Monocular Depth Estimation On Low Light Images via Transfer Learning

Abstract

Depth estimation is a classical problem in computer vision that is essential for many applications including scene understanding, scene reconstruction. Recent developments in convolutional neural networks (CNNs) have helped to gain a leap in the performance; there are still major problems in both quality and the resolution of these estimated depth maps. Incase of low light or monochrome images it gets even harder to estimate depth. We propose to use transfer learning in estimating depth from low light or monochrome grayscale images using a standard encoder-decoder architecture, by leveraging features extracted using high performing pre-trained networks when initializing our encoder. We show how, even for a simple decoder, our method is able to achieve close to state-of-the-art high resolution depth maps on the NYU depth dataset. Code and corresponding pre-trained weights are made publicly available.

1. Introduction

Accurate depth estimation from 2D images is a classical problem in computer vision that is essential for many applications including scene understanding, scene reconstruction. Having an accurate depth map of the real-world can be very useful in understanding scene, augmented reality, and image segmentation. Recent developments in convolutional neural networks (CNNs), visual transformers have helped to gain a leap in the performance in 2D and 3D reconstructions. While the performance has steadily increased, there are still major problems in both quality and the resolution of these estimated depth maps. Especially for low light or monochrome images it gets even harder to estimate depth. Recent applications in self-driving cars at night, drones navigation at night require fast computation of high resolution 3D reconstruction in order to be applicable. For such applications, it is critical to reconstruct discontinuity in the depth maps and avoid the large perturbations that are often present in depth estimation using current CNNs. Based on our experiments and analysis of existing architectures and training strategies, we set out to develop a simpler architecture that makes training and future modifications easier. To achieve this, we rely on transfer learning where we leverage features extracted using high performing pre-trained networks when initializing our encoder. This type of architecture makes the network more modular, where future advancements in one domain are easily transferable to depth estimations problems.

Contributions: We propose a simple encoder-decoder network architecture that produces depth estimations of higher accuracy and quality. Secondly, We propose a future work to create a testing dataset of photorealistic synthetic scenes, with perfect ground truth, to better evaluate depth estimation, using state of the art game engines (Unreal Engine, Unity)

2. Related Work

In the past, the approach to capturing a scene's depth has highly relied on hardware assistance which is expensive and time consuming. Recently, methods that rely on CNNs are able to produce reasonable depth maps from a single or multiple RGB images at real-time speeds. Let's look into recent solutions that are deep on deep neural networks.

Monocular depth estimation has been considered by many CNN methods as a regression of a dense depth map from a single RGB image. Performance of these methods have been increasing steadily over the years, both quality and resolution of the estimated depth maps leaves a lot of room for improvement. Preliminary results do indicate that a simple architecture like ours is able to produce close to state-of-the-art outputs.

Encoder-decoder and Transformer have made significant contributions in many NLP and vision related problems. In recent years, the use of such architectures have gained popularity as viable building blocks outside of their traditional use in NLP tasks and computer vision tasks. Such methods typically use one or more encoder-decoder networks as a sub part of their larger network. In this work, we employ a single straightforward encoder-decoder architecture with skip connections (see Fig. 1). Our results indicate that it is possible to achieve state-of-the-art high quality depth maps using a simple encoder-decoder architecture.

Transfer learning approaches have been shown to be very helpful in many different contexts. In recent work, Zamir et al. investigated the efficiency of transfer learning between different tasks, many of which are related to 3D reconstruction.

We propose to leverage heavily on this idea of transfer learning where we make use of image encoders originally designed for problems in image classification.

3. Method

In this section, we describe our method for estimating a depth map from a low light single RGB image. We first describe pre-processing step for data set. We then describe the encoder-decoder architecture. Next, we discuss experiments done and observations on complexity of both enoder and decoder and its relation to performance.

3.1. Network Architecture

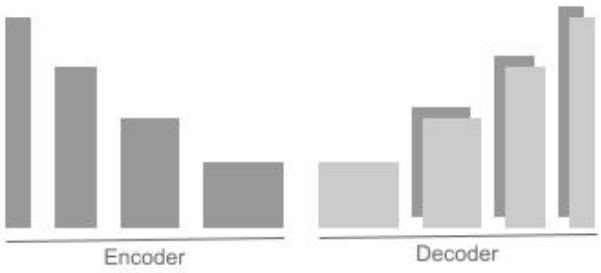

Figure 1. Overview of network architecture.

Fig. 1 and Tab. 1 is the representation of simple network architecture used for depth estimation. For encoder, the input RGB image is enccoded into a feature vector using DenseNet-169 newtork pre-trained on ImageNet. This vector is then fed to a successive series of up-sampling layers, in order to construct the final depth map at 75 percent of its input resolution. We should be experimenting with decoders containing batch normalization or other advanced layers recommended in the recent state of the art methods.

3.2. Loss Function.

A standard loss function for depth regression problems considers the difference between the ground truth depth map y and the prediction of the depth regression network $\hat{y}$.

3.3. Augmentation policy

Data augmentation by geometry and photo-metric transformations, is a standard to reduce overfitting leading to better generalization performance. Since our network is designed to estimate depth maps of an entire image, not all geometric transformations would be appropriate since vertical flip to an image capturing indoor scene may not contribute to the learning expected properties. Only horizontal flipping is considered.

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

Table 1. **Network architecture**

3.4. Evaluation

We quantitatively compare our method against state-of-the-art using the standard metrics used in prior work. These error metrics are defined as:

where y_p is a pixel in depth image y, $\hat{y}_p$ is a pixel in the predicted depth image $\hat{y}$, and n is the total number of pixels for each depth image.

- 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: $\frac{1}{n}\sum_p^n |\log_{10}(y_p) - \log_{10}(\hat{y}_p)|$;

4. Experimental Details

In this section we describe our experimental results and compare the performance of our network to existing state-of-the-art methods. We train our method on a subset

of around 5k images. The depth maps have an upper bound of 10 meters. Our encoder's architecture expects image dimensions to be divisible by 32, therefore we scale up or down based on the image size. We reduce the image brightness to 80%, 60%, 40%, 20% of the original Image brightness using advanced image processing techniques.

4.1. Implementation Details

We implemented our model using PyTorch and trained on one NVIDIA Tesla P100 GPU with 16GB memory. The weights for the decoder are randomly initialized. In all experiments, we used the ADAM optimizer with learning rate 0.0001 and parameter values $\beta 1 = 0.9$, $\beta 2 = 0.999$. The batch size is set to 5. The total number of trainable parameters for the entire network is approximately 21.3M parameters roughly half the size of base model.

4.2. Qualitative results.

We conducted experiments with four different brightness strengths 80%, 60%, 40%, 20% on NYU Depth V2 data set by freezing weight of encoder(DenseNet-161) and only learning decoder parameters along with experiment with original image and not freezing encoder. The first measure is a perception-based qualitative metric that measures the quality of the results by looking at the similarity of the resulting depth maps in image space. We do so by rendering a gray scale visualization of the ground truth and that of the predicted depth map. In Fig **??** blue, pink, orange, purple represents brightness strength of image reduced to 20%, 40%, 60%, 80% respectively. Fig. 4.2 shows a gallery of depth estimation results that are predicated using our method.

Tab. 3 and Tab. **??** show the results of our method compared to other methods and our method used on different low light images.

5. Data

NYU Depth v2 is a dataset that provides images and depth maps for different indoor scenes captured at a resolution of 640×480. The dataset contains 120K training samples and 654 testing samples. We train our method on a 50K subset. Missing depth values are filled using the inpainting method of. The depth maps have an upper bound of 10 meters. Our network produces predictions at half the input resolution, i.e. a resolution of 320×240. For training, we take the input images at their original resolution and downsample the ground truth depths to 320×240. Note that we do not crop any of the input image-depth map pairs even though they contain missing pixels due to a distortion correction preprocessing. At test time, we compute the final output by taking the average of an image's prediction and the prediction of its mirror image.

KITTI is a dataset that provides stereo images and corresponding 3D laser scans of outdoor scenes captured using equipment mounted on a moving vehicle. The RGB images have a resolution of around 1241×376 while the corresponding depth maps are of very low density with lots of missing data. We train our method on a subset of around 26K images, from the left view, corresponding to scenes not included in the 697 test set. The depth maps have an upper bound of 80 meters. At test time, we compute the final output by taking the average of an image's prediction and the prediction of its mirror image.

6. Conclusion

In this work, we proposed a convolutional neural network for depth map estimation for single RGB images by leveraging recent advances in network architecture and the availability of high performance pre-trained models. We show that by leveraging start-of-the-art pre-trained models for classification used to transfer knowledge and produce close to start-of-the-art depth estimations from a simple encoder decoder architecture, irrespective of what brightness of the images we train the model with.

7. References

[1] Bhat, Shariq Farooq, Ibraheem Alhashim, and Peter Wonka. "Adabins: Depth estimation using adaptive bins." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[2] X. Ye et al., "Deep Joint Depth Estimation and Color Correction From Monocular Underwater Images Based on Unsupervised Adaptation Networks," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 11, pp. 3995-4008, Nov. 2020, doi: 10.1109/TCSVT.2019.2958950.

[3] Choi, J., Jung, D., Lee, D. and Kim, C. (2020). SAFENet: Self-Supervised Monocular Depth Estimation with Semantic-Aware Feature Extraction. arXiv preprint arXiv:2010.02893

[4] S. Bianco, R. Cadene, L. Celona, and P. Napoletano. Benchmark analysis of representative deep neural network architectures. IEEE Access, 6:64270–64277, 2018.

[5] N. Kim, Y. Choi, S. Hwang, and I. S. Kweon, "Multispectral Transfer Network: Unsupervised Depth Estimation for All-Day Vision", AAAI, vol. 32, no. 1, Apr. 2018.

Method	$\delta 1 \uparrow$	$\delta 2 \uparrow$	$\delta 3 \uparrow$	rel $\downarrow$	rms $\downarrow$	log10 $\downarrow$
Eigen et al.	0.769	0.950	0.988	0.158	0.641	-
Hao et al	0.841	0.966	0.991	0.127	0.555	0.053
Fu et al.	0.828	0.965	0.992	0.115	0.509	0.051
Base model	0.848	0.96	0.994	0.141	0.478	0.0570
Ours (freezing encoder)	0.824	0.95	0.981	0.141	0.478	0.0570

Table 2. Comparisons of different methods on the NYU Depth v2 dataset.

Method	$\delta 1 \uparrow$	$\delta 2 \uparrow$	$\delta 3 \uparrow$	rel $\downarrow$	rms $\downarrow$	log10 $\downarrow$
Original Image	0.895	0.96	0.994	0.141	0.468	0.053
Brightness 80%	0.84	0.964	0.993	0.1393	0.495	0.054
Brightness 60%	0.855	0.9677	0.992	0.1347	0.501	0.055
Brightness 40%	0.8128	0.9644	0.9916	0.1473	0.503	0.059
Brightness 20%	0.7581	0.9403	0.9892	0.1700	0.589	0.0691

Table 3. Comparisons of our methods on the NYU Depth v2 dataset with different brightness.

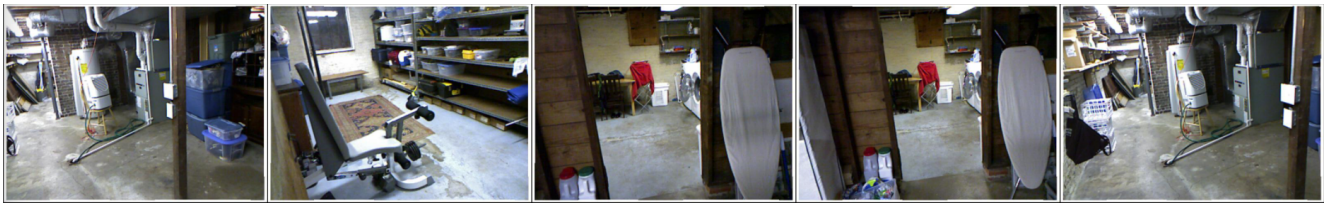

Figure 2. Original Image

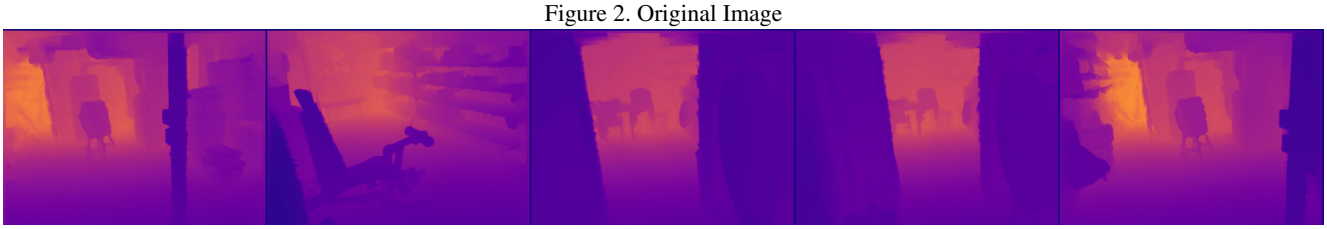

Figure 3. Ground Truth Depth

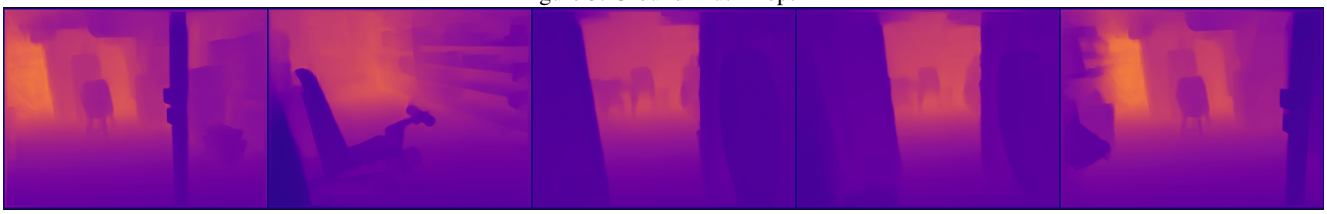

Figure 4. Estimated Original Depth

Figure 5. Brightness 60%

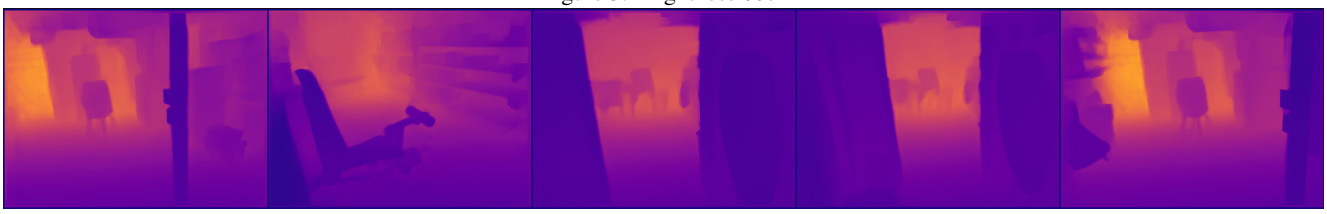

Figure 6. Estimated depth on brightness 60%

Figure 7. Brightness 20%

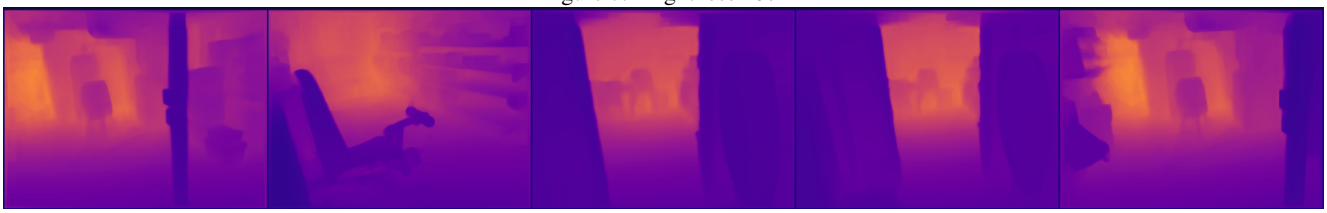

Figure 8. Estimated depth on brightness 20%

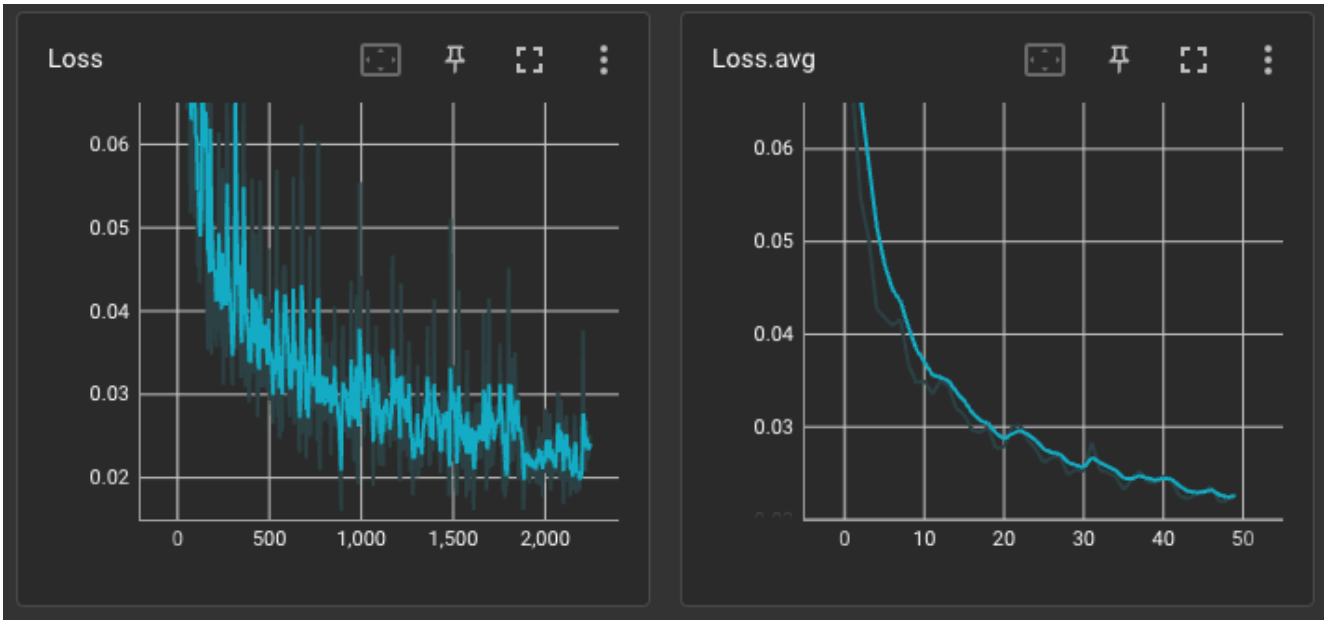

Figure 9. Loss vs Iteration of the original images

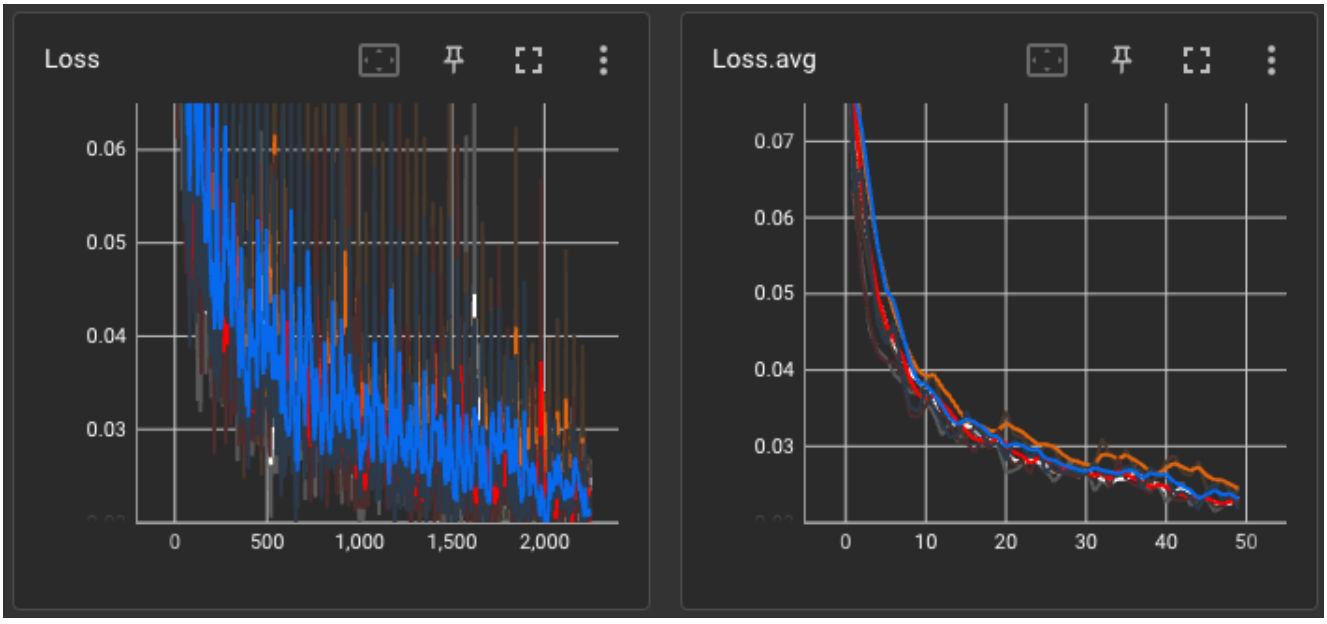

Figure 10. Loss vs Iteration for brightness of 20%, 40%. 60%, 80%