EFFICIENT ATTENTION GUIDED GAN ARCHITECTURE FOR UNSUPERVISED DEPTH ESTIMATION

by

Sumanta Bhattacharyya

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Electrical and Computer Engineering

Charlotte

2019

Approved by:

____

Dr. Chen Chen

____

Stephen Welch

____

Dr. Arindam Mukherjee

ABSTRACT

SUMANTA BHATTACHARYYA. Efficient Attention guided GAN architecture for unsupervised depth estimation. (Under the direction of DR. CHEN CHEN)

Single image depth estimation has always been a key interest in computer vision community. Although depth estimation from a single monocular image still is an ill-posed problem, stereo images are in rescue. Deep-learning based approaches to depth estimation are rapidly advancing, offering better performance over traditional computer vision approaches across many domains. However, for many critical applications, cutting-edge deep-learning based approaches require too much computational overhead to be operationally feasible. This is especially true for depth-estimations methods that leverage adversarial learning, such as Generative Adversarial Networks(GANs). I propose a computationally efficient GAN for unsupervised monocular depth estimation using factorized convolutions and an attention mechanism. Specifically, I leverage the Extremely Efficient Spatial Pyramid of Depth-wise Dilated Separable Convolutions(EESP) module of ESPNetv2 inside the network, leading to a total reduction of 25.6%, 33.82% and 31% in the number of model parameters, FLOPs and inference time respectively, as compared to the previous unsupervised GAN approach. Finally, I propose a context-aware attention architecture to generate detail-oriented depth images. I demonstrate the performance of our proposed model on two benchmark datasets, KITTI and Cityscapes.

ACKNOWLEDGEMENTS

I would like to present my gratitude to my advisor Dr. Chen Chen. He taught me how to do research, how to write a good thesis, and how to give a great presentation. He shoId extraordinary tolerance to my organized technical writing and he gave the greatest support he could to my study. I would also like to acknowledge Dr. Arindam Mukherjee of Electrical and Computer engineering Department at UNC, Charlotte and Stephen Welch of Computer Science Department at UNC, Charlotte as my co-advisor and I am gratefully indebted to there help and valuable comments on this thesis. My life in UNCC is enjoyable and fruitful because of you. Finally, I would like to thank my former advisor Dr. Andrew Willis, who invoked my interest into research and academics on my early days at UNCC.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Depth estimation is not an easy task for computational model like human. Human can exploit monocular signs such as texture, perspective, occlusion and object sizes. These cues are extremely important for a scene understanding problem. Recovering depth from an image without any prior information like optical flow or stereo images is very essential.

1.1 Motivation

Image-based depth estimation is a key problem in computer vision, with a wide range of applications from robotic navigation and virtual reality. Early applications of deep learning to depth estimation relied on supervised learning, directly regressing a depth estimate of each pixel, and training models using ground-truth depth maps. Eigen [4] demonstrated good performance using a multi-scale convolution neural network (CNN) to predict depth from single images. Probabilistic graphical models such as Conditional Random Field [5] have increased the performance when they are used in neural networks for optimization. In order to learn the pixel-wise transformation, supervised approaches require ground truth depth data for training. However, obtaining the ground truth depth data is non-trivial, and model performance may be limited by the amount of quality ground truth data that can be collected.

Unsupervised approaches estimate disparity maps from two different image views (rectified left and right image) of calibrated stereo camera. Remarkably, making ground truth depth data not required for training. This makes the unsupervised approaches more attractive and practical in practice. Godard [6] proposed a left-right cycle consistency loss as a constraint on this unsupervised approach. Pilzer [1] applied the left-right consistency in an adversarial learning approach in order to improve the generated images. Although adversarial learning-based unsupervised methods achieved excellent performance in depth estimation, these methods rely on the complex generative adversarial network (GAN) architecture which is generally computationally heavy. As a result, they are not able to run on real-time in resource constrained edge devices for practical applications, autonomous driving, flying car.

1.2 Objective

Single image depth estimation mostly considered as a pixel level continuous regression problem. Generally It requires to learn a non linear mapping between raw image pixels and their corresponding depth maps from an artificial neural network perspective. The only disadvantage of this learning is it demands a massive amount of ground truth depth values. Exploiting epipolar constraints [6], can give us an indirect way of depth map learning without any ground truth supervision. In a traditional computer vision method, for a calibrated stereo camera, depth map $\mathbf{D}$ can be estimated by $D = \frac{b \times f}{d}$, where b is the distance between the two cameras f is the focal length and d is the disparity map. As an unsupervised learning approach estimating disparity map from a pair of calibrated stereo images through image synthesis is an indirect way to learn depth without requiring ground truth. Adversarial learning is useful when it comes to image generation task but its only constraint is it's heavy architecture. I found convolution factorization and attention mechanism is a key to the solution.

1.3 Contribution

To address this challenge, I propose a computationally efficient model for depth estimation given stereo image pairs, based on the unsupervised GAN framework [1]. A context-aware attention mechanism is also introduced to improve depth estimation, yielding more accurate overall depth prediction. In summary, the main contributions of this thesis are:

1. I adopt the EESP module [7] inside a GAN architecture to significantly improve the computation efficiency, while concurrently reducing RMSE by 7% compared to the baseline model [1].

2. I introduce a context aware attention layer in the generator to get accurate depth images. To the best of our knowledge, our work is the first to explore the attention mechanism in the unsupervised depth estimation approach.

3. I conduct extensive experiments on the publicly available datasets KITTI and CityScapes. The results demonstrate both the efficiency and effectiveness of the proposed method. A detailed ablation study is also carried out to identify the relative contributions of the individual components.

1.4 Organization

In the following chapters, I discussed extensively on methodologies, various architectures, experimental setup, the result I achieved along with the quantitative and qualitative comparison with the state of the art methods.

CHAPTER 2: LITERATURE SURVEY and METHODOLOGIES

2.1 Literature Survey

Depth estimation helps to understand the 3D structure of a 2D image scene. In traditional computer vision, depth estimation algorithms rely on point correspondences between stereo image pairs [8, 9] and triangulation. Saxena [10] showed us how depth can be learned from handcrafted features using monocular cues of a single 2D image. Over a period of time, a lot of approaches have evolved using handcrafted feature representations [11, 12, 13, 14]. Although, with the rapid development of deep learning algorithms, we are no longer required to select handcrafted features, instead I can build end to end model for depth estimation which learn the mapping from raw pixel values. Here I evaluated models that take single input image and predicts the depth of the image.

Supervised Depth Estimation: Supervised learning relies on the ground truth depth data to achieve promising performance for image depth estimation. Indoor datasets like NYU [15] and outdoor datasets like KITTI [3] and Cityscapes [16] contribute to the evolution of supervised monocular depth estimation approaches. Eigen [4] proposed to have a two scale network to generate dense depth map trained on ground truth values. Probabilistic graphical models (MRF, CRF) also contributed in association with deep networks to boost accuracy [17]. Xu [5] offered a structured attention mechanism using CRF to combine multi-scale information obtained from the CNN layers. Although supervised Depth estimation task has been formulated as a regression problem, Cao first proposed [18] to accomplish this task as a pixelwise classification problem. Recent architectures have shown a promising development for multi-task learning strategies [19, 20] along with depth estimation. Although these approaches turn out to be very sophisticated, they highly rely on ground truth depth for training. Unlike earlier methods, our depth estimation does not require ground truth depth value while training.

Unsupervised Depth Estimation: Recently unsupervised depth estimation algorithms [21, 22] have emerged, given the advantage of not requiring ground truth depth data. Garg [23] proposed an unsupervised approach using stereo image pair. However, They perform Taylor approximation for linearization of loss as their model is not fully differentiable, it turns out to be hard to optimize. Succeeding works show how this can be solved using bilinear warping [24]. In a recent work, Godard [6] proposed a new left right cycle consistency loss along with image reconstruction loss for a better quality depth estimation. This new training loss for depth estimation has also been used in GAN architecture by Pilzer [1], a Cycle-GAN approach for unsupervised depth estimation based on stereo disparity estimation. In our work, I take this architecture as our baseline to implement further works. Recent efforts also show how joint learning strategy can be applied to unsupervised fashion [25, 26]. Unlike lightweght network, it is difficult for GAN [27] architectures to achieve real-time output on resource constrained embedded devices. Our work attempts to make it efficient and accurate than the existing approaches. To the best of our knowledge, I are the first one to explore the ideas of GAN efficiency.

Adversarial learning: Adversarial learning has been proven to be efficient in the image generation task. Recent works have demonstrated how various GAN architectures [27] can be utilized in dense depth map generation. Recent approach by Kundu [28] shoId how domain adaptation strategy for depth estimation can be utilized in adversarial learning fashion. Various GAN architectures like Conditional GAN framework [29], CycleGAN [30] also have been explored for depth estimation task [14]. Efforts on joint learning strategy [31, 32] in GANs also demonstrated depth estimation of high quality. In our work, I focused on building cost effective GAN architecture, which is significantly different than the previous works.

Attention: Attention models are very useful in computer vision for improving the performance in pixel-level prediction tasks as Ill as in the context of monocular depth prediction. A depth map has to to be accurate and detail oriented, so preserving details through attention layers will be helpful for 3D reconstruction as Ill. There are very few works in the literature enlightening the benefits of attention layer but only limited to supervised depth estimation approaches. Hao [33] demonstrated how attention mechanism can focus on the most informative part of the input image based on the context. Recent work also shoId attention as aggregation of image and pixel level information [14]. *The approach I present in this thesis is different from the prior works. First, ours is the first work to explore the attention mechanism in the* ***unsupervised*** *depth estimation problem. In addition, I leverage the advantage of multiscale feature fusion (local and global) to obtain attention aware features.*

2.2 GAN Architecture

Generative adversarial network [27] is a deep learning architecture for adversarial learning process containing two neural network, generator and discriminator. Generator generates new data instances and discriminator determines their validity until generator fools the discriminator (minimax game). Generator architecture is an encoder-decoder network and discriminator architecture is a single CNN network. Training of a GAN is very notorious as two networks must posses similar skill level. When the discriminator is trained, generator value will be constant, and When the generator is trained, discriminator value will be constant. Objective function for adversarial learning is following:

$$\mathbf{L_{GAN}} = \tfrac{1}{m}\textstyle\sum_{i=1}^{m} \log(D(x_i)) + \log(1 - D(G(z_i)) \tag{2.1}$$

z is noise vector, $G(z_i)$ is generator's output, x is training sample, $D(x_i)$ is discriminator's output for real training sample, $D(G(z_i))$ is discriminator's output for generated training sample.

Each network should train when the other is static. It gives generator a better understanding of the gradient it should learn. Each network of GAN can outperform the other one, if the generator is good enough than the discriminator, it can exploit the fakeness of the discriminator resulting in false negatives, in reverse, if the discriminator is strong enough, generator will not be able to understand the gradient carefully. These inconsistencies will lead to Convergence issues, i.e., mode collapse (generator collapses and fails to produce diverse variety of samples as in training dataset), oscillating gradient .

Spectral normalization [34] is an unique weight normalization process for stable training of GAN discriminator (prevents mode collapse). It controls the Lipschitz constant of the convolutional filters in discriminator network by constraining the spectral norm of each layer.

In order to get rid of slow learning and oscillating gradient, **two-time-update-rule (TTUR)** where generator and discriminator learns in two different learning rates. generally discriminator learns with around four times faster learning rate than generator, as generator needs to learn the gradient slowly and fool the discriminator.

2.3 Stereo Matching

Stereo Matching is an important subclass of computer vision involves corresponding pixels finding in rectified stereo images. Stereo images involve two cameras with little distance between them in order to capture two different viewpoints of the same scene, this setup is equivalent to human eyes in order to perceive depth.

As shown in Figure 2.1, rectified stereo images work by finding corresponding pixels in both images along with the same epipolar plane. The epipolar lines coincide with the horizontal scanlines if the cameras are parallel, the corresponding points in the two images should hence lie on the same horizontal scanline. Such stereo arrangements decrease the search for correspondences from $2D$ to $1D$.

Disparity refers to the distance between two corresponding pixels. While perform-

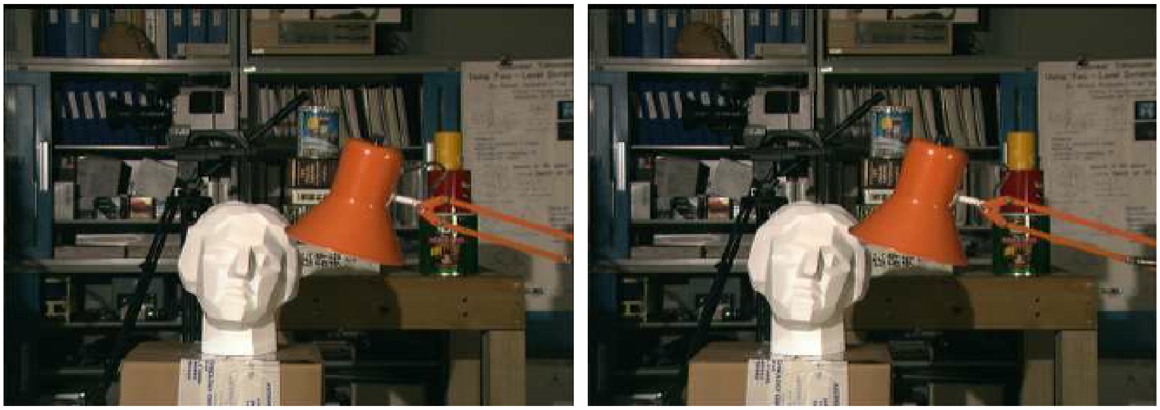

Figure 2.1: Sample stereo image pair, left side is left image and right side is right image.

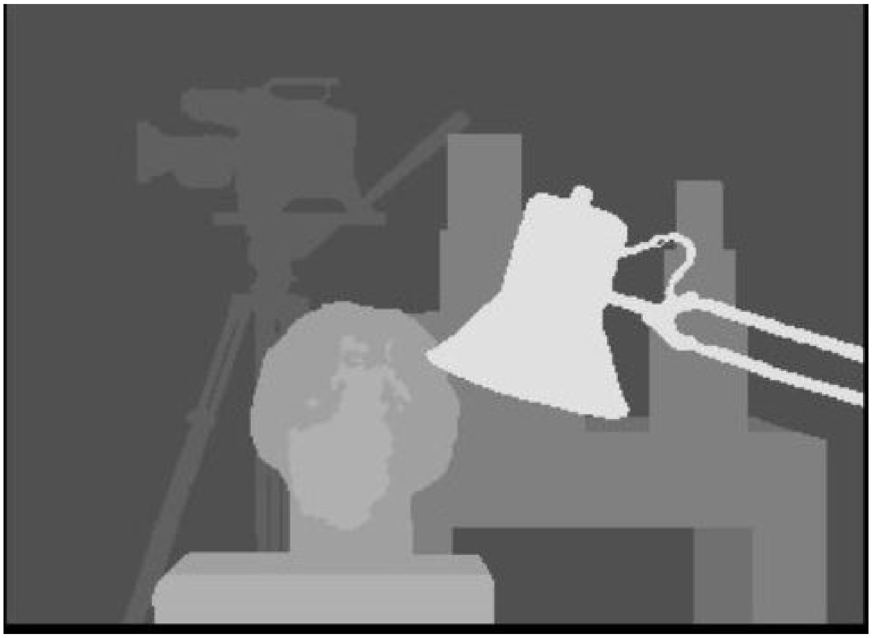

Figure 2.2: disparity map

ing this matching process for every pixel in the left hand image with the right hand image, computing distance between them end up with an image containing distance values of each pixel in the left image (Figure 2.2). Depth estimation can directly be done from this disparity map (d) by using calibrated camera parameters (baseline(b), focal length(f)), depth map $D = \frac{b \times f}{d}$.

2.4 Convolution Operation

In mathematical terminology, convolution is calculated from two given functions through integration which expresses how the shape of one function is modified by the other function in 1 dimension convolution.

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) * g(t - \tau) d\tau \tag{2.2}$$

The 2D convolution is a simple operation as well. It involves a kernel, denotes nothing but a small matrix of weights. This kernel slides over the 2 dimensional input data(generally image), executing an element-wise multiplication with the part of the input it is currently on, resulting in summing up the output into a single pixel. It repeats for every location it slides over, resulting a two dimensional matrix (features) into another two dimensional matrix (convolution output). These features are the weighted sums of the input features.

Convolution has multiple benefits-

Function smoothing: Instead of smoothing by the observed distribution of data points for a given function, it involves general approach, allowing to smooth the given function as it takes kernel function and at each point in the integral it puts a copy of kernel function multiplied by the value of function at that point.

Localized information aggregation: Convolutions can be interpreted as computing a weighted sum of the values of function in a way where the contribution from each data point is determined by its distance between the data point and output point.

Pattern matchers: Discrete convolution is a dot product between the filter weights and the values of the filter. Dot products indicate similarity of two vectors. Output-centered convolution involves a vector of weights along with a vector of input values multiplying and summing aligning entries equivalent to dot product calculation. It is true that one can enhance the value of a dot product by increasing the magnitude of vectors, but for a constant magnitude, the maximal value of the dot product is achieved when the vectors are directing towards a single direction.

2.5 Convolution in neural network

The discovery of deep learning frameworks has made it possible to implement convolution layers into a deep learning model in a very convenient way. Convolution Neural Network (CNN) has been a great example for that. As shown in Figure 2.3, a basic CNN architecture has been discussed. It consists of convolution (along with padding and stride), max pooling, fully connected layer, activation function etc.

Convolution: As I discussed earlier in 2D convolution, a filter slides over the image (two dimensional) resulting in output feature map. This has certain parameters to tune like padding and stride.

Padding: It adds on the the edges with extra pixels of values 0 (zero padding). So, while sliding it allows the original edge pixels to be at its center, extending into the extra pixels beyond the edge, yielding an output with same size as input.

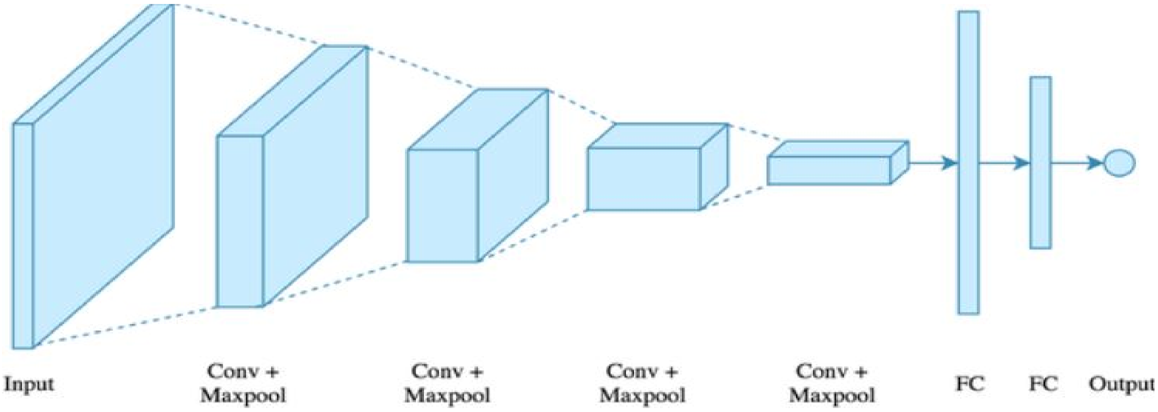

Figure 2.3: Basic CNN architecture.

Stride: The idea of the stride is to jump some of the slide locations of the kernel. A stride of n denotes pick slides n pixel apart, so stride 1 is a standard convolution. A stride of 2 downsizes the output by a factor of 2 and so on.

Maxpooling opearation: Max pooling can be interpreted as sample-based discretization process. The purpose is to down-sample an input representation, reduction of dimensionality. This helps in over-fitting due to an abstract form of the representation. It provides reduction of parameter and translation invariance to the internal representation.

Fully connected layer: The output from the convolutional layers is a representation of high-level features in the data. Fully-connected layer introduces non-linearity in the learning.

From an operational aspect,

1. Convolution and pooling mechanism breaks the image into features and analyzes them.

2. Fully connected layer utilizes the output of convolution/pooling to predict the image label.

2.6 Analysis of Computational Cost and Factorized Convolution

Although convolution plays a great role in feature extraction, It is a memory consuming process and ended up dealing with a lot of parameters. In order to speed up this process factorization of convolution is necessary. Here, I will discuss different types of convolution factorization.

Dilated convolution: It introduces the dilation rate parameter to convolutional layers, involves spacing between the values in a kernel. A 3x3 kernel with a dilation rate of 2 will have the same field of view as a 5x5 kernel, while only using 9 parameters. This delivers a wider field of view without any extra computation. Real-time segmentation architectures generally take advantage of it .

Depthwise separable convolution: It breaks the convolution in two ways (i) depth wise convolution- a spatial convolution performed independently over each channel of an input. (ii) pointwise convolution- a 1×1 convolution, projecting the channels output by the depthwise convolution onto a new channel space. Architectures like Xception and Mobilenet utilize this kind of convolution.

Group convolution: Group convolution was first introduced in the AlexNet paper [35]. it allows the network training in a resource constrained environment. Here, the filters are separated into different groups. Each group is responsible for a conventional 2D convolutions with certain depth.

Following table is a comparison in terms of parameters and receptive field of kernel.

Table 2.1: Comparison of different convolution operations considering a 3×3 kernel size, M input channels, N output channels, groups, general kernel size (for 3×3) for various dilation rate is $d_k = ((3-1) \times d + 1)$, d is the dilation rate.

Convolution Types	Parameters	Receptive field size
Standard convolution type	$3 \times 3 \times M \times N$	3×3
Group convolution	$\frac{3 \times 3 \times M \times N}{groups}$	3×3
Dilated convolution	$3 \times 3 \times M \times N$	$d_k \times d_k$
Depth-wise Dilated Separable Convolution	$3^2 \times M + M \times N$	$d_k \times d_k$

2.7 GAN generator and discriminator architecture

I used Resnet as a generator (encoder-decoder) architecture and CNN (as shown in Figure 2.3) as discriminator architecture.

Inception of Resnet:

Problem- When deeper networks starts converging, a new problem arises. As network depth increases, accuracy gets saturated and then goes down rapidly.

During backpropagation, when partial derivative of the error function with respect to the current weight in each iteration of training, this has the effect on computing gradients of the front layers in a multi-layer network. When the network is deep and gradients keep on getting multiplied in the reverse direction, it will become zero if the gradient is small by the time it reaches the end branch. (**vanished**). When the network is deep, and multiplying gradients of large numbers, it will become too large (**exploded**).

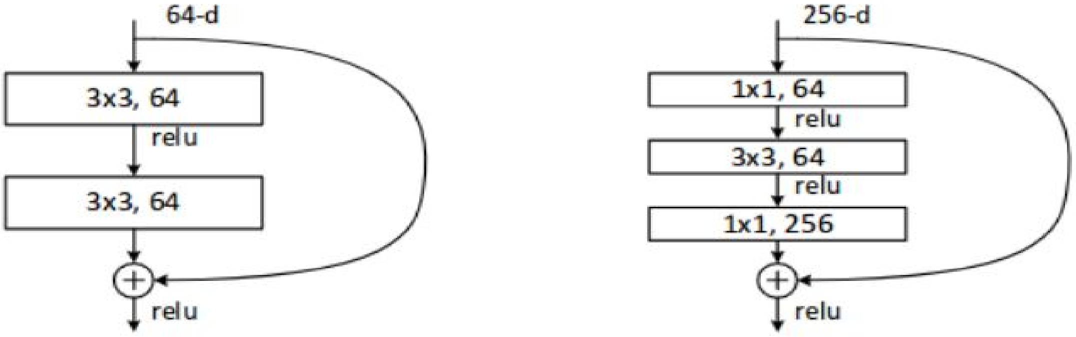

Figure 2.4: Residual block (i) normal (ii) Bottleneck

Solution- Instead of learning a direct mapping of x → y with a function H(x) (stacked non-linear layers), it defines the residual function (skip connection) using $F(x) = H(x)x$, which can be reframed into $H(x) = F(x) + x$, where $F(x)$ and x represents the stacked non-linear layers and the identity function(input=output) respectively.

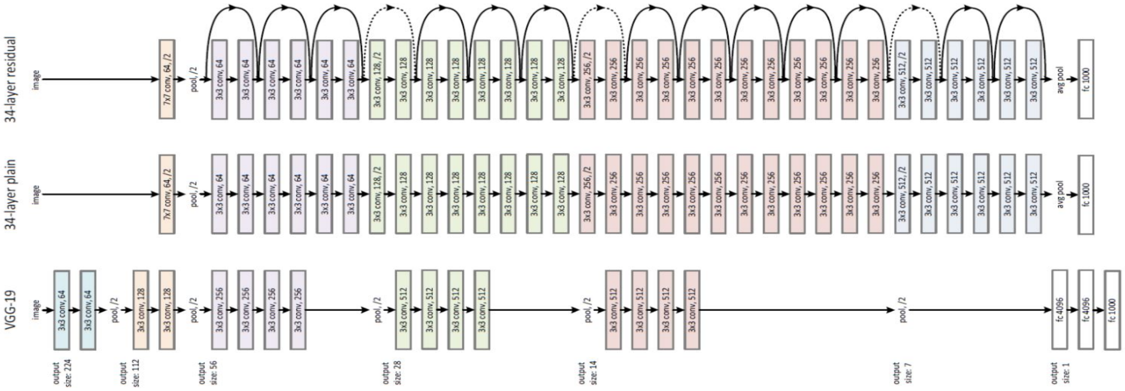

Figure 2.5: Resnet architecture

Variant of Resnet: Bottleneck architecture

The normal residual block is perfect for Resnet 18 or 34 but as many more layers added to the network for resnet50 and beyond, it is not possible to waste so much of resources on those expensive convolution operation, so I use BottleNeck blocks. A

BottleNeck block is very similar to a normal one. All it does is use a 1x1 convolution to reduce the channels of the input before performing the expensive 3x3 convolution, then using another 1x1 to project it back into the original shape.

CHAPTER 3: EXPERIMENTS

3.1 Method

This section describes our proposed method. I first introduce an efficient GAN model along with the EESP module and the proposed attention layer. I then demonstrate its accuracy over existing state of the art approaches.

3.2 Network Architecture

I follow Pilzer 's [1] work on Cycle-GAN architecture for depth estimation as the baseline of our approach. As shown in Figure 3.1, this architecture uses calibrated stereo camera images (pairwise) as input to estimate disparity map (d_m) through image synthesis. The generator network consists of two sub-networks. The upper sub-network generates a right disparity map (R_d) with the input I_l and synthesizes a right image view (I'_r) through the warping operation (pixel to pixel matching) $\mathbf{w}$, $I'_r = \mathbf{w}(R_d, I_l)$. Similarly, the loIr sub-network generates a left image view, $I'_l = \mathbf{w}(L_d, I_r)$. The reconstruction loss (L_r) is implemented between the synthesized and input images in order to optimize the generator networks:

$$L_r = \|I_r - \mathbf{w}(R_d, I_l)\| + \|I_l - \mathbf{w}(L_d, I'_r)\|. \tag{3.1}$$

The discriminator, $D1$, $D2$, is used to discriminate if the synthesized image, I'_l, I'_r, is fake or not, thus the adversarial loss can be formulated as

$$\begin{aligned} L_{GAN} = &\mathbb{E}_{I_r \sim P(I_r)}[\log D1(I_r)] \\ &+\mathbb{E}_{I_l \sim P(I_l)}[\log(1 - D1(\mathbf{w}(R_d, I_l)))] \\ &+\mathbb{E}_{I_l \sim P(I_l)}[\log D2(I_l)] \\ &+\mathbb{E}_{I_r \sim P(I_r)}[\log(1 - D2(\mathbf{w}(R_d, L_d)))]. \end{aligned} \tag{3.2}$$

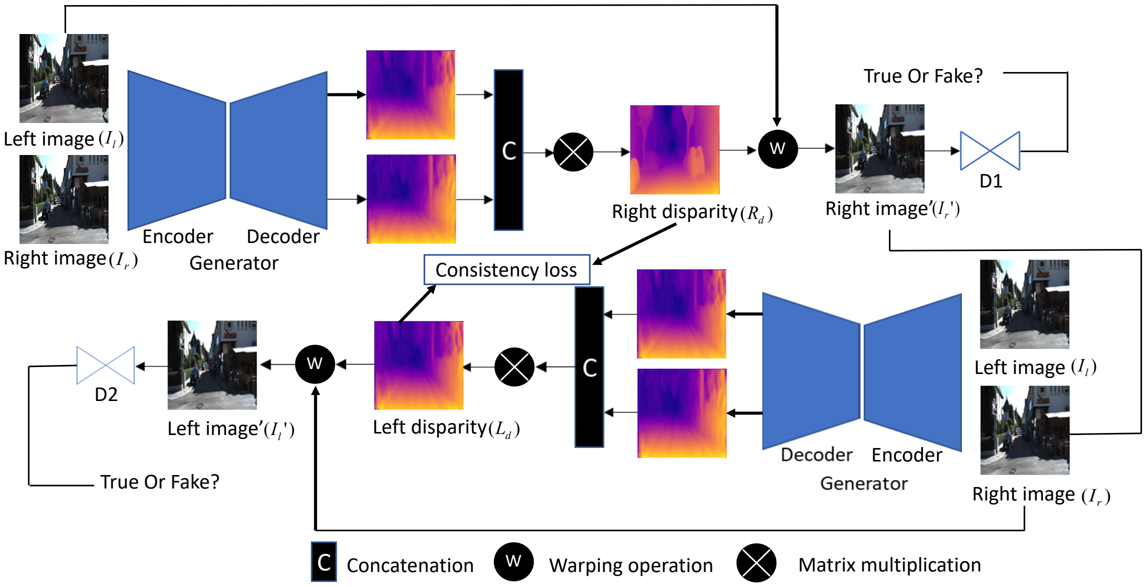

Figure 3.1: Unsupervised monocular depth estimation framework using Cycle-GAN.

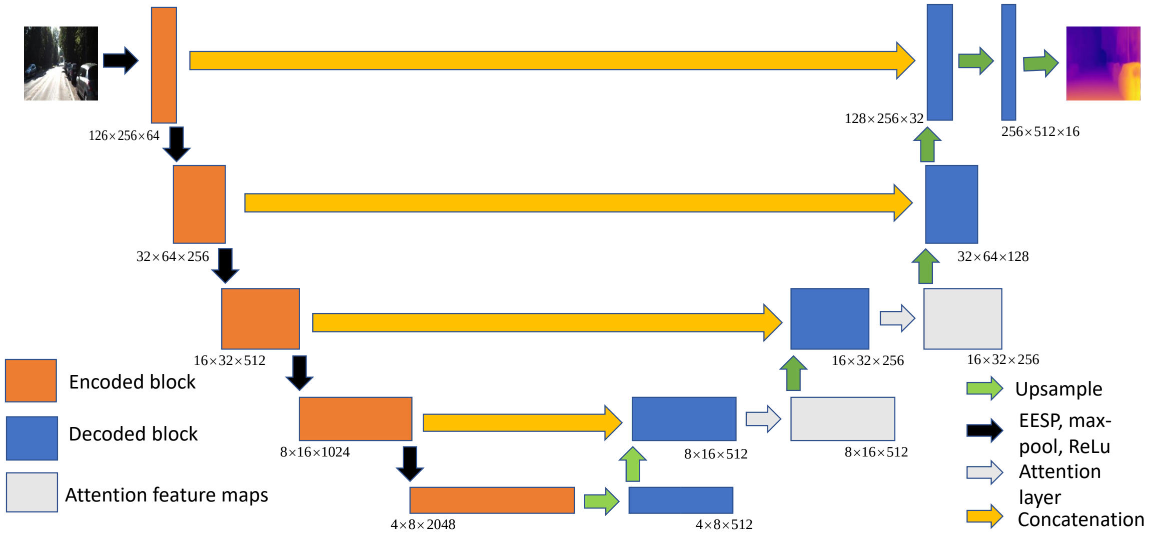

Figure 3.2: Decomposition of the generator part in Figure 3.1. Our method implements context-aware attention block in the decoder part of generator along with EESP unit for efficiency and accuracy. I apply the attention mechanism in the early stages of the decoder to better extract medium to high level features.

Each half generates disparities of different views, R_d, L_d. To enforce a view constraint, a consistency loss is formulated,

$$L_c = \|L_d - \mathbf{w}(L_d, R_d)\| \,. \tag{3.3}$$

I consider structural similarity loss (L_{SSIM}) along with the adversarial loss for better full-cycle optimization.

$$L_{SSIM} = \frac{(2 \times \mu_x \times \mu_y + C_1) \times (2 \times \sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{3.4}$$

where μ_x and μ_y denote local sample means of x and y respectively. σ_x and σ_y denote local sample standard deviations of x and y respectively. σ_{xy} denotes local sample correlation coefficient between x and y. C_1 and C_2 are constants for stabilization if denominator is too small. The total loss for the full cycle optimization is:

$$\mathbf{Loss} = \alpha L_r + \beta L_{GAN} + \gamma L_c + \delta L_{SSIM}. \tag{3.5}$$

α, β, γ and δ are the corresponding weights for different losses. The output disparity map is obtained by

$$\mathbf{D} = (L_d + \mathbf{w}(L_d, R_d))/2. \tag{3.6}$$

As shown in Figure 3.2, I modified the generator and discriminator parts in the original network by replacing the standard convolutions with EESP convolution factorization and implemented our proposed context aware attention layer in the decoder part of generator network. As proposed by Zhang [36], attention mechanism works best on middle to high level feature maps as it receives more evidence to choose the conditions. I apply our attention layer on the first two layers of decoder since they contain the high level feature maps of the generated image. In order to stabilize learning, I use spectral normalization [34] in the discriminator network, as it is critical for the generator to learn the multi-modal structure of the target distribution by controlling the performance of a discriminator. Spectral normalization puts constrains on the Lipschitz constant of the discriminator network without any extra hyper-parameter tuning [34] to improve GAN training stability.

3.3 EESP Module

The EESP module is empowered by group convolution and parallel branches of depth-wise dilated separable convolution. As shown in Figure 3(a), this technique reduces the high dimensional input features into a low dimensional space using group convolution. Then it learns the low dimensional feature representation in parallel branches using depth-wise dilated separable convolution with different dilation rates (larger dilation rate corresponds to larger size of receptive field) followed by hierarchical addition in order to remove the gridding artifacts. As proposed by Mehta [7] depth-wise dilated separable convolutions and group convolutions are more efficient.

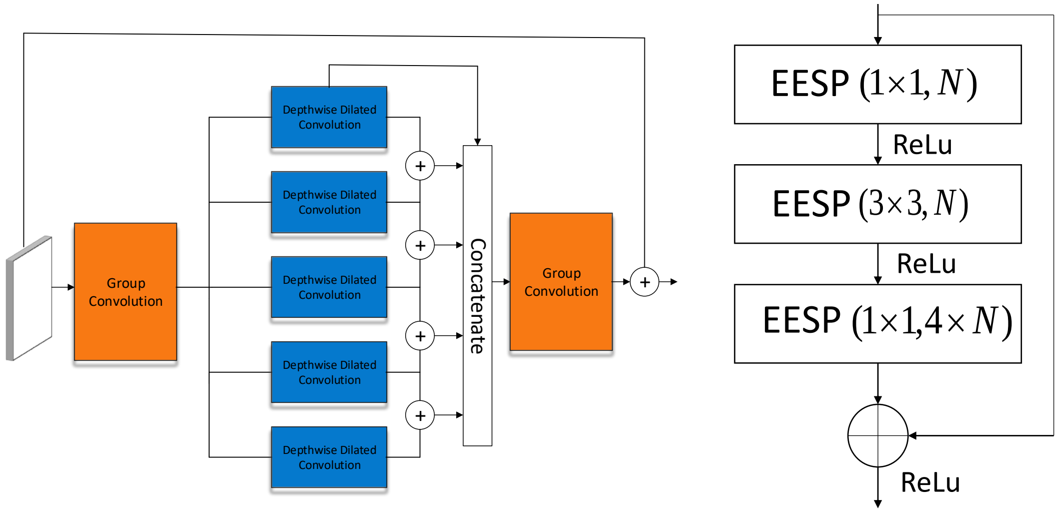
Group Convolution
Depthwise Dilated Convolution
Depthwise Dilated Convolution
Depthwise Dilated Convolution
Depthwise Dilated Convolution
Depthwise Dilated Convolution
+
+
+
+
Concatenate
Group Convolution
+
EESP $(1\times1, N)$
ReLu
EESP $(3\times3, N)$
ReLu
EESP $(1\times1, 4\times N)$
ReLu

Figure 3.3: Convolution factorization block (a) Schematic diagram of a single EESP unit (b) A bottleneck building block [2] using EESP (N = output dimension).

3.4 Attention Layer

Convolution layers process images in a local neighborhood of the image. Convolution layers alone do not capture long range dependencies. These long range dependencies are useful in generative networks like GAN to enhance the synthesized images. In this section, I will discuss about our context aware lightweight attention mechanism, as shown in Figure 3.4. This attention module is able to capture the detailed context information to enhance the estimated depth image. Our attention mechanism exploits multi-scale features fusion [37] for each layer with increasing receptive field of kernels. I also use factorized convolution (EESP module) for feature extraction. These features are concatenated with their corresponding global context features. Specifically, the global context features are obtained through global average pooling (channel wise attention) across the whole feature map for each layer. Finally, the multi-scale outputs are hierarchically fused to generate the final output feature map. Global average pooling outputs a 1D context vector which is replicated to the

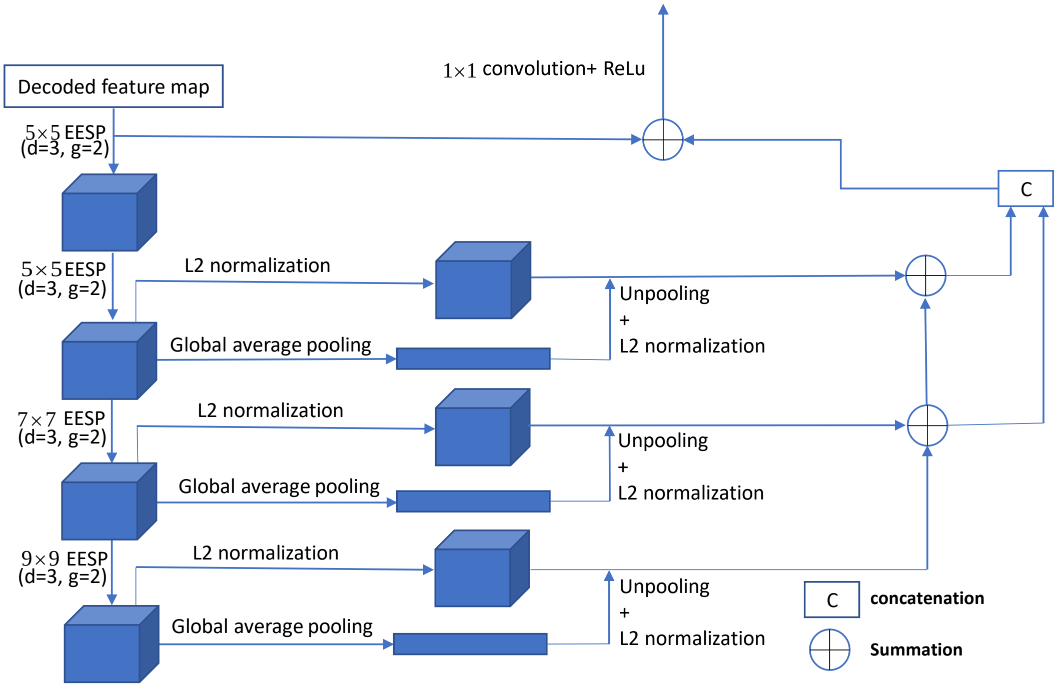

Figure 3.4: The context-aware attention architecture. Increasing receptive field of kernels (with dilation rate $(d) = 3$ and group $(g) = 2$) and their corresponding global feature context helps to obtain context aware attention features. The dilation rate and group number are fixed in the architecture for the factorized convolution. Intermediate decoded feature map is the input to the architecture as shown in Figure 3.2. In our case, hierarchical fusion of different layer features provides the best result.

same size of the feature maps to merge. Merging two features for each scale is not efficient enough to produce a good result because (i) different scales of the two feature maps and (ii) the unpooled global feature vector is not as dominant as large multi-scale feature maps, so plain concatenation may be futile. Although during training the weght might get adjusted, it requires heavy parameter tuning. So I apply per-pixel L_2 normalization to both features to be merged along with a learnable scale parameter for each channel. For an n-dimensional input X after L_2 normalization I obtain $X_1 = \zeta * \frac{X}{\|\mathbf{X}\|_2}$, where $\|\mathbf{X}\|_2$ is $\sqrt{\sum_{n=1}^{d} |X_n|^2}$ and ζ is a scaling parameter.

3.5 Dataset

KITTI dataset [3] contains several outdoor scenes from LIDAR sensor and car-mounted cameras while driving. I use the data split as suggested by Eigen [4] for both training and testing. It contains 22600 training image pairs and 697 test image pairs. The input images have been down sampled to 512×256 resolution image with respect to original resolution of 1224×368. Random data augmentation has been done by flipping of images during training. Cityscapes dataset [16] consists of 22,973 training stereo pairs captured across various German cities. It gives higher resolution image quality and variety compared to KITTI. Both of these datasets are highly recognized for various computer vision tasks, segmentation, classification, depth prediction .

CHAPTER 4: RESULTS

In this section, I evaluated the proposed model in terms of efficiency along with qualitative and quantitative demonstrations using KITTI [3] and Cityscapes [16] datasets in order to show the effectiveness of the proposed model.

4.1 Implementation Details

In our experiments, I set the dilation rate d in EESP module proportional to the number of branches in the EESP (for our experiments, I used 5 parallel branches, dilation rates from 2^0 to 2^4, with number of groups 2). The effective receptive field of the EESP unit grows with the number of branches, as shown in Figure 3(a). As shown in Figure 3.2, the generator networks use a Resnet-50 network for the encoder and the decoder contains five deconvolutional layers with ReLU operations. For the first two layers in the decoder, I integrate the attention layer in order to process the large feature maps for context information. Skip connections are used to pass information from encoder to decoder in order to aggregate efficient feature representation. All the convolution operations in the generator part are replaced by the factorized EESP module. $D1$ and $D2$ each has five consecutive EESP operations. I use the bilinear sampler for the warping operation by following [6].

4.2 Experimental Setup

The proposed model is implemented using TensorFlow [38] and takes 21 hours to train using a single NVIDIA-GTX 1080Ti GPU. The batch size is set to 8. The

Table 4.1: Efficiency comparison between two networks. Our network achieves a significant reduction of computational complexity as compared to the original GAN approach [1].

Network	Arch.	FLOPs (bil.)	Param (mil.)
Original GAN [1]	GAN	8993	125
Ours	GAN	**5833**	**96.5**

Table 4.2: Total inference time using a single NVIDIA-GTX 1080Ti GPU on the KITTI dataset. It shows our approach outperforms the baseline model [1].

Network	Architecture	Inf. time (sec.)
Original GAN [1]	GAN	0.190
Ours	GAN	**0.130**

initial learning rate is 10^{-6} and is reduced by half at $[40k, 70k]$ steps. I use ADAM [39] optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.9$ and Weight decay $= 0.006$ to train the model with 100 epochs.

Table 4.3: Quantitative evaluation of different modification of the network on KITTI dataset as ablation study. I can clearly observe that the attention mechanism is able to improve the performance as compared to using EESP module alone. Red indicates the best result and blue is the second best.

Method	Sup	Abs.Rel ↓	Sq.Rel ↓	RMSE ↓	RMSE(log) ↓	$\delta < 1.25$↑	$\delta < 1.25^2$↑	$\delta < 1.25^3$↑
Baseline [1]	N	0.198	1.990	6.655	0.292	0.721	0.884	0.949
Ours (EESP)	N	0.195	1.76	6.09	0.292	0.758	0.905	0.958
Ours (attention)	N	0.138	0.915	4.571	0.247	0.831	0.919	0.964
Ours (EESP+attention)	N	**0.1196**	**0.889**	**4.329**	**0.192**	**0.865**	**0.943**	**0.989**

4.3 Evaluation

As per the previous works [4], I evaluate our depth estimation using the following evaluation metrics. Considering d_i and d_{gi} are the estimated depth and ground truth depth value for pixel i. T is the total number of valid pixels in the test set.

$$\text{Abs.Rel} = \frac{1}{T}\sum_i \frac{d_i - d_{gi}}{d_{gi}} \tag{4.1}$$

$$\text{Sq.Rel} = \frac{1}{T}\sum_i \frac{|d_i - d_{gi}|^2}{d_{gi}} \tag{4.2}$$

$$\log \text{RMSE} = \sqrt{\frac{1}{T}\sum_i ||\log(d_i) - \log(d_{gi})||^2} \tag{4.3}$$

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_i ||d_i - d_{gi}||^2} \tag{4.4}$$

The accuracy with threshold t so that $\delta = max(\frac{d_{gi}}{d_i}, \frac{d_i}{d_{gi}}) < t$, where t=1.25, 1.25^2, 1.25^3.

I compare the proposed model with the state of the art supervised and unsupervised depth estimation methods for both datasets. Tables 4.4 to 4.5 and Figures 5 to 6 are the respective quantitative and qualitative analysis of our method and other approaches on KITTI and Cityscapes. Compare with the supervised approaches, I have achieved very similar results to the best performing method Xu [40]. From the unsupervised approaches, our approach significantly outperforms Godard [6], which represents the state of the art among unsupervised approaches to this task. Finally, I also compare with Pilzer 's [1] full-cycle+D training and ours yields better results.

4.4 Experiments

In this section, we evaluated our proposed model extensively using KITTI [3] and Cityscapes [16] datasets. We present quantitative and qualitative results to demonstrate the effectiveness of the proposed model.

4.5 Ablation Study

To validate the contribution of our context aware attention strategy and the convolution factorization to overall performance, I present an ablation study on KITTI dataset, (i) replacing convolution operations with EESP (ii) implementation of attention mechanism into baseline with convolution operations. Table 4.6 shows the breakdown of various components involved in each experiments. These individual experiments highlight the impact of particular components , EESP, attention layer into the baseline model.

Table 4.4: Quantitative Comparison with the state of the art methods trained and tested on KITTI dataset. Supervised and unsupervised methods are labelled as "Y" and "N". Results are obtained on Eigen split dataset. I train the Pilzer [1] method using the Full-Cycle+D method. Red indicates the best result and blue is the second best. Our approach outperforms the state of the art on 5 out of 7 evaluation metrics.

Method	Sup	Abs.Rel ↓	Sq.Rel ↓	RMSE ↓	RMSE(log) ↓	$\delta < 1.25$↑	$\delta < 1.25^2$↑	$\delta < 1.25^3$↑
Eigen [4]	Y	0.190	1.515	7.156	0.270	0.692	0.899	0.967
Liu [41]	Y	0.202	1.614	6.523	0.275	0.678	0.895	0.965
Xu [40]	Y	**0.132**	**0.911**	–	**0.162**	0.804	**0.945**	**0.981**
Zhou [26]	N	0.208	1.768	6.856	0.283	0.678	0.885	0.957
AdaDepth [28]	N	0.203	1.734	6.251	0.284	0.687	0.899	0.958
Garg [23]	N	0.169	1.080	**5.104**	0.273	0.740	0.904	0.962
Godard [6]	N	0.148	1.344	5.927	0.247	**0.803**	0.922	0.964
Pilzer [1]	N	0.198	1.990	6.655	0.292	0.721	0.884	0.949
Wang [22]	N	0.151	1.257	5.583	0.228	0.810	0.936	0.974
Ours (EESP+attention)	N	**0.1196**	**0.889**	**4.329**	**0.192**	**0.865**	**0.943**	0.989

Table 4.5: Quantitative Comparison on Cityscapes dataset. Supervised and unsupervised methods are labeled as "Y" and "N". I train Pilzer [1]'s network using the Full-Cycle+D method. Red indicates the best result and blue is the second best. Our approach outperforms existing state of the art approaches on 4 out of 7 evaluation metrics. Note that we directly apply our model trained on KITTI dataset without any specific tuning.

Method	Sup	Abs.Rel ↓	Sq.Rel ↓	RMSE ↓	RMSE(log) ↓	$\delta < 1.25$↑	$\delta < 1.25^2$↑	$\delta < 1.25^3$↑
Pilzer [1]	N	0.440	6.036	5.443	0.398	0.730	0.887	0.944
Wang [22]	N	0.148	1.187	5.496	0.226	0.812	0.938	0.975
Godard [6]	N	**0.097**	**0.896**	**5.093**	**0.176**	**0.879**	**0.962**	**0.986**
Zhou [26]	N	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Ours (EESP+attention)	N	**0.090**	**0.813**	**4.633**	**0.193**	**0.832**	**0.974**	**0.978**

Table 4.6: The components in our proposed model for ablation study.

Methods	Convolution operation	EESP operation	Context aware attention mechanism
Baseline [1]	✓	✗	✗
Ours (EESP)	✗	✓	✗
Ours (attention)	✓	✗	✓
Ours (EESP+attention)	✗	✓	✓

As can be seen from Table 4.3, the model with only EESP operation performs a little better than the baseline model. However, the factorization operation leads

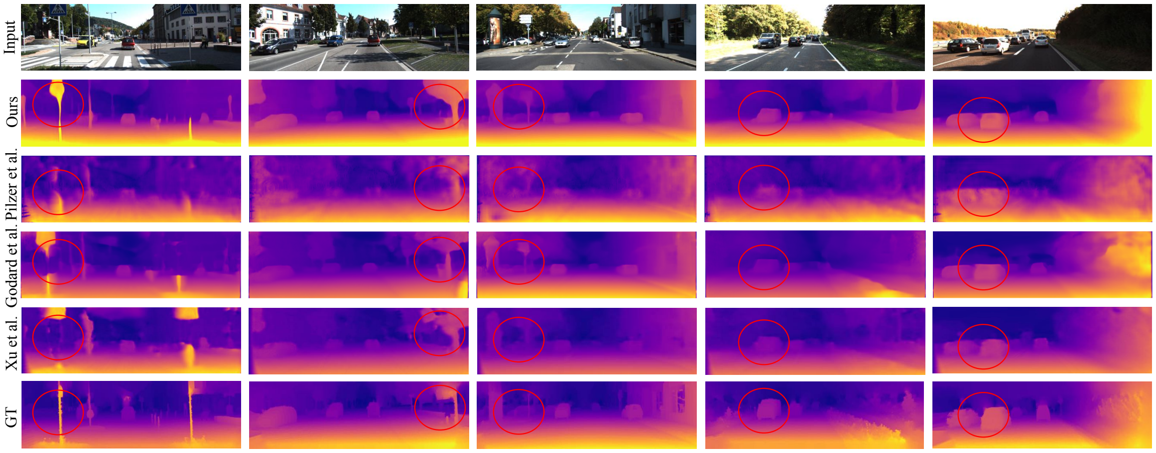

Figure 4.1: Qualitative measurements on KITTI dataset [3]. Due to the attention layer, our approach generates the subtle structural details of the image compared to the other state of the art unsupervised methods. The ground truth depth maps are interpolated from sparse LIDAR points for visualization purpose only.

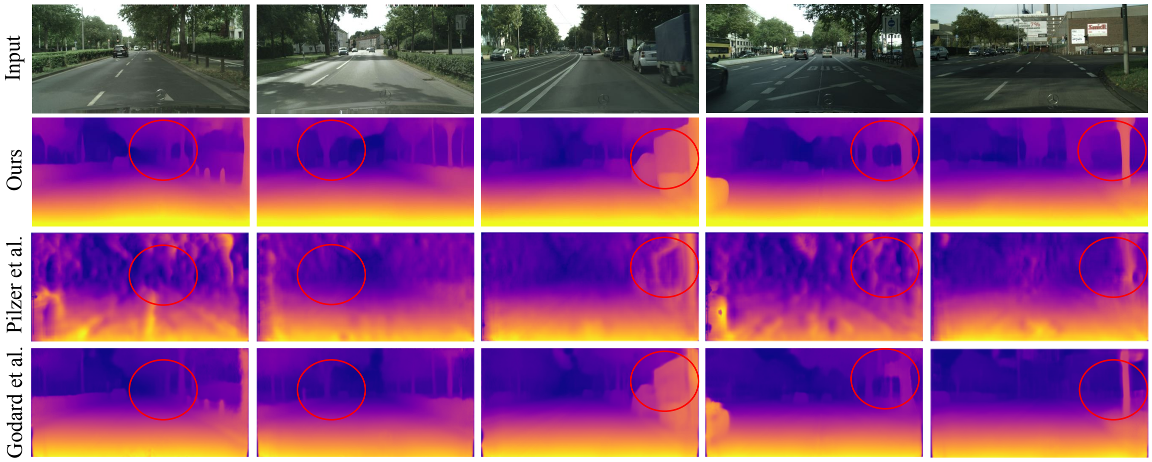

Figure 4.2: Qualitative comparison on the Cityscapes dataset. Our model is trained on KITTI dataset and evaluated on Cityscapes without any specific tuning.

to significant ease of computation. Then, I implement attention mechanism with baseline which clearly generates enhanced depth map. This verifies our intuition of integrating attention with EESP operation to improve the generated depth image.

As illustrated in Figure 4.3, I also qualitatively demonstrate the impact of context-aware attention for model learning, by presenting the predicted depth maps from initial training epochs. The evolution of these predicted depth maps reveal that our context aware attention architecture is able to focus on the salient objects in the

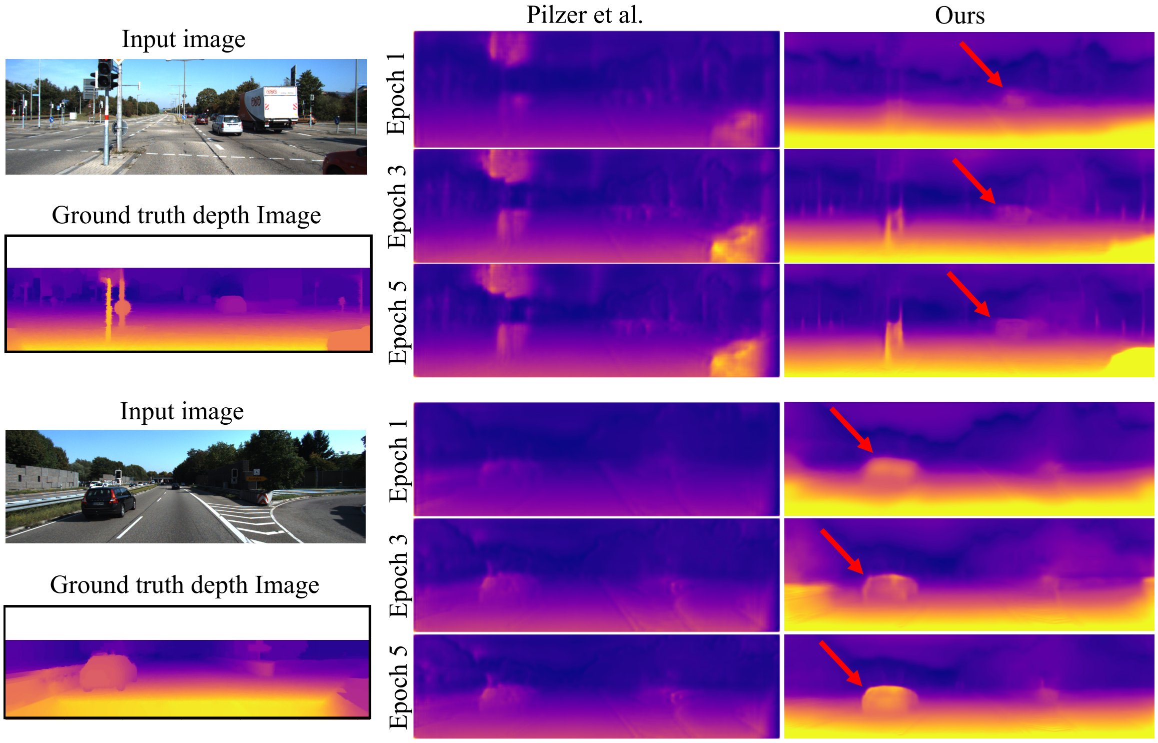

Figure 4.3: I investigate the impact of context aware attention learning for depth prediction. Our proposed context-aware attention mechanism provides effective learning and captures refined image details in the early stage of learning. For example, it learns the structure of the far-away car in epoch 1 for both images.

image and captures the depth information in the early stage (e.g. the first epoch) of training. Based on the comparison between our proposed method with the baseline model at the early stages of learning, I believe that the attention mechanism improves the network ability to learn fine image details quickly.

4.6 More Visual Comparison

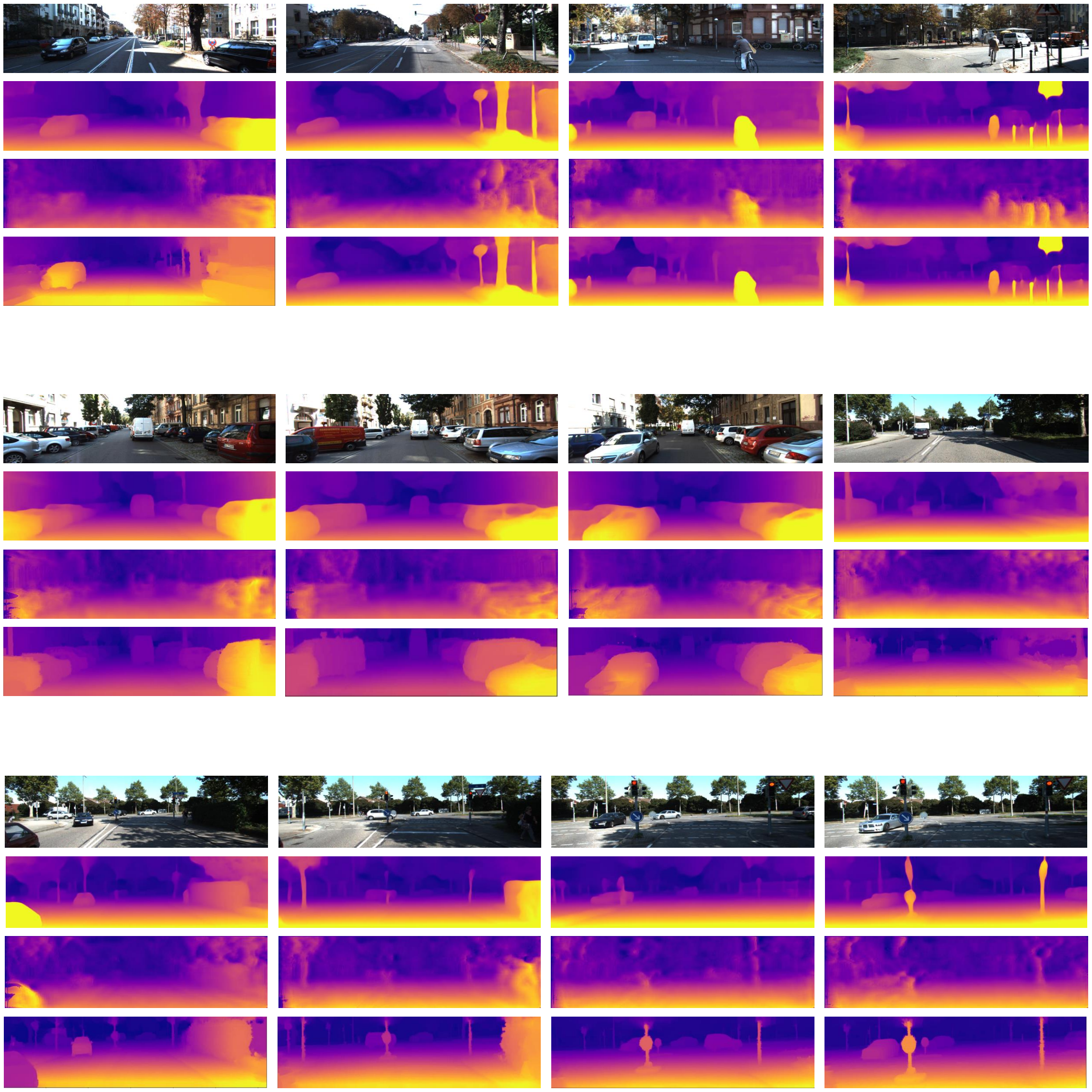

Figure 4.4: This visualization contains input image (1st row), our result (2nd row), Pilzer approach [1] (3rd row) and ground truth depth maps (4th row). The ground truth depth maps are interpolated from sparse LIDAR points for visualization purposes only.

CHAPTER 5: CONCLUSION and FUTURE WORK

5.1 Conclusion

We introduced an efficient approach to build a GAN architecture for unsupervised depth estimation. Our network takes advantage of the convolution factorization for learning richer image representation along with more efficient com- putation. Our proposed attention mechanism also provides structured scene output. Our work shows significant reduction of computation is possible for deep networks without compromising performance. Experiments on publicly available datasets demonstrate the efficiency of our approach and competitive performance compared to the state of the art approaches.

5.2 Future Work

Depth estimation from Underwater image is a really growing field. Although, the only thing human binocular vision is not able to perceive is underwater depth from far, inside water it is very tough to distinguish how much far the object is. Due to limitation of generalized dataset this field is not well explored, still work of Gupta et al.[42] is fascinating.

REFERENCES

[1] A. Pilzer, D. Xu, M. Puscas, E. Ricci, and N. Sebe, "Unsupervised adversarial depth estimation using cycled generative networks," in *2018 International Conference on 3D Vision (3DV)*, pp. 587–595, IEEE, 2018.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[3] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, IEEE, 2012.

[4] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, pp. 2366–2374, 2014.

[5] D. Xu, W. Wang, H. Tang, H. Liu, N. Sebe, and E. Ricci, "Structured attention guided convolutional neural fields for monocular depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3917–3925, 2018.

[6] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 270–279, 2017.

[7] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, "Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network," *arXiv preprint arXiv:1811.11431*, 2018.

[8] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[9] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deepstereo: Learning to predict new views from the world's imagery," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5515–5524, 2016.

[10] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Advances in neural information processing systems*, pp. 1161–1168, 2006.

[11] A. Saxena, M. Sun, and A. Y. Ng, "Learning 3-d scene structure from a single still image," in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, IEEE, 2007.

[12] J. Konrad, M. Wang, P. Ishwar, C. Wu, and D. Mukherjee, “Learning-based, automatic 2d-to-3d image and video conversion,” *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3485–3496, 2013.

[13] D. Hoiem, A. A. Efros, and M. Hebert, “Recovering surface layout from an image,” *International Journal of Computer Vision*, vol. 75, no. 1, pp. 151–172, 2007.

[14] R. Chen, F. Mahmood, A. Yuille, and N. J. Durr, “Rethinking monocular depth estimation with adversarial training,” *arXiv preprint arXiv:1808.07528*, 2018.

[15] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *European Conference on Computer Vision*, pp. 746–760, Springer, 2012.

[16] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[17] B. Li, C. Shen, Y. Dai, A. Van Den Hengel, and M. He, “Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1119–1127, 2015.

[18] 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*, vol. 28, no. 11, pp. 3174–3182, 2017.

[19] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille, “Towards unified depth and semantic prediction from a single image,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2800–2809, 2015.

[20] D. Xu, W. Ouyang, X. Wang, and N. Sebe, “Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 675–684, 2018.

[21] H. Zhan, R. Garg, C. Saroj Weerasekera, K. Li, H. Agarwal, and I. Reid, “Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 340–349, 2018.

[22] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey, “Learning depth from monocular videos using direct methods,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2022–2030, 2018.

[23] R. Garg, V. K. BG, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *European Conference on Computer Vision*, pp. 740–756, Springer, 2016.

[24] T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang, and A. A. Efros, "Learning dense correspondence via 3d-guided cycle consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 117–126, 2016.

[25] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5667–5675, 2018.

[26] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1851–1858, 2017.

[27] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[28] J. Nath Kundu, P. Krishna Uppala, A. Pahuja, and R. Venkatesh Babu, "Adadepth: Unsupervised content congruent adaptation for depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2656–2665, 2018.

[29] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[30] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

[31] A. CS Kumar, S. M. Bhandarkar, and M. Prasad, "Monocular depth prediction using generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 300–308, 2018.

[32] Y. Almalioglu, M. R. U. Saputra, P. P. de Gusmao, A. Markham, and N. Trigoni, "Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks," *arXiv preprint arXiv:1809.05786*, 2018.

[33] Z. Hao, Y. Li, S. You, and F. Lu, "Detail preserving depth estimation from a single image using attention guided networks," in *2018 International Conference on 3D Vision (3DV)*, pp. 304–313, IEEE, 2018.

[34] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[36] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *arXiv preprint arXiv:1805.08318*, 2018.

[37] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403, 2015.

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

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[40] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, "Monocular depth estimation using multi-scale continuous crfs as sequential deep networks," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[41] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2024–2039, 2016.

[42] H. Gupta and K. Mitra, "Unsupervised single image underwater depth estimation," *arXiv preprint arXiv:1905.10595*, 2019.