AsiANet: Autoencoders in Autoencoder for Unsupervised Monocular Depth Estimation

Conference Paper · January 2019

DOI: 10.1109/WACV.2019.00053

2 authors, including:

John Paul Yusiong
University of the Philippines Visayas
16 PUBLICATIONS **133** CITATIONS

SEE PROFILE

2019 IEEE Winter Conference on Applications of Computer Vision

AsiANet: Autoencoders in Autoencoder for Unsupervised Monocular Depth Estimation

John Paul T. Yusiong[1,2] Prospero C. Naval, Jr.[1]
[1]Computer Vision and Machine Intelligence Group, Department of Computer Science,
College of Engineering, University of the Philippines Diliman,
Diliman, Quezon City, Philippines
[2] Division of Natural Sciences and Mathematics,
University of the Philippines Visayas Tacloban College,
Tacloban City, Leyte, Philippines
jtyusiong@up.edu.ph pcnaval@dcs.upd.edu.ph

Abstract

Monocular depth estimation is extremely challenging because it is inherently an ambiguous and ill-posed problem. The unsupervised approach to monocular depth estimation using convolutional neural networks is gaining a lot of interest since learning from a set of rectified stereo image pairs without ground truth depths and predicting scene geometry from a single image have become feasible. The proposed approach requires training an encoder-decoder network architecture, referred to as autoencoders in autoencoder (AsiANet), in an unsupervised fashion to discover the implicit relationship between a single image and its corresponding depth map. AsiANet uses a unique Inception-like pooling module based on fractional max-pooling for dimensionality reduction. Experiments on the KITTI benchmark dataset show that the proposed architecture trained using the Charbonnier loss function achieved superior performance on depth map prediction compared to previous unsupervised monocular depth estimation methods.

1. Introduction

Depth estimation is a fundamental problem in computer vision and is essential for automated driving systems, robotics, and scene understanding. It involves identifying geometric properties of a scene with the goal of predicting the depth values of every pixel in the image [30, 32]. In recent years, researchers have developed different depth estimation methods that either improve on model accuracy, minimize the computational cost to improve on the efficiency of the model or provide other ways of handling uniform regions, occluded areas, and depth discontinuities [22]. These include learning-based depth estimation methods that are either supervised or unsupervised [47] while depth estimation problems can either be monocular or multi-view [11]. However, in this paper, the emphasis is on the unsupervised learning for monocular depth estimation which involves estimating depth directly from rectified stereo image pairs without ground truth depth labels.

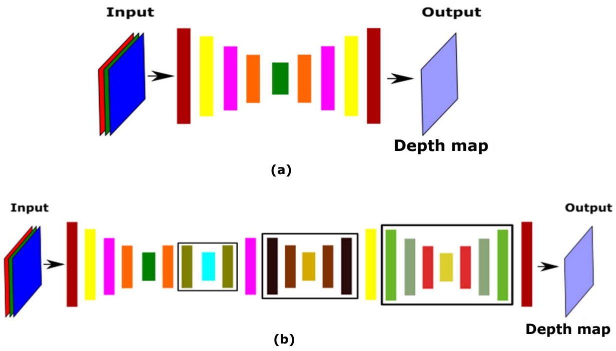

Figure 1. Autoencoder versus Autoencoders in autoencoder.

Monocular depth estimation (MDE) is extremely challenging because it is inherently ambiguous and ill-posed. Monocular depth estimation involves assigning a depth value to each pixel, which requires both global and local scene information from various cues to infer depth from a single image [9]. Nevertheless, despite the challenges, various metrics used in evaluating existing monocular depth estimation methods confirm that recovering depth information of a scene from a single image is achievable [4]. Buoyed by the remarkable success of convolutional neural networks (CNNs) in a variety of computer vision tasks, researchers are now using CNNs to perform monocular depth estimation. However, the success of CNN-based monocular depth estimation methods to handle complex non-linear

978-1-7281-1975-5/19/$31.00 ©2019 IEEE
DOI 10.1109/WACV.2019.00053

tasks is highly dependent on the design of the CNN architecture [11, 15, 25]. Existing methods for unsupervised monocular depth estimation have certain things in common such as adopting an autoencoder architecture based on DispNet, U-Net, ResNet among others, using a loss function that consists of multiple terms, and applying L1 or L2 to the terms in the loss function.

This research work is another step towards CNN-based monocular depth estimation using the unsupervised learning approach. Our paper is substantially different from the previous studies since we present the first work in the area of unsupervised monocular depth estimation that adopts a stacked autoencoder architecture, as shown in Figure 1(b), in a multi-scale setting with a fractional max-pooling module, and applies the Charbonnier function instead of L1 or L2 to the terms in the loss function. Specifically, we propose an unsupervised monocular depth estimation method using an encoder-decoder network architecture referred to as autoencoders in autoencoder or AsiANet to improve performance over existing state-of-the-art methods. Our key contribution is we employ a unique Inception-like pooling module based on fractional max-pooling at the encoder section and use multi-scale cascaded autoencoders at the decoder side to exploit features at multiple scales when upsampling the output of the encoder in order to gain better local and global context. We evaluate the performance of AsiANet using the Charbonnier loss function. Lastly, we demonstrate the effectiveness of the new approach on a challenging driving dataset.

2. Related Work

Recent works on depth estimation exploit deep learning methods [17] to achieve better performance than the conventional methods that depend on hand-crafted features, probabilistic graphical models, and non-parametric approaches [19, 21, 27, 31]. Although researchers continue to focus on multi-view depth estimation techniques, there is an increasing interest in monocular depth estimation due to the availability of images captured via monocular imaging systems that are prevalent in today's consumer products [37]. In this section we will only consider deep learning-based monocular depth estimation methods using either supervised or unsupervised approaches.

Supervised monocular depth estimation methods have advanced dramatically despite the need to acquire a vast amount of data with manually annotated ground truth depth. Laina *et al.* [25] proposed a modified residual neural network where a novel up-sampling block replaced the fully-connected layer to obtain fine-grained depth predictions. Eigen *et al.* [9] proposed the use of multi-scale CNNs for monocular depth estimation and extended their work to simultaneously perform monocular depth estimation, surface normal estimation, and semantic labeling tasks [8]. Other researchers combined CNN with other paradigms such as random forests [30] and conditional random fields (CRFs) [47, 40, 26]. Unlike the previous works mentioned which considered the monocular depth estimation problem as a regression task, Cao *et al.* [5] formulated the problem as a classification task, while Fu *et al.* [11] proposed a regression-classification approach based on the so-called "compromise principle."

In unsupervised monocular depth estimation, ground truth depth labels are not necessary during training and only requires single RGB images during the testing phase. A typical approach involves formulating the problem as an image reconstruction task. For instance, Garg *et al.* [12] and Xie *et al.* [39] proposed a CNN-based approach to predict the depth map for the source image by minimizing the image reconstruction error. Godard *et al.* [15], used a similar approach but added a left-right consistency constraint. Moreover, in [43], unsupervised monocular depth estimation has been applied in robotic surgery using a framework that consists of an autoencoder and a differentiable spatial transformer. Also, instead of merely training a CNN model for monocular depth estimation, Zhou *et al.* [46] suggested to jointly train a depth estimation CNN and a camera pose estimation CNN from unlabeled videos while Yang *et al.* [41] proposed a framework for joint depth, surface normal and edge learning.

Kuznietsov *et al.* [23] proposed to perform monocular depth estimation in a semi-supervised manner by making use of the supervised and unsupervised methods simultaneously. The proposed network architecture is based on the encoder-decoder scheme and trained using a novel semi-supervised loss function.

On the other hand, Yin and Shi [44] considered motion decomposition in performing depth estimation while other researchers made use of different constraints during training, such as constraining the depth predictions to be consistent with the predicted surface normals [42], enforcing temporal depth consistency by using an approximate ICP-based geometry loss [28], and normalizing the predicted depth maps before computing the smoothness term [36].

Nonetheless, the usual approach for these researchers was to include a single convolutional autoencoder architecture in their framework. In contrast, we present a multi-scale cascaded convolutional autoencoder architecture and use this architecture for unsupervised monocular depth estimation. Moreover, unlike some existing architectures [29, 35] that feature similar stacked autoencoder structures to solve other tasks, our proposed model solves the unsupervised monocular depth estimation problem by applying this technique only during upsampling which is after the encoding stage of the network, and each of these autoencoders has a different scale and a different network structure depending on the scale.

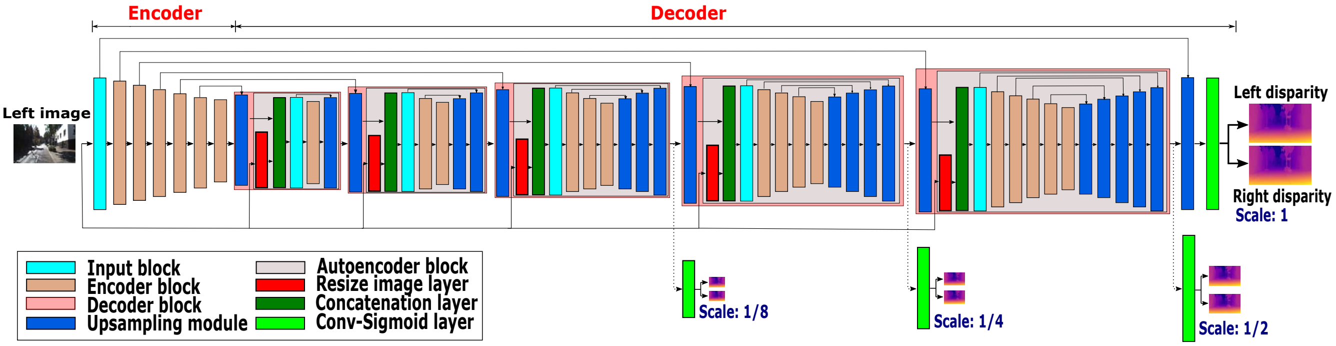

Figure 2. AsiANet for unsupervised monocular depth estimation.(Best viewed in color.)

3. The Autoencoders in Autoencoder Network for MDE

This paper examines the monocular depth estimation problem as an image reconstruction task using the unsupervised learning approach. The solution involves training an autoencoder architecture, the AsiANet (see Fig. 2) to generate a per-pixel disparity from a single RGB image. The training process is similar to [15, 43], but we modified it by using our proposed architecture and a Charbonnier loss function [3]. Moreover, the framework requires a set of rectified stereo image pairs as inputs during training. Hence, datasets that consist of RGB/Depth (RGB-D) pairs only or those without stereo images, such as Make3D (outdoor scenes) [31] and NYU Depth v2 (indoor scenes) [33], cannot be used to train AsiANet.

3.1. Network Architecture

As described in [9], developing multi-scale CNN models is a step towards improving performance. Following this view, AsiANet utilizes multi-scale cascaded autoencoders at the decoder section of an autoencoder with long skip connections between the encoder and decoder. This design facilitates in exploiting features at multiple scales when upsampling the output of the encoder section and generating fine detail estimates arising from improved local and global context information.

A considerable amount of effort was taken to highlight the technical details of the network architecture for reproducibility. Fig. 2 depicts the overall network architecture of AsiANet while Table 1 details its main components. The Encoder section consists of the Input block and several encoder blocks while the Decoder section consists of several

Table 1. Main Components of AsiANet. Conv: Convolutional layer; ELU: Exponential Linear Unit; IFMP: Inception-like fractional max-pooling module; Deconv: Deconvolutional layer.

Block/Module	Composition
Input Block	Conv-ELU-Conv-ELU
Encoder Block	IFMP-Conv-ELU-Conv-ELU
Decoder Block	Upsampling-Autoencoder
IFMP Module	(see Fig. 3)
Upsampling Module	Deconv-ELU-Addition-Conv-ELU-Conv-ELU
Autoencoder (AE) Module	Image Resize-Concatenation-AE[32, 16, 8, 4, 2]

Table 2. Autoencoder (AE) Modules. I: Input block; E: Encoder block; U: Upsampling module; Dim: Input image dimension.

AE Module	Module I/O	Composition	Output Channels	Output Dimension
AE[32]	1027/32	I-E-U	32-64-32	Dim/ [32-64-32]
AE[16]	515/32	I-E-E-U-U	32-64-128-64-32	Dim/ [16-32-64-32-16]
AE[8]	259/32	I-E-E-E-U-U-U	32-64-128-256-128-64-32	Dim/ [8-16-32-64-32-16-8]
AE[4]	131/32	I-E-E-E-E-U-U-U-U	32-64-128-256-512-256-128-64-32	Dim/ [4-8-16-32-64-32-16-8-4]
AE[2]	67/32	I-E-E-E-E-E-U-U-U-U-U	32-64-128-256-512-1024-512-256-128-64-32	Dim/ [2-4-8-16-32-64-32-16-8-4-2]

Table 3. Network Architecture of AsiANet. I: Input block; E: Encoder block; D: Decoder block; Dim: Input image dimension; Image: left RGB image; disp: disparity map; AE: Autoencoder Module; Up: Upsampling layer; IR: Image resize layer; C_S: Convolutional layer with Sigmoid activation function.

Block	Module/Layer	I/O	Input	Output Dimension
I1	-	3/32	Image	Dim
E1	-	32/64	I1	Dim/2
E2	-	64/128	E1	Dim/4
E3	-	128/256	E2	Dim/8
E4	-	256/512	E3	Dim/16
E5	-	512/1024	E4	Dim/32
E6	-	1024/2048	E5	Dim/64
D6	Up6	2048/1024	E6, E5	Dim/32
	AE[32]	1027/32	Up6, IR[32]	Dim/32
D5	Up5	32/512	AE[32], E4	Dim/16
	AE[16]	515/32	Up5, IR[16]	Dim/16
D4	Up4	32/256	AE[16], E3	Dim/8
	AE[8]	259/32	Up4, IR[8]	Dim/8
disp4	C_S	32/2	AE[8]	Dim/8
D3	Up3	32/128	AE[8], E2	Dim/4
	AE[4]	131/32	Up3, IR[4]	Dim/4
disp3	C_S	32/2	AE[4]	Dim/4
D2	Up2	32/64	AE[4], E1	Dim/2
	AE[2]	67/32	Up2, IR[2]	Dim/2
disp2	C_S	32/2	AE[2]	Dim/2
-	Up1	32/32	AE[2], I1	Dim
disp1	C_S	32/2	Up1	Dim

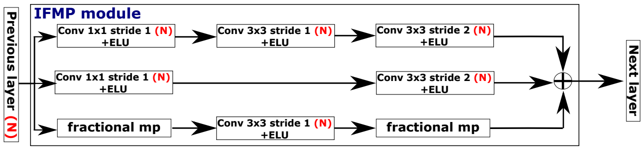
IFMP module
Previous layer (N)
Conv 1x1 stride 1 (N) +ELU
Conv 3x3 stride 1 (N) +ELU
Conv 3x3 stride 2 (N) +ELU
Conv 1x1 stride 1 (N) +ELU
Conv 3x3 stride 2 (N) +ELU
fractional mp
Conv 3x3 stride 1 (N) +ELU
fractional mp
Next layer

Figure 3. The IFMP module.

decoder blocks. Convolutional and deconvolutional layers within a block/module use the same kernel of size 3×3 and have the same number of output channels. An exponential linear unit (ELU) [6] is applied to the output of these layers except for the last convolutional layers where the sigmoid activation function is used to generate the disparity maps similar to [15].

Also, instead of using the max-pooling layer to reduce spatial dimension, we introduce an Inception-like [34] pooling module that uses fractional max-pooling (FMP $\sqrt{2}$) [16], as shown in Fig. 3. Like the Inception module, this pooling module improves the performance of the model since the model can obtain multiple features from multiple filters while reducing the spatial dimension.

In the decoder section, instead of merely performing upsampling, we proposed to perform "autoencoding" after each upsampling step to exploit features at multiple scales and obtain fine detail estimates at every scale. Table 2 details the structure of the autoencoder (AE) module at different scales while Table 3 defines the various layers of AsiANet.

AsiANet predicts a pair of disparity maps, D_L and D_R, from image I_L at four different scales. The sampler from the differentiable spatial transformer (ST) [18] transforms these maps along with image I_R to reconstruct I_L^* and I_R^* at each scale by performing bilinear interpolation. The primary objective of training the network is to minimize the image reconstruction errors between the input image I and the reconstructed image I^* using a loss function based on the Charbonnier penalty function.

3.2. The Charbonnier Loss Function

A typical loss function involves the L_1 penalty function and consists of a weighted sum of different terms [15, 43]. However, training the network with a Charbonnier loss function has been shown to achieve high-quality image reconstruction. A Charbonnier loss function is a smooth approximation and a differentiable variant of L_1-norm, and is very effective for gradient-based optimization methods because it can handle outliers better [10, 24]. In this paper, we replaced the L_1 penalty function used in [15, 43] with the Charbonnier penalty function as shown in (1). Like in [15], our Charbonnier loss function consists of three terms as shown in (2).

$$||\mathbf{Z}||_1 = \frac{1}{N}\sum_{i,j}|Z_{i,j}| \approx \rho(\mathbf{Z}) = \frac{1}{N}\sqrt{\mathbf{Z}^2 + \varepsilon^2}, \quad \varepsilon = 0.001 \quad (1)$$

$$\mathcal{C}_s = \alpha\mathcal{C}_{app} + \beta\mathcal{C}_{smooth} + \gamma\mathcal{C}_{cons} \quad (2)$$

where N is the number of pixels in the image, $\mathcal{C}_{app} = \mathcal{C}_{app}^L + \mathcal{C}_{app}^R$, $\mathcal{C}_{smooth} = \mathcal{C}_{smooth}^L + \mathcal{C}_{smooth}^R$, $\mathcal{C}_{cons} = \mathcal{C}_{cons}^L + \mathcal{C}_{cons}^R$, and α, β, γ are constants. $\mathcal{C}_{app}$ measures the deviation of the reconstructed image from the corresponding input image in the training set, $\mathcal{C}_{smooth}$ enforces smoothness by penalizing discontinuities, and $\mathcal{C}_{cons}$ ensures consistency of the two predicted disparity maps.

In terms of the left image, Eqns (3), (4), and (5) define the appearance dissimilarity term consisting of a single-scale SSIM [38] and a Charbonnier function, a spatial (disparity) smoothness term with a Charbonnier penalty and based on the second-order disparity gradients weighted by the image gradients [45], and the left-right disparity consistency term [15], respectively. For any non-negative $\omega < 1$,

$$\mathcal{C}_{app}^L = \frac{1}{N}\sum_{i,j}\omega\frac{1-SSIM(I_L(i,j),I_L^*(i,j))}{2} + (1-\omega)\,\rho\left(I_L - I_L^*\right) \quad (3)$$

$$\mathcal{C}_{smooth}^L = \rho\left(\partial_x^2 D_L\right)e^{-\rho\left(\partial_x^2 I_L\right)} + \rho\left(\partial_y^2 D_L\right)e^{-\rho\left(\partial_y^2 I_L\right)} \quad (4)$$

$$\mathcal{C}_{cons}^L = \rho\left(\hat{D}_L\right), \text{ where } \hat{D}_L(i,j) = D_L(i,j) - D_R(i, j - D_L(i,j)). \quad (5)$$

In terms of the right image, the corresponding terms of the loss function are solved similarly, except for the left-right disparity consistency term which is defined in (6). Hence, for each stereo image pair in the training set, we compute for the loss using (2) at all of the output scales and obtain their mean during the training phase.

$$\mathcal{C}_{cons}^R = \rho\left(\hat{D}_R\right), \text{ where } \hat{D}_R(i,j) = D_R(i,j) - D_L(i, j + D_R(i,j)). \quad (6)$$

As shown in (2), our training loss at each scale s is a combination of three terms: appearance dissimilarity, disparity smoothness and left-right consistency, and is aggregated through four different scales for a total loss of $\mathcal{L} = \sum_{s=1}^{4}\mathcal{C}_s$.

3.3. Disparity and Depth map outputs

The depth information Z, which is the distance along the camera's Z-axis (in meters), can easily be recovered using (7) given a disparity map d (in pixels), along with B, the baseline distance between cameras (in meters), and f, the focal length of the camera (in pixels) [15].

$$Z = \frac{fB}{d} \quad (7)$$

Although AsiANet predicts a pair of disparity maps, D_L and D_R, from image I_L at different output scales, only the left disparity map, D_L, at the highest scale (scale 1) is relevant during the testing and evaluation phases. In the testing phase, the trained AsiANet model accepts as input a single RGB image and predicts a disparity map. Since the baseline and focal length are known, then the equivalent depth map can be obtained using (7). The evaluation phase assesses the

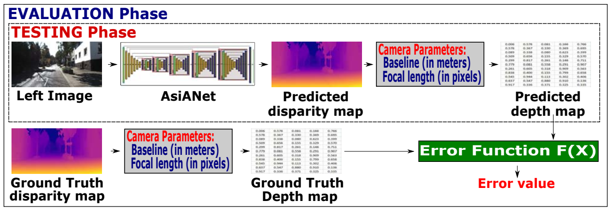

Figure 4. MDE testing and evaluation framework.

performance of the model if a ground truth disparity map of the input image is available. In other words, ground truths are only used for evaluating the performance of a trained model and are never used to scale the predictions. Fig. 4 illustrates these two phases.

4. Experiments

In this section, we describe an extensive evaluation of our proposed method using the raw sequences of the KITTI [13] dataset, which is a well-known dataset for training and evaluating unsupervised monocular depth estimation methods and used for quantitative comparisons. We also used the Make3D [32] dataset for evaluating the generalization ability of the model.

4.1. Implementation Details

The depth estimation network is implemented in TensorFlow [1] using a single GTX 1080 Ti (11GB) GPU. We used the Adam optimizer [20] for loss minimization with $B_1 = 0.9$, $B_2 = 0.999$, $\epsilon = 10^{-8}$, and a mini-batch size of 4. The learning rate is $\lambda = 0.0001$ for the first 30 epochs, and afterward, reduced by half for every ten (10) epochs. Our Charbonnier loss function is a weighted sum of the appearance dissimilarity term, the disparity smoothness term, and the left-right disparity consistency term. We set the weighting of the different terms of the loss function to $\alpha = 1$, $\beta = \frac{0.1}{r}$, and $\gamma = 1$, where r is the scaling factor $(1, 2, 4, 8)$ of the corresponding disparity map output. For training, we resized the stereo image pairs to 256×512 to improve on the training efficiency, but at testing time, the network can generate depth maps for single images of arbitrary dimension. We use the same train/test split as in [9], which is referred to as the Eigen split. The Eigen split has $22,600$ stereo image pairs in the training set, and 697 images in the test set. The weight parameters are initialized randomly according to the Xavier initialization [14]. We also applied L_1 regularization on all the weight parameters to avoid overfitting by adding a small constant, 0.00001. Moreover, for the training set, we performed data augmentation on the fly similar to [15] and trained the AsiANet architecture using the augmented training set, which took nearly six (6) days to complete 50 epochs. However, a trained AsiANet only takes approximately 160 milliseconds to generate a 256×512 disparity map on a single test image with the same resolution.

4.2. Results and Discussion

We present several experimental results demonstrating the effectiveness of AsiANet in monocular depth estimation using the unsupervised approach. To compare the performance of our proposed approach with existing methods on the KITTI dataset, we provide the quantitative results of our models in Table 4 using the same performance metrics as in [9, 12, 15], which are indicative of the accuracy of the model in depth prediction. This table shows that the proposed unsupervised monocular depth estimation framework achieves accurate depth predictions and outperforms existing methods. The experimental evaluation demonstrates clear advantages of the proposed method over competing works on the KITTI benchmark. The proposed method works because AsiANet benefits from the greater expressive power of stacked autoencoders [29] to effectively process and combine features across scales, from fractional max-pooling's ability to encode features more effectively [16] and from the robustness of the Charbonnier function [2].

However, the depth range affects the depth prediction accuracy wherein a smaller range yields better performance. In other words, the closer an object is to the camera, the larger the disparity while the farther away an object is, the smaller is the disparity. Our results show that AsiANet works better for close-range depth (1 to 50 meters) and is very effective in minimizing larger disparity because AsiANet trained using a Charbonnier loss function can process features across multiple scales and combines these features more effectively to best capture the various spatial relationships which is useful in determining the depth boundaries. On the other hand, increasing the depth range causes more features to move away from the camera which leads

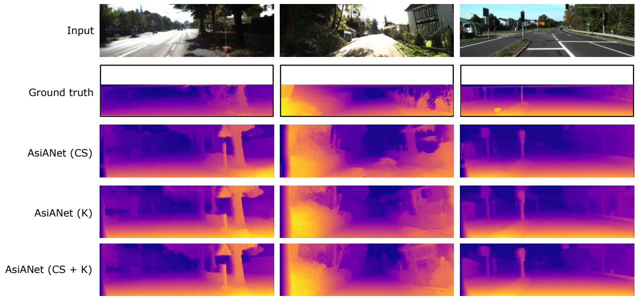

Figure 5. Qualitative results on the KITTI test set of the different AsiANet models. The AsiANet models trained using the Cityscapes and KITTI datasets, respectively, can capture the essential scene structures but our final model that was pre-trained on the Cityscapes dataset and fine-tuned on the KITTI dataset produces depth maps with finer details than the other two models. The sparse ground truth depth maps are interpolated for visualization purpose.

Table 4. Monocular depth estimation results on the KITTI test set. For training, K means trained on the KITTI dataset, CS means trained on the Cityscapes dataset, and CS+K means pre-trained on CS and fine-tuned on K. The results of the other models are taken directly from published works. * means the model was trained using the supervised learning method. The red and **bold** values indicate the best results.

Method	Depth Range	Training Dataset	Error Metric (Lower is better)				Accuracy Metric (Higher is better)		
			ARD	*SRD*	*RMSE (linear)*	*RMSE (log)*	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen *et al.* [9] Coarse*	0-80 m	K	0.194	1.531	7.216	0.273	0.679	0.897	0.967
Eigen *et al.* [9] Coarse+Fine*	0-80 m	K	0.190	1.515	7.156	0.270	0.692	0.899	0.967
DDVO [36]	0-80 m	K	0.151	1.257	**5.583**	**0.228**	0.810	**0.936**	**0.974**
GeoNet [44]	0-80 m	K	0.155	1.296	5.857	0.233	0.793	0.931	0.973
Godard *et al.* [15]	0-80 m	K	0.148	1.344	5.927	0.247	0.803	0.922	0.964
LEGO [41]	0-80 m	K	0.162	1.352	6.276	0.252	-	-	-
Mahjourian *et al.* [28]	0-80 m	K	0.163	**1.240**	6.220	0.250	0.762	0.916	0.968
Yang *et al.* [42]	0-80 m	K	0.182	1.481	6.501	0.267	0.725	0.906	0.963
Zhou *et al.* [46]	0-80 m	K	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Kuznietsov *et al.* [23] *unsupervised only*	0-80 m	K	0.308	9.367	8.700	0.367	0.752	0.904	0.952
Ours (AsiANet)	0-80 m	K	**0.145**	1.349	5.909	0.230	**0.824**	**0.936**	0.970
Zhou *et al.* [46]	0-80 m	CS	0.267	2.686	7.580	0.334	0.577	0.840	0.937
Ours (AsiANet)	0-80 m	CS	**0.162**	**1.281**	**5.750**	**0.242**	**0.784**	**0.934**	**0.971**
DDVO [36]	0-80 m	CS + K	0.148	1.187	5.496	0.226	0.812	0.938	**0.975**
GeoNet [44]	0-80 m	CS + K	0.153	1.328	5.737	0.232	0.802	0.934	0.972
Godard *et al.* [15]	0-80 m	CS + K	**0.124**	**1.076**	**5.311**	0.219	0.847	0.942	0.973
LEGO [41]	0-80 m	CS + K	0.159	1.345	6.254	0.247	-	-	-
Mahjourian *et al.* [28]	0-80 m	CS + K	0.159	1.231	5.912	0.243	0.784	0.923	0.970
Zhou *et al.* [46]	0-80 m	CS + K	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Ours (AsiANet)	0-80 m	CS + K	0.128	1.161	5.470	**0.213**	**0.858**	**0.947**	0.974
Garg *et al.* [12] L12 Aug 8x	1-50 m	K	0.169	1.080	5.104	0.273	0.740	0.904	0.962
GeoNet [44]	1-50 m	K	0.147	0.936	4.348	0.218	0.810	0.941	0.977
Godard *et al.* [15]	1-50 m	K	0.140	0.976	4.471	0.232	0.818	0.931	0.969
Mahjourian *et al.* [28]	1-50 m	K	0.155	0.927	4.549	0.231	0.781	0.931	0.975
Zhou *et al.* [46]	1-50 m	K	0.201	1.391	5.181	0.264	0.696	0.900	0.966
Kuznietsov *et al.* [23] *unsupervised only*	1-50 m	K	0.262	4.537	6.182	0.338	0.768	0.912	0.955
Ours (AsiANet)	1-50 m	K	**0.122**	**0.786**	**4.014**	**0.198**	**0.864**	**0.953**	**0.978**
Zhou *et al.* [46]	1-50 m	CS	0.260	2.232	6.148	0.321	0.590	0.852	0.945
Ours (AsiANet)	1-50 m	CS	**0.144**	**0.802**	**4.252**	**0.216**	**0.813**	**0.949**	**0.979**
Godard *et al.* [15]	1-50 m	CS + K	0.117	0.762	3.972	0.206	0.860	0.948	0.976
Mahjourian *et al.* [28]	1-50 m	CS + K	0.151	0.949	4.383	0.227	0.802	0.935	0.974
Zhou *et al.* [46]	1-50 m	CS + K	0.190	1.436	4.975	0.258	0.735	0.915	0.968
Ours (AsiANet)	1-50 m	CS + K	**0.107**	**0.663**	**3.717**	**0.184**	**0.893**	**0.960**	**0.981**

to a smaller disparity among these features, resulting in smaller variations in the performance of the various models.

Fig. 5 shows sample predictions made by the three AsiANet models at a depth range of 0 to 80 meters. The qualitative results reveal that the AsiANet architecture can effectively recover scene structures although the amount of details recovered depends on the dataset used during training. The depth maps reveal that the sky pixels are correctly assigned with the maximum depth, object boundaries are accurately captured, the foreground objects such as traffic signs, traffic poles, and tree trunks are distinguishable, and objects close to the camera are quite sharp.

Similar to previous studies, we also experimented by pre-training the network on the Cityscapes [7] dataset and then fine-tuning on KITTI, and this setup yielded better results than training on the KITTI (K) dataset or Cityscapes (CS) dataset only. Fig. 6 and 7 show sample predictions generated by our final (CS+K) model at a depth range of 0 to 80 meters. Qualitative results show that a trained AsiANet architecture reconstructs sharp and more detailed predictions yielding more consistent depth maps, that is, AsiANet can preserve depth boundaries and capture distant objects better including traffic poles, street signs, and tree trunks.

Nevertheless, the results in the final model indicate that the results could still be improved if trained on a much larger but similar training data.

To quantify the impact on the performance of the different design choices and validate our model design we performed an ablation study, and the results are shown in Table 5. We strictly adhered to the conventional way of performing an ablation study which is by removing or replacing each major component from the full model and then evaluating this model on the benchmark dataset such as in [23, 28, 42]. In our ablation study, we replaced each major component (pooling, loss) of our full model and evaluated each model on the KITTI dataset. We did not evaluate the different terms of the loss function since it is not part of our objectives to determine the best combination of terms for the loss function. Rather, we want to show that the model's performance can be improved by applying the Charbonnier function instead of L1 or L2 to the terms in the loss function. It can be seen that replacing any component of our AsiANet model increases the error values, resulting in less accurate predictions.

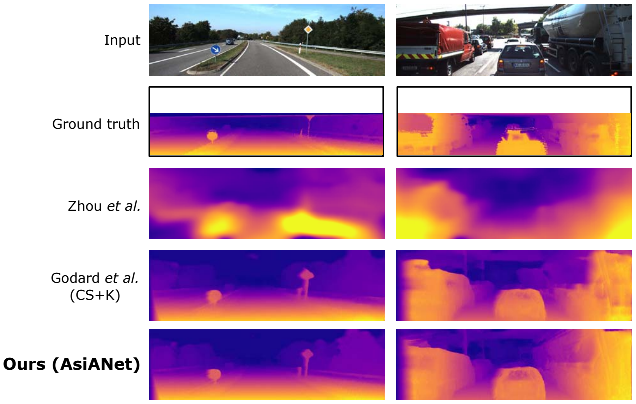

Figure 6. Qualitative results on the KITTI test set and the failure cases of Zhou *et al.* [46]. Our final model can recover scene structures with clearer object boundaries (highway guard rails and a bridge) even on textureless regions such as vast open areas, and objects that are in front of the camera are more detailed. The sparse ground truth depth maps are interpolated for visualization purpose. Objects with high intensity values are close to the camera.

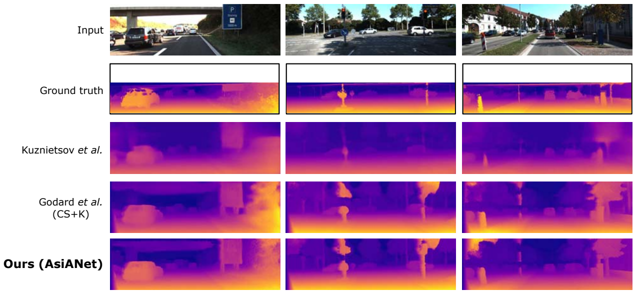

Figure 7. Qualitative results on the KITTI test set and the failure cases of Kuznietsov *et al.* [23]. Our final model captures more details and achieves better object reconstruction even for structures that have few image cues such as bridges. The sparse ground truth depth maps are interpolated for visualization purpose.

Furthermore, to demonstrate the generalization ability of AsiANet we performed a quantitative evaluation on the Make3D dataset which is unseen during training. The scenes in this dataset are very diverse unlike in the KITTI dataset because these scenes were not captured from a camera installed on a vehicle. Due to the difference in camera aspect ratio of the Make3D dataset we evaluated our proposed method on the central crop of the test images like in the previous studies. As shown in Table 6, our AsiANet model performs better than the other unsupervised depth estimation methods. Moreover, as shown in Fig. 8, our trained model can still recover global scene structures reasonably well even without any training on the Make3D images, and these predictions are quite sharp despite the differences in image data and camera parameters although there are border artifacts on the left side of these predicted depth maps.

From the experiments, we make several observations. First, the exponential linear unit (ELU) is an effective nonlinear activation function for AsiANet, a very deep network with approximately 233 million parameters, since the vanishing gradient problem did not occur during training. Second, using two similar datasets for pre-training and fine-tuning, such as the Cityscapes dataset (urban scenes) and the KITTI dataset (driving), helps improve performance since the CS+K model outperformed both the K and CS models. Third, the AsiANet model trained with the Charbonnier loss function can generate high-quality depth maps

Table 5. Ablation experiment results on the KITTI test set with depth range of 0-80 meters. The red and **bold** values indicate the best results. MP: Max-pooling layer; IFMP: Inception-like fractional max-pooling module.

Method			Error Metric (Lower is better)			
	Pooling	Loss	ARD	SRD	RMSE (linear)	RMSE (log)
AsiANet-L1	IFMP	L1	0.148	1.417	5.981	**0.230**
AsiANet-L2	IFMP	L2	0.152	1.497	6.071	0.236
AsiANet-MP	MP	Charbonnier	0.153	1.522	6.114	0.236
AsiANet	IFMP	Charbonnier	**0.145**	**1.349**	**5.909**	**0.230**

Table 6. Quantitative results on the Make3D test set. Errors are only computed based on a depth range of 0-70 meters in a central image crop. The red and **bold** values indicate the best results.

Method	Training Dataset	ARD	SRD	RMSE (linear)	RMSE (log)
Godard *et al.* [15]	CS	0.535	11.990	11.513	**0.156**
LEGO [41]	CS	0.352	7.731	**7.194**	0.346
Ours (AsiANet)	CS	**0.323**	**3.212**	8.312	0.435
Kuznietsov *et al.* [23]	K	0.421	-	8.237	**0.190**
DDVO [36]	K	0.387	4.720	**8.090**	0.204
Zhou *et al.* [46]	CS+K	0.383	5.321	10.470	0.478
Ours (AsiANet)	K	0.402	4.772	8.696	0.451
Ours (AsiANet)	CS+K	**0.367**	**4.383**	8.193	0.418

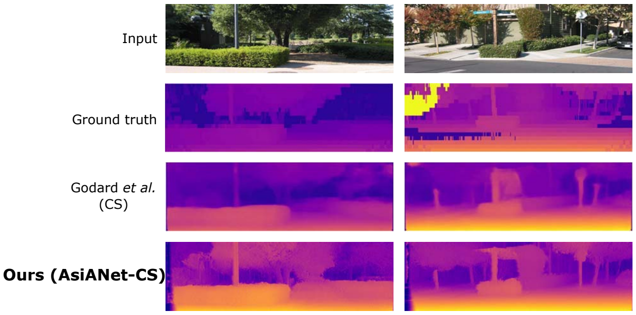

Figure 8. Qualitative results of AsiANet (trained on CS) on images from the Make3D test set.

with better global scene layout, more detailed structures, and clearer depth boundaries. Lastly, occluded areas on the left borders of a depth map are inaccurately assigned with the maximum depth due to insufficient information.

5. Conclusions

In this work, we proposed to address the problem of monocular depth estimation in an unsupervised setting. In essence, the key features of our proposed framework are as follows: a decoder structure with multi-scale cascaded autoencoders (decoder = upsampling + autoencoding), an Inception-like pooling module based on fractional max-pooling, and the use of the Charbonnier loss function for unsupervised monocular depth estimation. Ground truth depth labels are not required since our network only needs stereo image pairs during training. Experiments conducted on the KITTI dataset show that AsiANet is effective for monocular depth estimation as it can infer depth maps with fine details even for distant objects. Our proposed approach outperforms existing unsupervised monocular depth estimation methods. AsiANet is also able to generalize well to a similar dataset, the Make3D dataset, as it can generate visually plausible depth maps.

6. Acknowledgements

This work has been supported by the Department of Science and Technology-Engineering Research and Development for Technology Program.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, and M. D. et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv 1603.04467*, 2016.

[2] J. Barron. A more general robust loss function. *arXiv 1701.03077*, 2017.

[3] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/kanada meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[4] C. Cadena, Y. Latif, and I. D. Reid. Measuring the performance of single image depth estimation methods. *In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 35:4150–4157, 2016.

[5] Y. Cao, Z. Wu, and C. Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[6] D.-A. Clevert, T. Unterthiner, , and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *In: International Conference on Learning Representations (ICLR)*, 2016.

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. *In: CVPR*, pages 3213–3223, 2016.

[8] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *In: ICCV*, pages 2650–2658, 2015.

[9] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *In: Annual Conference on Neural Information Processing Systems (NIPS)*, 2014.

[10] P. Fischer, T. Pohl, T. Köhler, A. Maier, and J. Hornegger. A robust probabilistic model for motion layer separation in x-ray fluoroscopy. *Information Processing in Medical Imaging*, pages 288–299, 2015.

[11] H. Fu, M. Gong, C. Wang, and D. Tao. A compromise principle in deep monocular depth estimation. *arXiv 1708.08267*, 2017.

[12] R. Garg, V. Kumar, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. *In: ECCV*, pages 740–756, 2016.

[13] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. *In: CVPR*, 2012.

[14] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *In: Artificial Intelligence and Statistics Conference (AISTATS)*, 9:249–256, 2010.

[15] C. Godard, O. M. Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *In: CVPR*, 2017.

[16] B. Graham. Fractional max-pooling. *arXiv 1412.6071*, 2014.

[17] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, and G. Wang. Recent advances in convolutional neural networks. *Pattern Recognition*, 2017.

[18] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. *In: Annual Conference on Neural Information Processing Systems (NIPS)*, pages 2017–2025, 2015.

[19] K. Karsch, C. Liu, and S. B. Kang. Depth extraction from video using non-parametric sampling. *In: ECCV*, pages 775–788, 2012.

[20] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *In: International Conference on Learning Representations (ICLR)*, 2015.

[21] J. Konrad, M. Wang, and P. Ishwar. 2d-to-3d image conversion by learning depth from examples. *In: CVPRW*, pages 16–22, 2012.

[22] G. Kordelas, D. Alexiadis, P. Daras, and E. Izquierdo. Enhanced disparity estimation in stereo images. *Image and Vision Computing*, 35:31–49, 2015.

[23] Y. Kuznietsov, J. Stuckler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. *In: CVPR*, pages 2215–2223, 2017.

[24] W. Lai, J. Huang, N. Ahuja, and M. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. *In: CVPR*, 2017.

[25] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. *In: Fourth International Conference on 3D Vision (3DV)*, pages 239–248, 2016.

[26] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. *In: CVPR*, 2015.

[27] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. *In: CVPR*, pages 716–723, 2014.

[28] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. *In: CVPR*, 2018.

[29] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. *In: ECCV*, 2016.

[30] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. *In: ICCV*, 2015.

[31] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. *In: Annual Conference on Neural Information Processing Systems (NIPS)*, 2005.

[32] A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *International Journal of Computer Vision*, 76(1):53–69, 2008.

[33] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. *In: ECCV*, 2012.

[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *In: CVPR*, pages 2818–2826, 2016.

[35] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. DeMoN: Depth and motion network for learning monocular stereo. *In: CVPR*, 2017.

[36] C. Wang, J. Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. *In: CVPR*, 2018.

[37] K. Wang, E. Dunn, J. Tighe, and J. Frahm. Combining semantic scene priors and haze removal for single image depth estimation. *In: IEEE Winter Conference on Applications of Computer Vision*, pages 800–807, 2014.

[38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error measurement to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[39] J. Xie, R. Girshick, and A. Farhadi. Deep3d: fully automatic 2d-to-3d video conversion with deep convolutional neural networks. *In: ECCV*, 2016.

[40] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. *In: CVPR*, 2017.

[41] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia. LEGO: Learning edge with geometry all at once by watching videos. *In: CVPR*, 2018.

[42] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia. Unsupervised learning of geometry with edge-aware depth-normal consistency. *arXiv 1711.03665*, 2017.

[43] M. Ye, E. Johns, A. Handa, L. Zhang, P. Pratt, and G. Yang. Self-supervised siamese learning on stereo image pairs for depth estimation in robotic surgery. *In: Hamlyn Symposium on Medical Robotics*, 2017.

[44] Z. Yin and J. Shi. GeoNet: Unsupervised learning of dense, optical flow and camera pose. *In: CVPR*, 2018.

[45] Y. Zhang, R. Ji, X. Fan, Y. Wang, F. Guo, Y. Gao, and D. Zhao. Search-based depth estimation via coupled dictionary learning with large-margin structure inference. *In: ECCV*, pages 858–874, 2016.

[46] T. Zhou, M. Brown, N. Snavely, and G. Lowe. Unsupervised learning of depth and ego-motion from video. *In: CVPR*, 2017.

[47] L. Zhu, X. Wang, D. Wang, and H. Wang. Single image depth estimation based on convolutional neural network and sparse connected conditional random field. *Optical Engineering*, 55(10):03101, 2016.