Monocular Depth Estimation

Authors

Mahdi Abdollah Chalaki

Masoud Jafaripour

UNIVERSITY OF ALBERTA

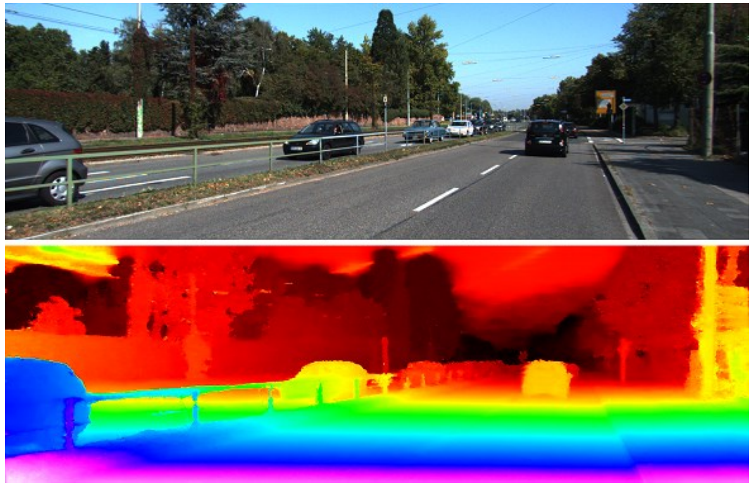
Outline

- Problem Statement
- Applications
- Methods
- Deep Learning
- Baseline Model
- Improved Models
- Comparing Methods
- Implementation

Problem Statement

- Depth estimation (DE):
  - Prediction of depth of a scene using 2D images
  - Easy task for humans and difficult for computational models
  - **Input**: 2D RGB pictures
  - **Output**: depth map representing the distance between the object and the camera viewpoint

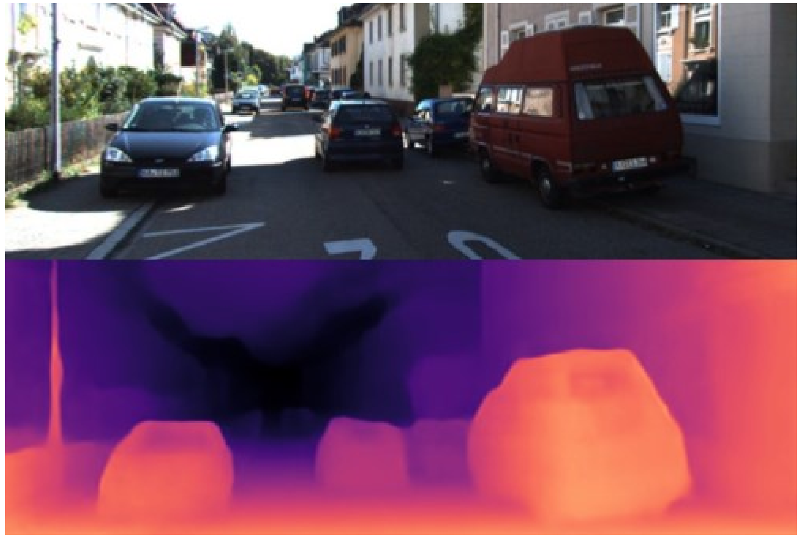
- Monocular Depth estimation (MDE):
  - Estimating depth from a single RGB image



Source: Smolyanskiy et al (2018 CVPR)

Applications

Navigation

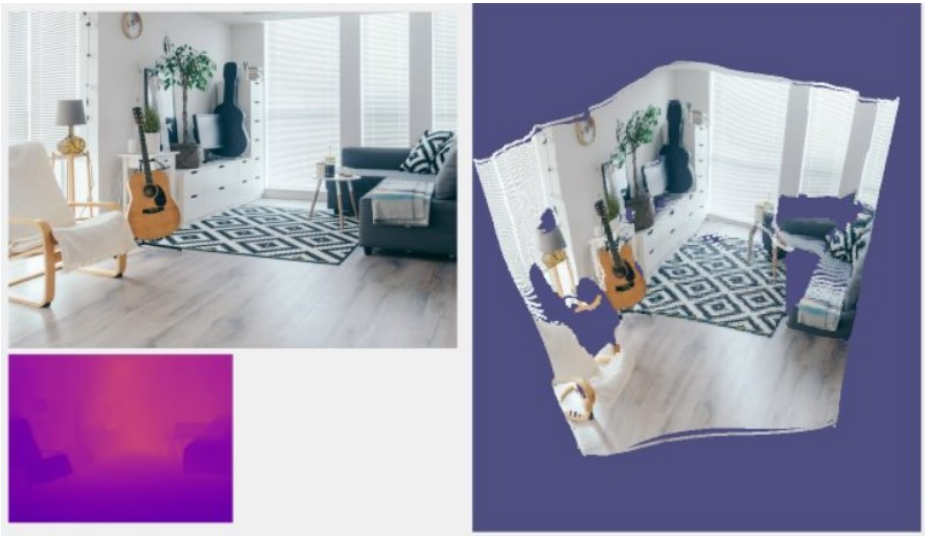


Source: github.com/nianticlabs/monodepth2

Augmented Reality

Source: thegamer.com/improved-depth-estimation-in-augmented-reality-is-here/

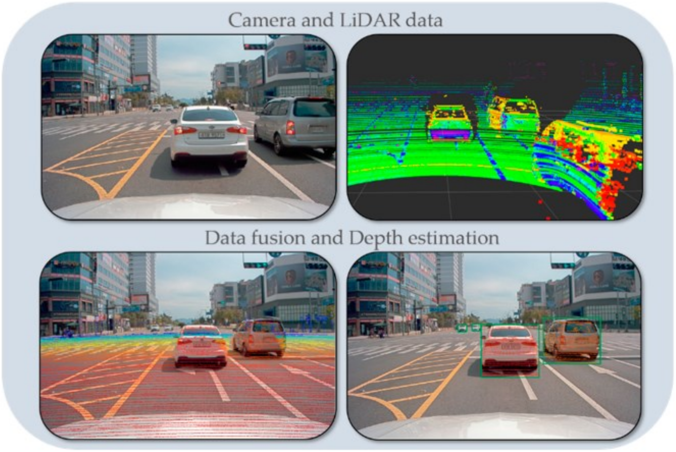
3D Scene Reconstruction



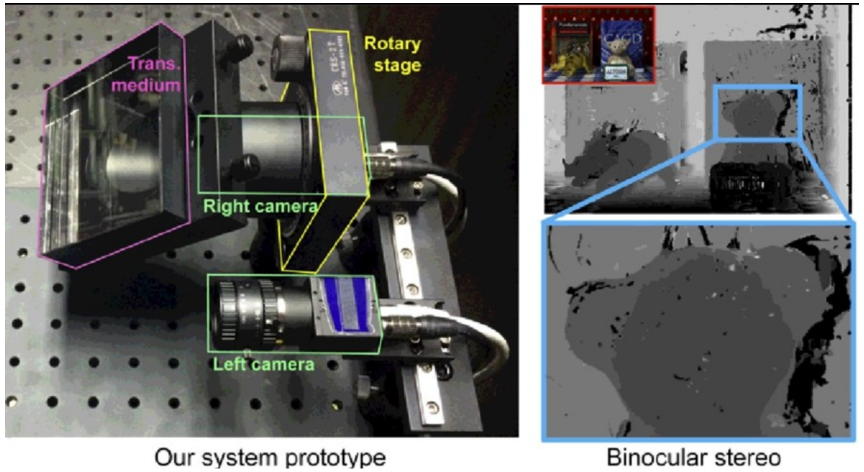
Source: github.com/ialhashim/DenseDepth

Methods

- Binocular/Multi-View DE
- Using Depth Sensors for DE (**active**)
  - LiDAR
  - RGB-D cameras
- Classical Methods for Stereo DE (**passive**)
  - Focus/defocus
  - SIFT
- Monocular Depth Estimation:
  - Benefits
  - Challenges



Source: Kumar et al. (Symmetry 2020)

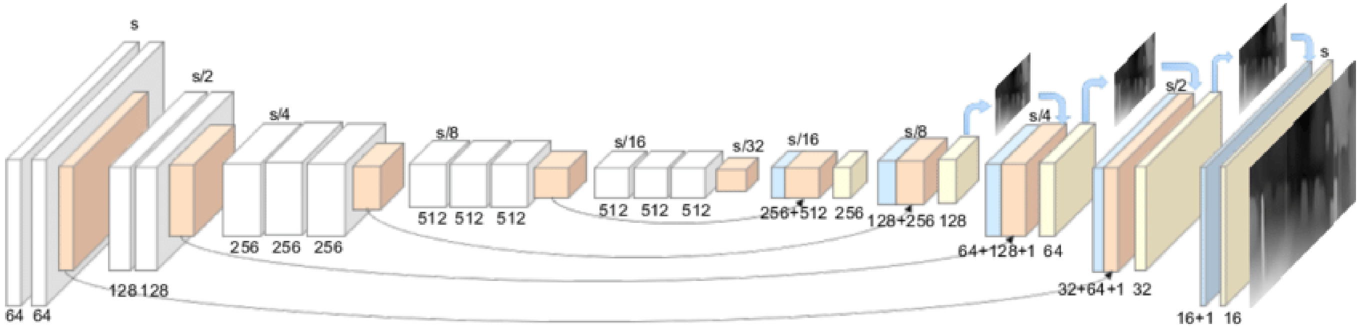


Source: Baek et al. Computer Vision and Image Understanding (May 2016)

UNIVERSITY OF ALBERTA

Deep Learning

- Our goal
- Supervised learning
- Convolutional Neural Networks (CNNs)
- Common model architecture:



Source: Guo et al. arXiv:1808.06586v1

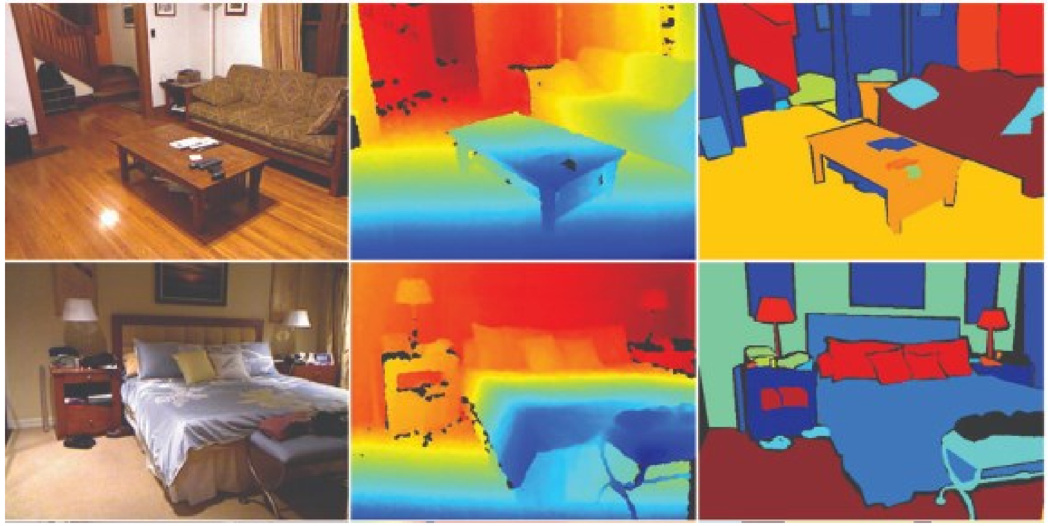
Let’s get to training!

UNIVERSITY OF ALBERTA

Datasets

NYUv2

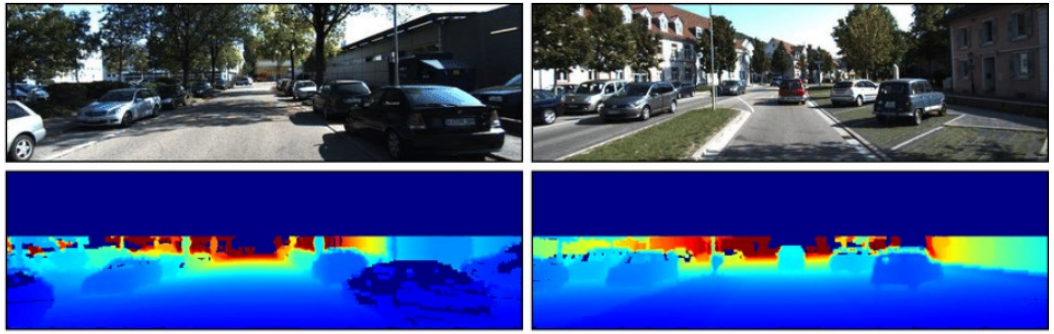
- Pictures of indoor scenes captured by Microsoft Kinect
- A resolution of 640*480
- Contains 120K training samples and 654 testing samples
- The depth maps have an upper bound of 10 meters



Source: Silberman et al. (ECCV12)

KITTI

- 44K stereo images and corresponding 3D laser scans of outdoor scenes
- captured using equipment mounted on a moving vehicle
- A resolution of around 1241*376
- corresponding depth maps have lots of missing data



Source: Geiger et al. (2012CVPR)

UNIVERSITY OF ALBERTA

Evaluation

Metrics

[Eigen et al. (NIPS 2014)]

- Average relative error (rel): $$\frac{1}{n}\sum_{p}^{n}\frac{\left|y_p - \hat{y}_p\right|}{y}$$
- Root mean squared error (rms): $$\sqrt{\frac{1}{n}\sum_{p}^{n}\left(y_p - \hat{y}_p\right)^2)}$$
- Average ($\log_{10}$) error: $$\frac{1}{n}\sum_{p}^{n}\left|\log_{10}(y_p) - \log_{10}(\hat{y}_p)\right|$$
- Threshold accuracy (δ_i): $$\%\text{ of } y_p \text{ s.t. } \max\left(\frac{y_p}{\hat{y}_p}, \frac{\hat{y}_p}{y_p}\right) = \delta < thr \text{ for } thr = 1.25, 1.25^2, 1.25^3$$

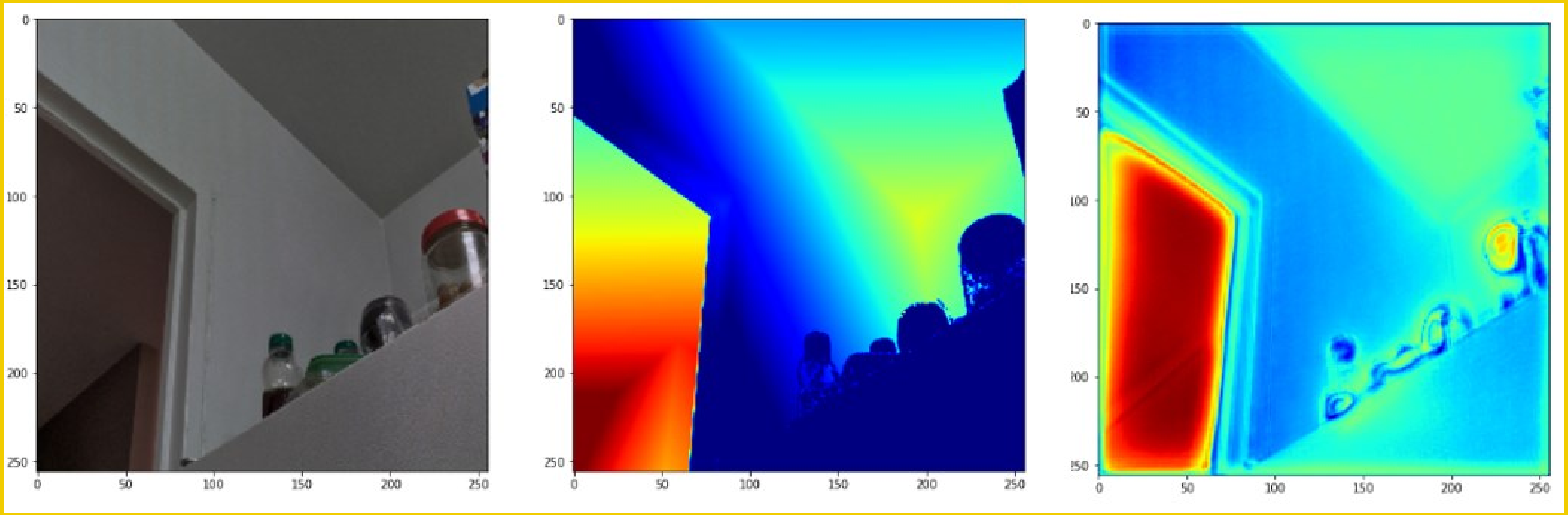
$\boldsymbol{y_p}$: a pixel in depth image y
$\widehat{\boldsymbol{y_p}}$: a pixel in the predicted depth image $\hat{y}$
n: total number of pixels for each depth image

Baseline Model

Model Architecture:

- Encoder – Decoder model : 3 Conv layers & 3 Deconv layers
- Loss function: Structural similarity index(SSIM) + L1-loss
- Input size: 256*256
- Output size: 256*256
- Batch size: 32
- Epochs: 30
- Optimizer: Adam (learning rate = 0.0002)
- Activation function: ReLU function
- Trained on the NYU Depth V2 dataset

Results:



Input image

Ground truth depth map

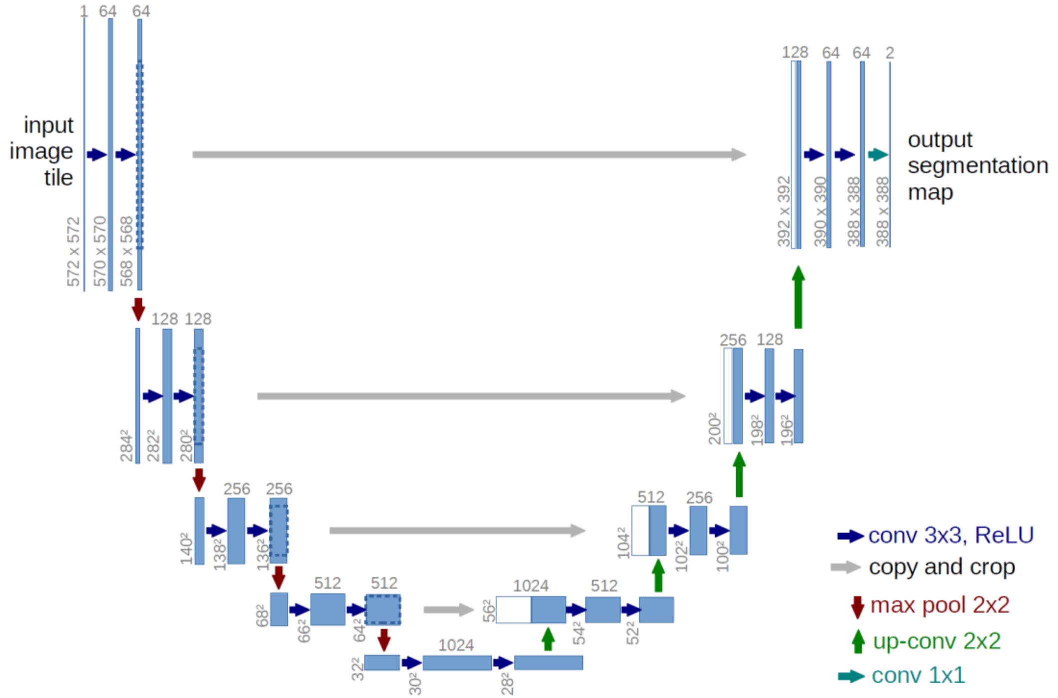
Predicted depth map

We should use improved models!

Using Transfer learning

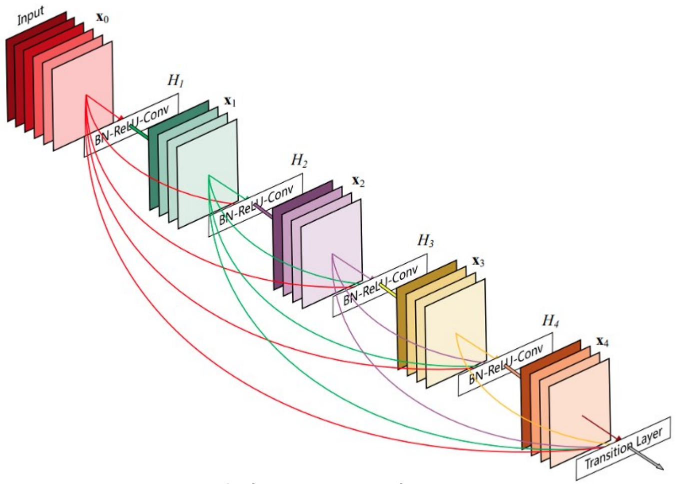
U-net

- Convolutional Networks for Biomedical Image Segmentation
- Skip connections
- Architecture



Source: Ronneberger et al. (MICCAI 2015)

Dense-Net 169

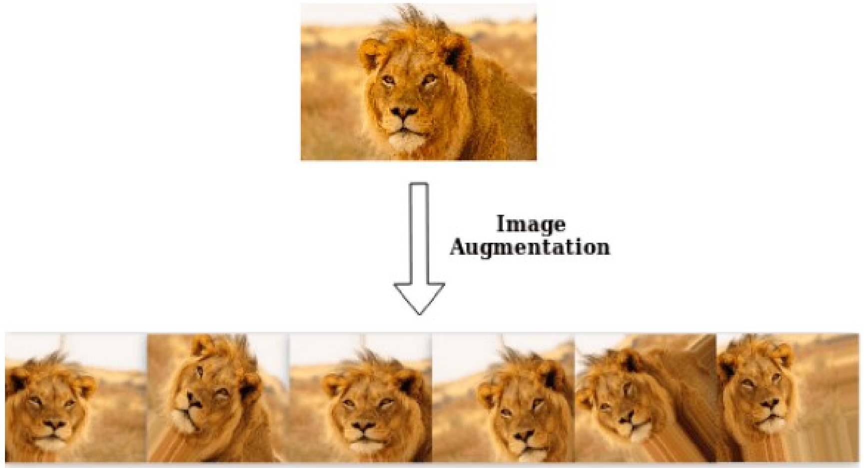


Source: Huang et al. (CoRR 2016)

UNIVERSITY OF ALBERTA

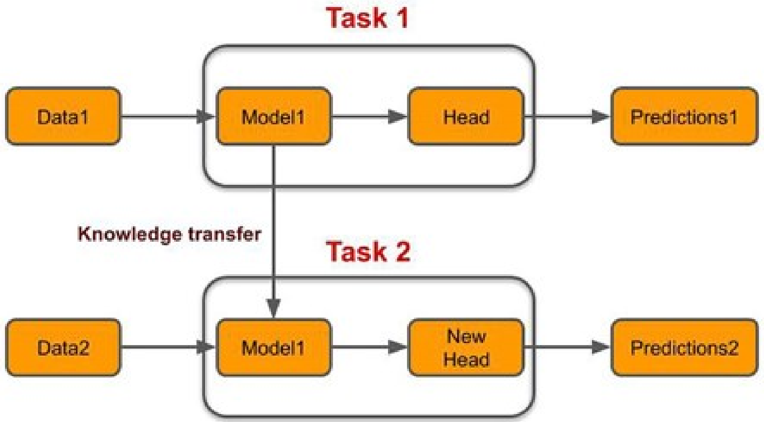
Using Transfer learning

Data augmentation

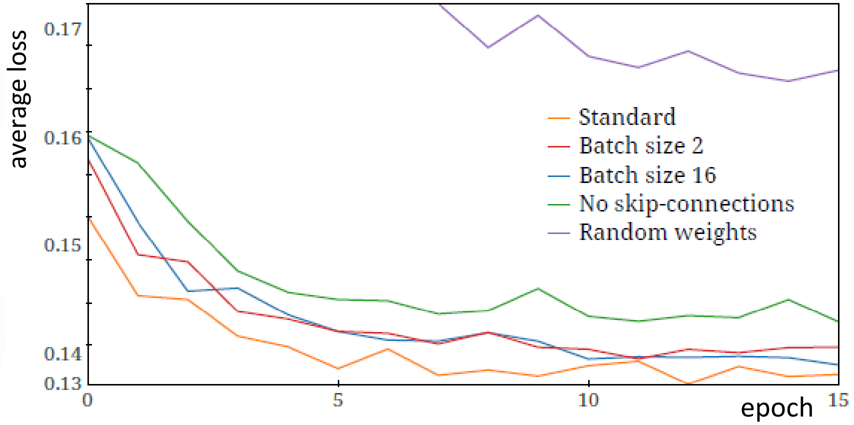


Source: towardsdatascience.com/machinex-image
-data-augmentation-using-keras-b459ef87cd22

Transfer learning



Source: topbots.com/transfer-learning-in-nlp/



Source: Alhashim et al. arXiv:1812.11941v2

Loss function:
Previous one +

$$L_{\text{grad}}(y, \hat{y}) = \frac{1}{n} \sum_{p}^{n} \left| g_{\mathbf{x}}(y_p, \hat{y}_p) \right| + \left| g_{\mathbf{y}}(y_p, \hat{y}_p) \right|$$

Results

Method	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	rel$\downarrow$	sq. rel$\downarrow$	rms$\downarrow$	$log_{10} \downarrow$
Using transfer learning	0.886	0.965	0.986	0.093	0.589	4.170	0.171

Source: Alhashim et al. arXiv:1812.11941v2

UNIVERSITY
OF ALBERTA

Results

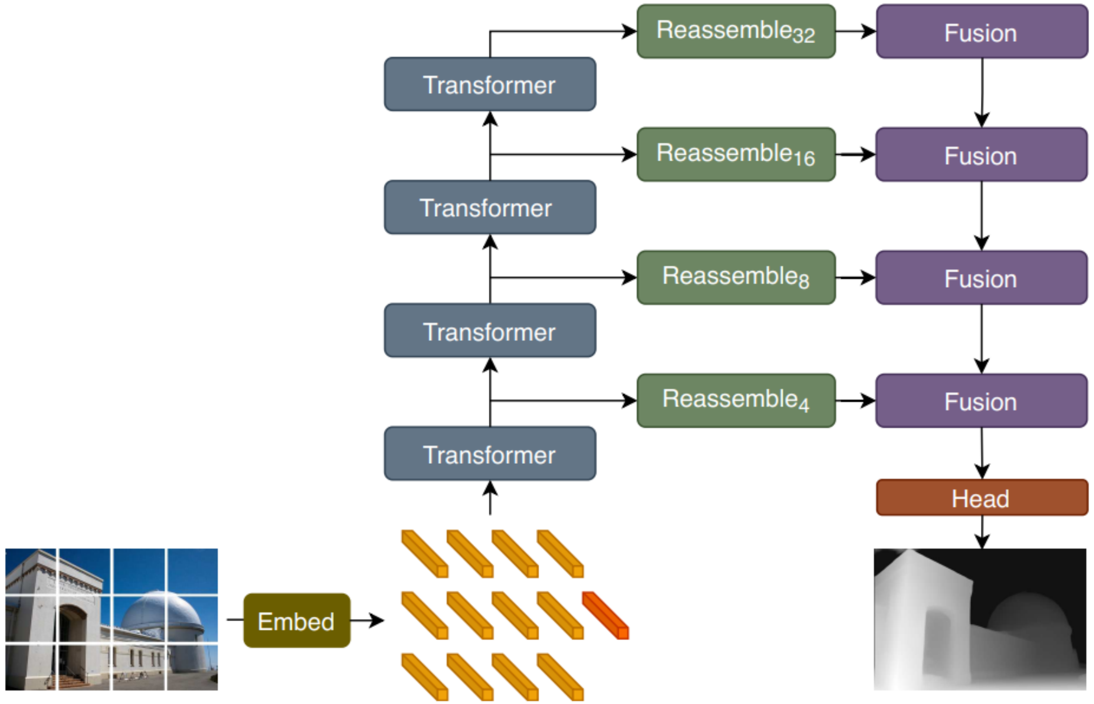
Input image

Predicted
depth map

Vision Transformers for Dense Prediction

- **Goal:** Using vision transformers as a backbone on encoder-decoder design

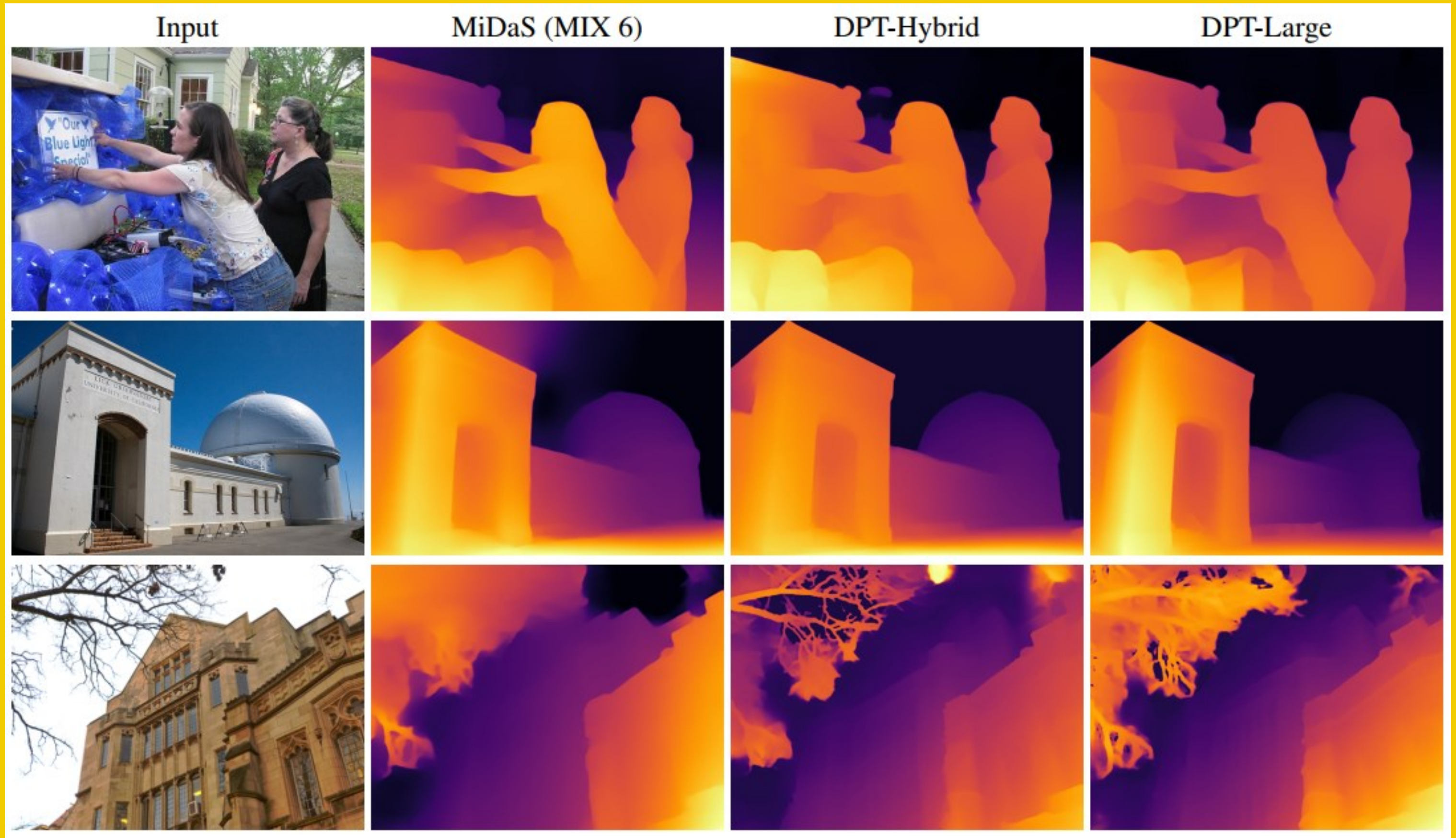
- **Network Architecture:**
  - Transformer encoder
  - Convolutional decoder
    - (Reassemble & Fusion)
  - **Loss Functions:**
    - scale- & shift-invariant trimmed loss together with the gradient-matching loss



Source: Ranftl et al. (2021ICCV)

Results:

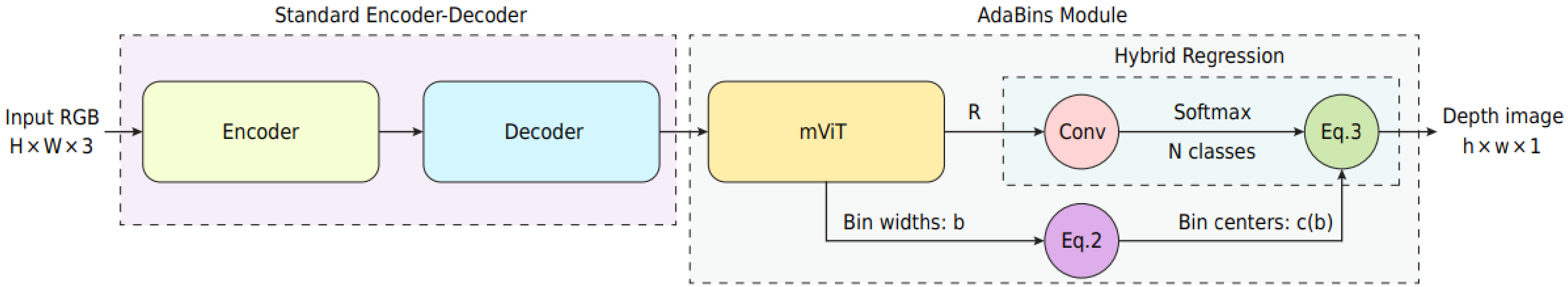
- Comparison to the state-of-the-art model



Source: Ranftl et al. (2021ICCV)

AdaBins: Depth Estimation using Adaptive Bins

- **Goal:** Transforming MDE to a classification task by discretizing depth range using bins which changing adaptively
- **Network Architecture**



Source: Bhat et al. (2021CVPR)

- **Loss Function:** Scale-Invariant loss (SI) & bi-directional Chamfer Loss

$$\mathcal{L}_{pixel} = \alpha\sqrt{\frac{1}{T}\sum_i g_i^2 - \frac{\lambda}{T^2}(\sum_i g_i)^2} \qquad g_i = \log \tilde{d}_i - \log d_i$$

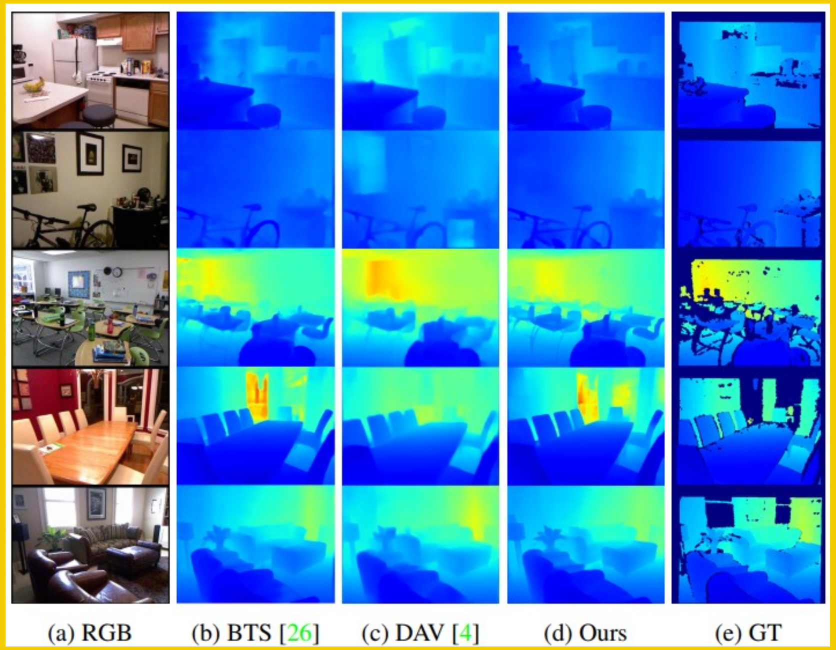
$$\mathcal{L}_{bins} = chamfer(X, c(\mathbf{b})) + chamfer(c(\mathbf{b}), X)$$

$$\mathcal{L}_{total} = \mathcal{L}_{pixel} + \beta\mathcal{L}_{bins}$$

UNIVERSITY OF ALBERTA

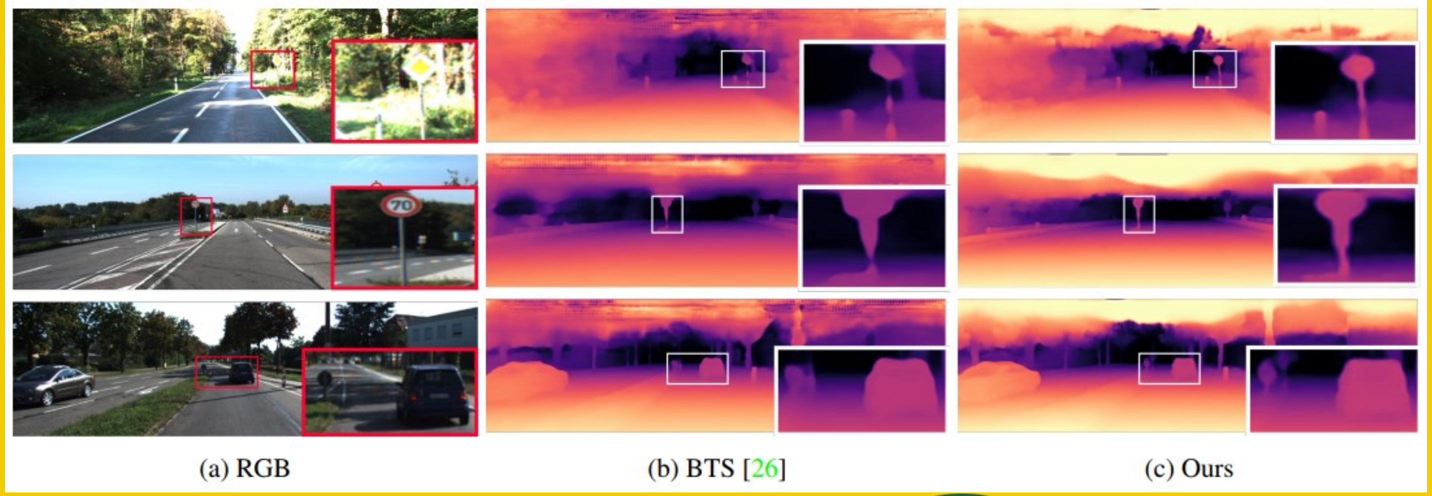
Results:

- In comparison to the state-of-the-art methods outperform them in all metrics
  - On **NYU-Depth-v2** dataset



Source: Bhat et al. (2021CVPR)

  - On **KITTI** dataset



Source: Bhat et al. (2021CVPR)

Comparing Methods:

- Comparison of performances on the **NYU-Depth-v2** dataset

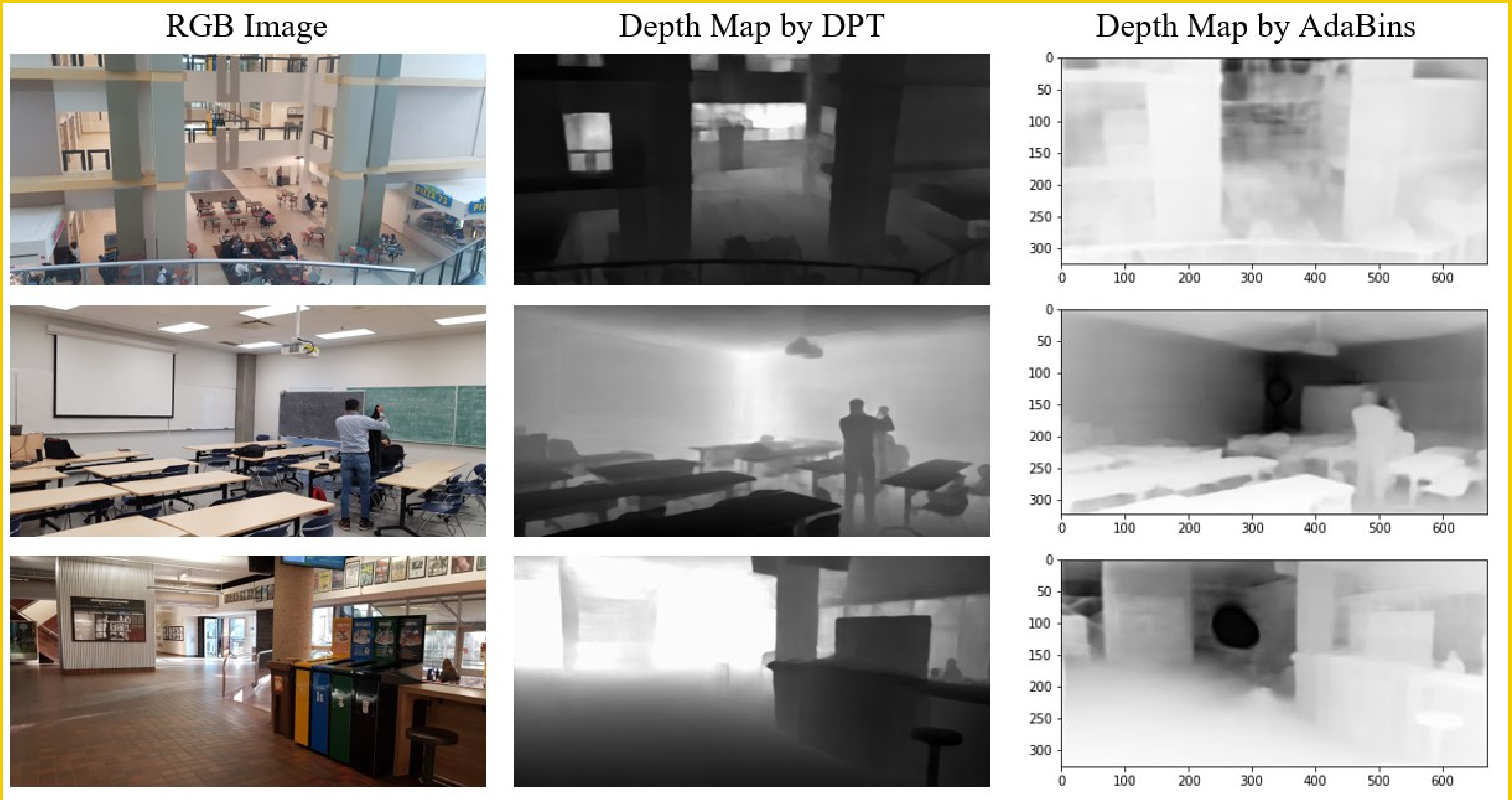
Method	δ_1	δ_2	δ_3	REL	RMS
BTS (*Lee et all, arXiv preprint 2019*)	0.885	0.978	0.994	0.110	0.392
Transfer Learning (Dense-Net 169)	**0.846**	**0.974**	**0.994**	0.123	0.465
Dense Prediction Transformer (DPT)	0.904	0.988	0.998	0.110	**0.357**
AdaBins	0.903	0.984	0.997	**0.103**	0.364

- Comparison of performances on the **KITTI** dataset

Method	δ_1	δ_2	δ_3	REL	RMS
BTS (Lee et all, arXiv preprint 2019)	0.956	0.993	0.998	0.059	2.756
Transfer Learning (Dense-Net 169)	**0.886**	**0.965**	**0.986**	0.093	4.170
Dense Prediction Transformer (DPT)	0.959	0.995	0.999	0.062	2.573
AdaBins	0.964	0.995	0.999	**0.058**	**2.360**

Implementation Results:

- Testing last two approaches on RGB photos taken by ourselves from campus (all are indoor)



Thanks for your Attention

UNIVERSITY OF ALBERTA