High Quality Monocular Depth Estimation via Transfer Learning

Constantinescu Maria-Ecaterina
Enache George-Vlad
Ghiorghiu Bianca-Alexandra

Table of Contents

- Introduction
- Related Work
- Dataset
- Data Augmentation
- Encoder-Decoder
- Implementation Details
- Loss Functions
- Evaluation
- Studies and Experiments

Introduction

- Accurate depth estimation from images is important for scene understanding and reconstruction.
- Existing solutions for depth estimation often produce low-resolution and blurry results.
- The paper presents a convolutional neural network for high-resolution depth map estimation from a single RGB image using transfer learning.
- The proposed model follows an encoder-decoder architecture with high-performing pre-trained networks for feature extraction.
- Data augmentation and training strategies are used to produce more accurate results.
- The network outperforms state-of-the-art methods on two datasets and produces qualitatively better results that capture object boundaries more faithfully.

Related Work

- **Monocular depth estimation:** process of estimating the depth information of a scene from a single camera or image. It involves using computer vision techniques to infer the 3D structure of a scene from a 2D image
- **Single image depth estimation**
- **Transfer learning:** a technique in machine learning where a model trained on one task is reused as the starting point for a model on a new, related task.
- **Encoder-decoder architecture:** consists of an encoder that compresses the input into a compact representation and a decoder that generates the output from that representation. It can effectively capture complex patterns and dependencies in the data, making it useful for transforming one type of input into another.

Dataset

The **KITTI** dataset is a widely used computer vision dataset that provides images and sensor data collected from a car driving in urban environments.

The dataset includes over 120k high-resolution images, lidar point clouds, and calibrated camera poses, and is primarily used for tasks such as object detection, object tracking, and depth estimation.

The **NYU Depth v2** dataset is a popular computer vision dataset that provides RGB-D images captured from a Microsoft Kinect camera. The dataset includes over 1449 densely labeled pairs of aligned RGB and depth images, along with segmentation masks for 27 object classes.

The dataset is a benchmark dataset for depth estimation research and has been used to evaluate many state-of-the-art depth estimation models.

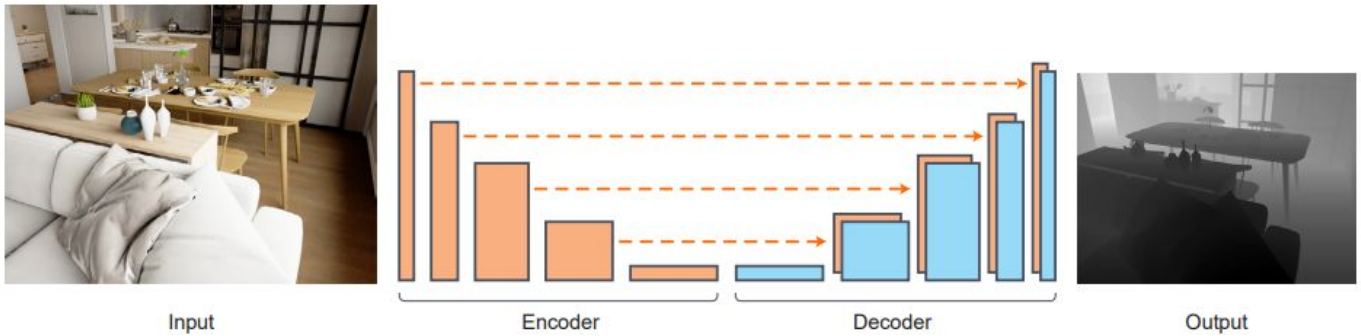
Data Augmentation

- Horizontal flipping (i.e. mirroring) of images at a probability of 0.5.

- Applying different color channel permutations, e.g. swapping the red and green channels on the input, results in increased performance while also being extremely efficient. The probability for this color channel augmentation was set to 0.25.

Encoder-Decoder

- Encoder: pre-trained truncated DenseNet-169
- Decoder: basic blocks of convolutional layers applied on the concatenation of the 2× bilinear upsampling of the previous block with the block in the encoder with the same spatial size after upsampling
- Skip connections



Encoder-Decoder

- The input RGB image is encoded into a feature vector using the DenseNet-169 network pretrained on ImageNet
- This vector is then fed to a successive series of up-sampling layers in order to construct the final depth map at half the input resolution.
- These upsampling layers and their associated skip-connections form the decoder, which does not contain any Batch Normalization or other advanced layers recommended in recent state-of-the-art methods

Layer	Output	Function
INPUT	$480 \times 640 \times 3$	
CONV1	$240 \times 320 \times 64$	DenseNet CONV1
POOL1	$120 \times 160 \times 64$	DenseNet POOL1
POOL2	$60 \times 80 \times 128$	DenseNet POOL2
POOL3	$30 \times 40 \times 256$	DenseNet POOL3
...	...	...
CONV2	$15 \times 20 \times 1664$	Convolution 1×1 of DenseNet BLOCK4
UP1	$30 \times 40 \times 1664$	Upsample 2×2
CONCAT1	$30 \times 40 \times 1920$	Concatenate POOL3
UP1-CONVA	$30 \times 40 \times 832$	Convolution 3×3
UP1-CONVB	$30 \times 40 \times 832$	Convolution 3×3
UP2	$60 \times 80 \times 832$	Upsample 2×2
CONCAT2	$60 \times 80 \times 960$	Concatenate POOL2
UP2-CONVA	$60 \times 80 \times 416$	Convolution 3×3
UP2-CONVB	$60 \times 80 \times 416$	Convolution 3×3
UP3	$120 \times 160 \times 416$	Upsample 2×2
CONCAT3	$120 \times 160 \times 480$	Concatenate POOL1
UP3-CONVA	$120 \times 160 \times 208$	Convolution 3×3
UP3-CONVB	$120 \times 160 \times 208$	Convolution 3×3
UP4	$240 \times 320 \times 208$	Upsample 2×2
CONCAT3	$240 \times 320 \times 272$	Concatenate CONV1
UP2-CONVA	$240 \times 320 \times 104$	Convolution 3×3
UP2-CONVB	$240 \times 320 \times 104$	Convolution 3×3
CONV3	$240 \times 320 \times 1$	Convolution 3×3

Implementation Details

- Weights of decoder are randomly initialized
- ADAM optimizer with learning rate 0.0001 and parameter values $\beta 1 = 0.9$, $\beta 2 = 0.999$
- Batch size set to 8
- 42.6M trainable parameters
- 1M iterations for NYU Depth v2, 300k iterations for KITTI

Loss Function

$$L(y, \hat{y}) = \lambda L_{depth}(y, \hat{y}) + L_{grad}(y, \hat{y}) + L_{SSIM}(y, \hat{y})$$

$$L_{depth}(y, \hat{y}) = \frac{1}{n} \sum_{p}^{n} |y_p - \hat{y}_p|$$

$$L_{grad}(y, \hat{y}) = \frac{1}{n} \sum_{p}^{n} |\boldsymbol{g_x}(y_p, \hat{y}_p)| + |\boldsymbol{g_y}(y_p, \hat{y}_p)|$$

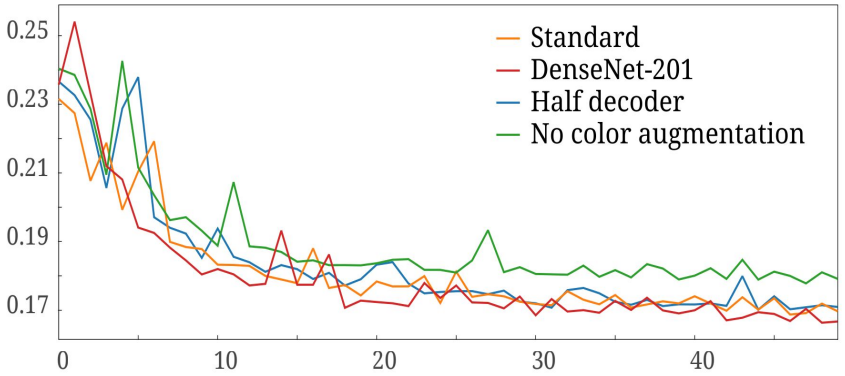
$$L_{SSIM}(y, \hat{y}) = \frac{1 - SSIM(y, \hat{y})}{2}$$

Evaluation

- average relative error (rel): $\frac{1}{n}\sum_p^n \frac{|y_p - \hat{y}_p|}{y}$;
- root mean squared error (rms): $\sqrt{\frac{1}{n}\sum_p^n (y_p - \hat{y}_p)^2}$;
- average ($\log_{10}$) error: $\frac{1}{n}\sum_p^n |\log_{10}(y_p) - \log_{10}(\hat{y}_p)|$;
- threshold accuracy (δ_i): % of y_p s.t. $\max(\frac{y_p}{\hat{y}_p}, \frac{\hat{y}_p}{y_p}) = \delta < thr$ for $thr = 1.25, 1.25^2, 1.25^3$;

Studies and Experiments

- **Encoder Depth Variation.** The pre-trained DenseNet-169 Encoder has been replaced with the deeper DenseNet-201. The performance gains did not outweighed the massive increase in learning time.
- **Decoder Depth Variation.** The features fed into the decoder have been halved in depth. A decrease in model performance and an added instability has been observed.
- **Color Augmentation.** The color channel swapping-based augmentation has been turned off. A significant decrease in performance has been observed.



References

- [1812.11941] High Quality Monocular Depth Estimation via Transfer Learning
- Digging Into Self-Supervised Monocular Depth Estimation | Papers With Code
- [1702.02706] Semi-Supervised Deep Learning for Monocular Depth Map Prediction