor).

Motivation

- Many methods takes multi-modality/multi-sensor data fusion for estimating accurate depth maps.

- However, these methods require 64-line LiDARs which are expensive. For example, the well-known Velodyne HDL-64E LiDAR is around 75,000 US dollars.

- We aim to enable accurate depth prediction from only 1-line LiDARs.

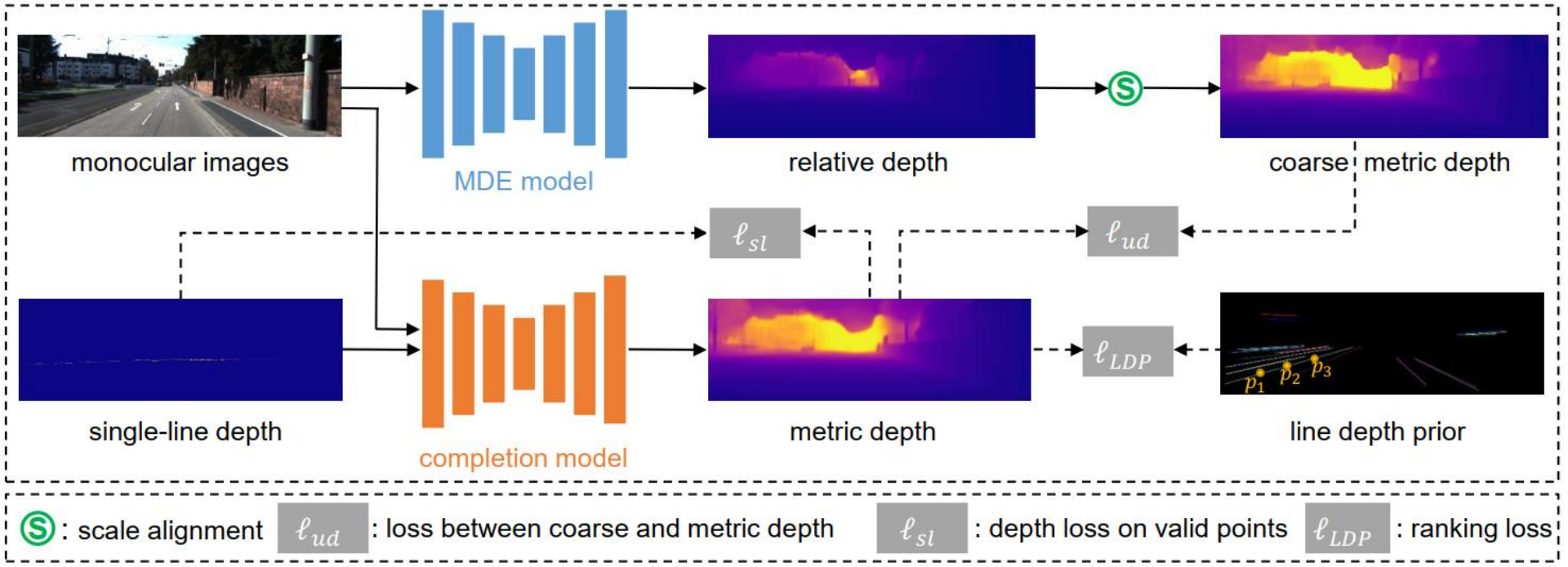
Related work

Main categories	Sub-categories	Major characteristics
Unguided methods (Sec. 3)	Sparsity-aware CNNs (**SACNN**, Sec. 3.1)	Using the binary validity mask to indicate missing elements during convolution.
	Normalized CNNs (**NCNN**, Sec. 3.2)	1). Built on normalized convolution 2). Replacing the validity mask with continuous confidence mask.
	Training with Auxiliary Images (**TwAI**, Sec. 3.3)	Integrating image reconstruction into latent or output space to encourage learning semantic cues. Image guided training and unguided inference are employed.
RGB guided methods (Sec. 4)	Early fusion models (**EFM**, Sec. 4.1) • Encoder-decoder networks (**EDN**, Sec. 4.1.1) • Coarse to refinement prediction (**C2RP**, Sec. 4.1.2)	Directly aggregating the image and sparse depth map input or fusing the multi-modality features at the first convolutional layer.
	Late fusion models (**LFM**, Sec. 4.2) • Dual-encoder networks (**DEN**, Sec. 4.2.1) • Double encoder-decoder networks (**DEDN**, Sec. 4.2.2) • Global and Local Depth Prediction (**GLDP**, Sec. 4.2.3)	The framework usually consists of dual encoders or two sub-networks; the one is used for extracting RGB features and the other is used for extracting depth features. Fusion is conducted at the intermediate layers, e.g., fusing extracted features from encoders.
	Explicit 3D representation models (**E3DR**, Sec. 4.3) • 3D-aware convolution (**3DAC**, Sec. 4.3.1) • Intermediate surface normal representation (**ISNR**, Sec. 4.3.2) • Learning from point clouds (**LfPC**, Sec. 4.3.3)	Explicitly learning 3D representations, such as applying 3D convolutions, embedding surface normals, and learning from 3D point clouds.
	Residual depth models (**RDM**, Sec. 4.4)	Learning a coarse depth map and a residual depth map. Their combination generates the final depth map.
	SPN-based models (**SPM**, Sec. 4.5)	1). Based on the spatial propagation network. 2). First learning the affinity matrix, and then applying affinity based depth refinement.

Junjie Hu, Chenyu Bao, Mete Ozay, Chenyou Fan, Qing Gao, Honghia, Liu and Tin Lun Lam. "Deep Depth Completion from Extremely Sparse Data." IEEE Transactions on Pattern Analysis and Machine Intelligence. (TPAMI 2022).

Solution

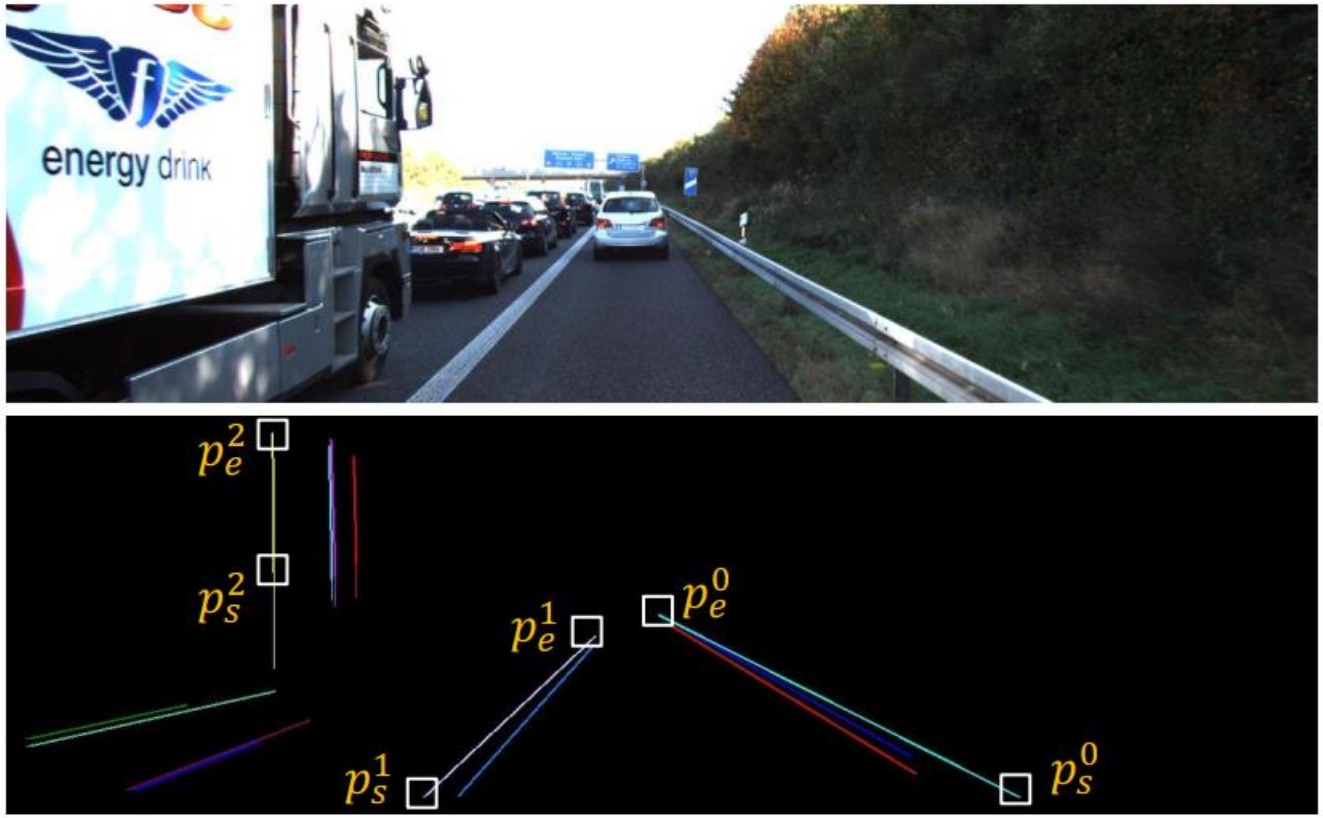
- We propose the Relative-to-Metric (R2M) depth distillation method that allows training a depth completion model by distilling from a monocular depth estimator.



Junjie Hu, Chenyou Fan, Xiyue Guo, Liguang Zhou and Tin Lun Lam. " Self-supervised Single-line LiDAR Depth Completion“, RAL 2023 submission.

Solution

- Line Depth Prior (LDP): we observed that points on straight lines naturally form a prior regarding relative depths. To enforce the consistency of relative depths among pixels on lines, we propose a ranking loss based on LDP.



Solution

- The final loss function measures
- the dissimilarity between our completed depth map and the MDE model's output,
- a depth consistency loss that enforces agreement between predicted and observed depths on valid points in single-line depth maps,
- and a ranking loss that encourages the correct depth relations of points on straight lines.

$$\mathcal{L} = \ell_{ud} + \alpha \ell_{sl} + \beta \ell_{LDP}.$$

Experimental Results

TABLE I

COMPARISONS OF DIFFERENT METHODS OF SINGLE-LINE LIDAR COMPLETION ON THE KITTI VALIDATION DATASET. THE METHOD OF LU ET AL. [27] WAS ORIGINALLY EVALUATED ON A DIFFERENT TEST SPLIT; WE MARK IT WITH *.

Method	Self-supervised	Scale-aware	RMSE ↓	REL ↓	log 10 ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
Balanced DC (MDE) [49]	✗	✓	3.951	-	-	-	-	-
Balanced DC [49]	✗	✓	3.921	-	-	-	-	-
Ryu et al. (MDE) [34]	✗	✓	3.625	-	-	-	-	-
Ryu et al. [34]	✗	✓	3.616	-	-	-	-	-
Lu et al.* [27]	✓	✓	4.582	0.106	-	0.871	0.951	0.982
MDE model: MonoDepth2 [7]	✓	✗	4.198	0.134	0.054	0.854	0.974	0.993
MDE model: CADepth [48]	✓	✗	3.914	0.120	0.048	0.877	0.977	0.994
Ours (MonoDepth2 → S2D)	✓	✓	3.700	0.100	0.041	0.920	0.983	0.995
Ours (MonoDepth2 → DCVAN)	✓	✓	3.723	0.098	0.040	0.920	0.983	0.995
Ours (CADepth → S2D)	✓	✓	**3.404**	**0.088**	**0.036**	**0.933**	**0.987**	**0.996**

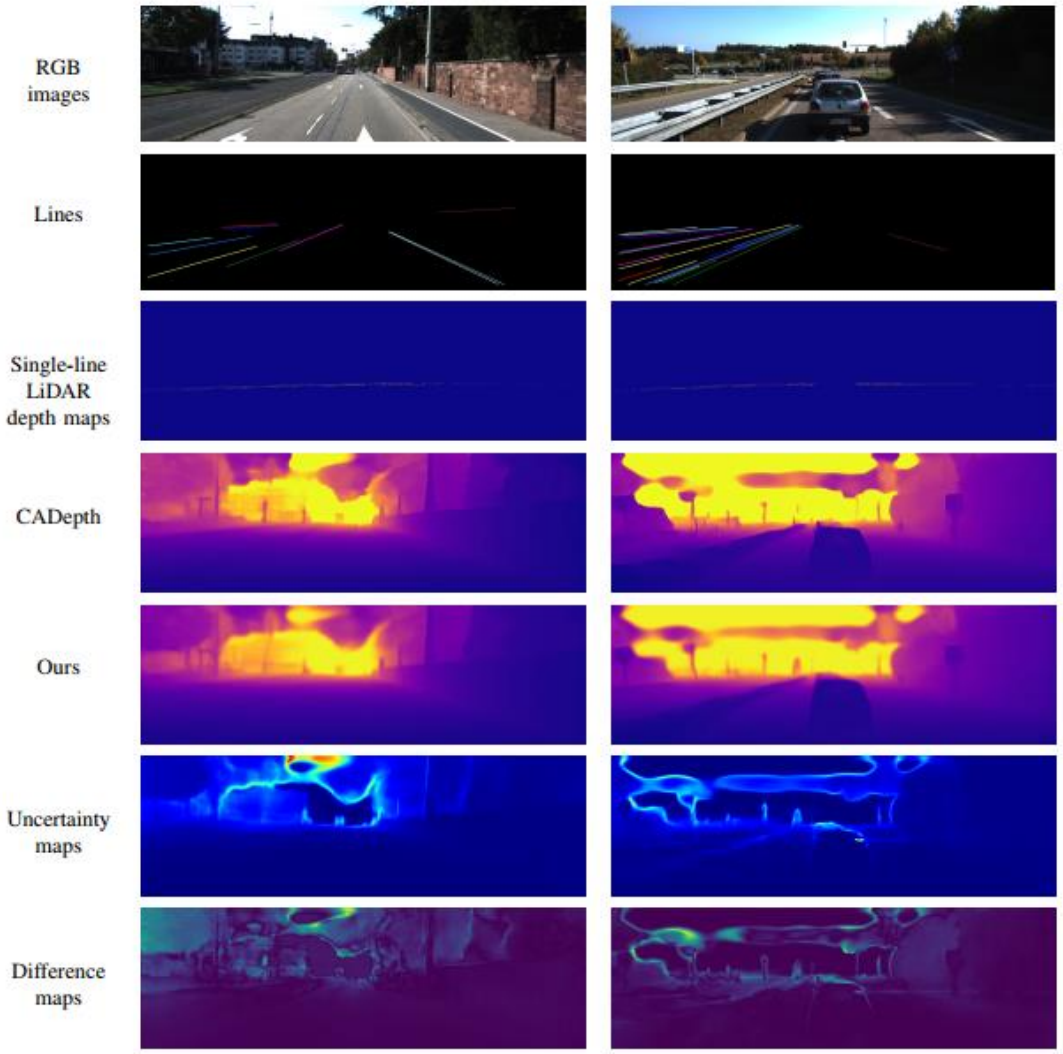
Experimental Results

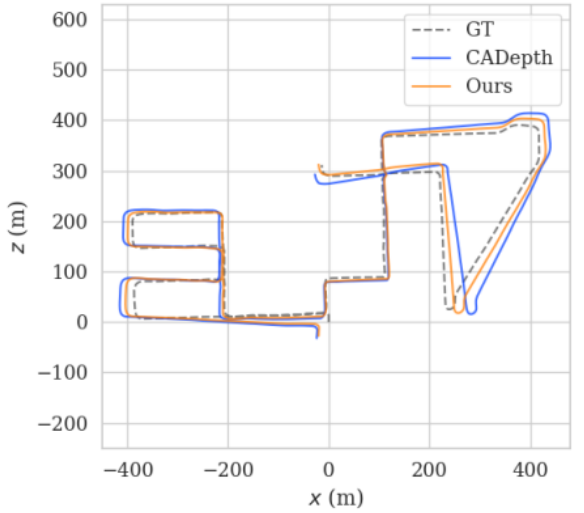
- Ablation study on loss functions

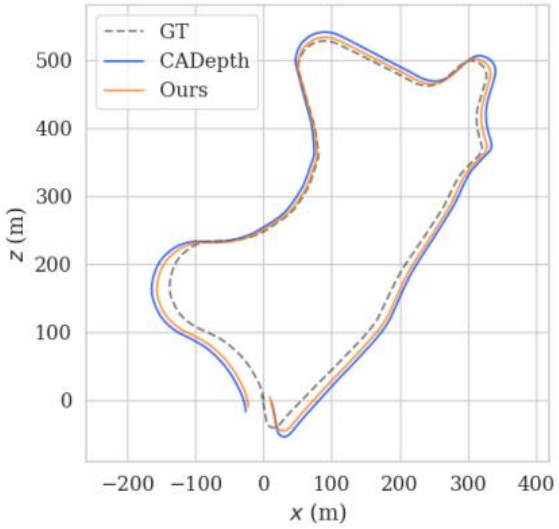
TABLE II
ABLATION STUDIES ON EACH LOSS TERM. WE SHOW RMSE RESULTS FOR COMPARISON.

	MonoDepth2 → S2D	MonoDepth2 → DCVAN	CADepth → S2D
ℓ_{ud}	3.809	3.890	3.538
$\ell_{ud} + \alpha\ell_{sl}$	3.739	3.886	3.461
$\ell_{ud} + \alpha\ell_{sl} + \beta\ell_{LDP}$	**3.700**	**3.723**	**3.404**

Experimental Results







Conclusions

- Visual methods are only accurate on fixed domains. They suffer from:
  - Low perceptual quality.
  - Low Interpretability
  - Vulnerability
  - Low generalizability
  - Time-consuming
  - Cannot infer metric depth in practice
- We provide comprehensive reviews of the above problems and introduce effective solutions.