CS231A Final Project Report: Self-supervised Monocular Depth Estimation with Semantic Guidance

Rongbin Li*
Stanford University
rongbli@stanford.edu

Chih-Ying Liu*
Stanford University
ying1029@stanford.edu

Chuanqi Chen*
Stanford University
cchuanqi@stanford.edu

Abstract

In monocular depth estimation, recent researches show that the unsupervised or self-supervised methods are often more attractive than supervised ones, since they avoid collecting large amount of depth data which are often impractical in various real world scenarios. Additionally, the accuracy of depth estimation can be enhanced by fusing geometric representation from semantic information of images into depth features using different attention mechanism to guide the depth prediction. In this study, we reproduce the results of [5, 10] and deeply study the attention methods used in these two depth-semantic fusion modules. Furthermore we design a new self-adaptive attention method and compare the performances of these three ways of attention via extensive experiments on KITTI dataset [1]. In addition, we spend efforts on lightweight monocular depth estimation models which enables the possibility of real-time inference on an embedded platform with lower power consumption. Our new self-adaptive attention methods with lightweight models outperform [5] and are on par with [10] but with faster training and more efficient inference. The code is available at https://github.com/chuanqichen/CS231A_Final_Project.git.[1]

1. Introduction

Accurate depth estimation is crucial for many tasks like perception, navigation and path planning in the computer vision and robotics field. Traditionally expensive 3d hardware like Lidar sensor and multiple camera are used to measure the depth of the object in the surrounding environments, but the performance of these sensors and cameras can degrade greatly in the low lighting or noisy environment and becomes less reliable due to hardware failures.

Supervised methods that use ground truth depth data to predict per pixel depth from an image have shown great promise to serve as alternative to these expensive 3d sensors. But it requires collecting large amount of depth data and label for supervised model training. It is labor intensive and also impractical in various real world scenarios. Therefore, unsupervised or self-supervised ways of depth estimation that only need stereo pairs or monocular video are more attractive [2, 3].

Furthermore, since semantic of image provides important information about objects, say, their geometry and boundaries information between them, it is natural to think of incorporating semantic information into depth estimation. Recent work [5, 10] fuse semantic features into depth features using different attention mechanisms, to guide the depth prediction, and achieve obviously better results than using depth features only.

In this study, our contributions are summarized as following: first of all, based on [5], we use semantic information to support self-supervised monocular depth estimation (we implement our code based on the code of FSRE-depth [5][2]). Secondly, we add ROIFormer module to reproduce the result of [10]. Thirdly, we introduce a new self-adaptive attention method. Finally, we integrate with several computation efficient backbones to achieve lightweight high-efficient depth estimation models. We have extensive experiments on KITTI dataset [1] showing that new self-adaptive attention methods integrated with lightweight Vovnet backbone [6] outperforms [5] and is on par with [10], but with faster training and inference speed, and thus should have less power consumption and low latency.

2. Related Work

2.1. Self-supervised monocular depth estimation

Self-supervised depth estimation can be solved as a learning problem to reconstruct the target image from the source image in another point of view. Several methods use the reconstruction error of the transferred view to guide the model learning correct depth map (as well as pose arguments).

[1] * Equally Contribution

[2] https://github.com/hyBlue/FSRE-Depth

[2] performs depth estimation on rectified calibrated stereo pair by using predicted disparity maps to estimate depth. Disparity maps are predicted for both views and used for the reconstruction of both images from the opposite view. [3] performs depth estimation on both stereo pairs and monocular video. The model learns the depth and pose network simultaneously, and use the predicted depth map and pose arguments to reconstruct different views.

2.2. Semantic-guided for depth estimation

Since semantic of image gives model information about geometry, several works improve monocular depth estimation by incorporating semantic information. [5] proposes novel ideas to improve self-supervised monocular depth estimation by leveraging cross-domain scene semantics information. [10] leverages semantic information by proposing a framework called ROIFormer, that performs efficient local attention for better feature fusion between depth and semantic information.

2.3. Real-time Monocular Depth Estimation

State-of-the-art monocular depth estimation algorithms are based on fairly complex deep neural networks that are too slow for real-time inference on an embedded platform, for example, a mobile or a micro aerial vehicle. Hence, in recent years, lightweight and high-efficiency depth models have gained more and more importance. [9] proposes an efficient and lightweight encoder-decoder network architecture (FastDepth), which can run at 178 fps on an NVIDIA Jetson TX2 GPU and at 27 fps when using only the TX2 CPU.

3. Method

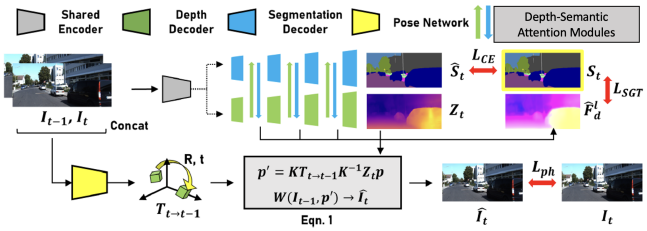


Figure 1. Framework [5]

3.1. Framework

Figure 1 shows our framework. It includes a pose network, a shared encoder, a depth decoder, a segmentation decoder, and depth-semantic attention modules that incorporate depth and semantic information. To reduce computation, depth network and segmentation network share the same encoder. The segmentation decoder is trained supervisedly with cross entropy loss. Following we introduce the methodology for self-supervised depth estimation and how we utilize semantic information.

3.1.1 Self-Supervised Depth Estimation

Our goal is to predict Z_t with only a monocular image I_t during inference time. Given source image $I_{t'}$ ($t' \in \{t-1, t+1\}$) and target image I_t, we can compute relative pose $T_{t\to t'}$. Our training objective is to predict $I_{t'\to t}$ which is a reference image of target image I_t via geometric transformation (changing viewpoint from $I_{t'}$ to I_t). With known camera matrix K and six degree-of-freedom relative pose $T_{t\to t'}$ given by pose network. We can derived the projected pixel coordinates and use them to form $I_{t'\to t}$.

$$p' = KT_{t\to t'}Z_t K^{-1} p$$

$$\hat{I}_t = W_t(I_{t-1}, p')$$

where $W_t(.)$ is bi-linear sampler that linearly interpolates nearby pixels at p' and assigns value in $\hat{I}_t$. To minimize the discrepancy between $I_{t'\to t}$ and I_t. We use L1 loss and structural similarity index measure (SSIM) as photometric loss L_{ph}.

$$L_{ph} = \alpha \frac{1 - \text{SSIM}(I_t, \hat{I}_t)}{2} + (1-\alpha)|I_t - \hat{I}_t|$$

To enhance the smoothness in depth map prediction, we also added edge-aware smoothness loss L_{sm}.

$$L_{sm} = |\partial_x d_t| e^{-|\partial_x I_t} + |\partial_y d_t| e^{-|\partial_y I_t|}$$

The total baseline loss (without semantic guidance) is

$$L_{\text{baseline}} = L_{ph} + \beta L_{sm}$$

where β is hyper-parameter.

3.1.2 Semantic-guided Triplet Loss

We use semantic-guided triplet loss L_{sgt} proposed by [5] to enhance semantic information. The motivation of this loss is that adjacent pixels within each object instance have similar depth value, whereas those across semantic boundary have larger depth difference. We divide an image into $K \times K$ patches, and define the center pixel of each patch as anchor pixel. For each patch with anchor i, we define pixels with same semantic class as anchor pixel as positive pixels $\mathcal{P}_i^+$, and pixels with different classes as negative pixels $\mathcal{P}_i^-$. Figure 2 shows the illustration of semantic-guided triplet loss.

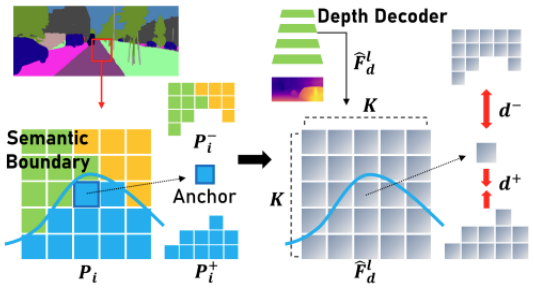


Figure 2. Semantic guided triplet loss. [5]

We define the positive distance d^+ and negative distance d^- as mean of the Euclidean distance of the L2 normalization of depth feature pairs.

$$d^+(i) = \frac{1}{|\mathcal{P}_i^+|} \sum_{j \in \mathcal{P}_i^+} \sqrt{\hat{F}_d^l(i) - \hat{F}_d^l(j)}$$

$$d^-(i) = \frac{1}{|\mathcal{P}_i^-|} \sum_{j \in \mathcal{P}_i^-} \sqrt{\hat{F}_d^l(i) - \hat{F}_d^l(j)}$$

where $\hat{F}_d^l = F_d^l / \|F_d^l\|$, and F_d^l is depth feature at layer l. To reduce distance between positive features and anchor feature, and increase distance between negative feature and anchor feature, the triplet loss for a margin m (a hyper-parameter) is defined as

$$\mathcal{L}_{P_i} = \max(0, d^+(i) + m - d^-(i))$$

We only consider those patches that are on the semantic boundary: we define a hyper-parameter T. A patch is on semantic boundary, if $|\mathcal{P}_i^+|$ and $|P_i^-|$ are both larger than T. The final loss containing only patches on semantic boundary is

$$\mathcal{L}_{SGT} = \frac{\sum_i \mathbb{1}\left[|\mathcal{P}_i^+|, |\mathcal{P}_i^-| > T\right] \cdot \mathcal{L}_{P_i}}{\sum_i \mathbb{1}\left[|\mathcal{P}_i^+|, |\mathcal{P}_i^-| > T\right]}$$

We add the baseline loss with $\mathcal{L}_{SGT}$ multiplied by δ.

3.2. Depth-Semantic Attention Modules

Both FSRE [5] and ROIFormer [10] follow the framework shown in Figure 1. In their design, each decoder comprises five layers (l = 0, 1, 2, 3, 4) and the spatial resolution of the feature map is doubled for each.

For layer l, the depth (segmentation) decoder generates a feature map, F_d^l (resp. F_s^l), whose spatial resolution are $(H/2^{4-l}, W/2^{4-l})$. There is a depth-semantic fusion module located between each corresponding layers of the two decoders (actually only for l = 1, 2, 3), which enables cross-modal interactions between two task-specific decoders. Specifically, the attention module takes depth feature map F_d^l and semantic feature map F_s^l as inputs, and outputs the refined feature maps $F_d^{l'}$ and $F_s^{l'}$, which are the fused representation of depth and semantic. $F_d^{l'}$ and $F_s^{l'}$ would then be passed to next block as inputs.

In this section, we will dive into the the depth-semantic attention modules used by FSRE [5] and RoiFormer [10], and, later, we will design a new attention module based on [10]. The differences between the three methods are how to sample the features that be attended to and how to assign weights over them. We will show experimental results on all of these three methods in Section 4.

3.2.1 FSRE attention method

In FSRE paper [5], they propose a Cross-task Multiembedding Attention (CMA) module to produce semantics-aware depth features through the representation subspaces and utilize them to refine depth predictions. They use two CMA modules simultaneously to enable bidirectional feature enhancement, where depth (segmentation) becomes the target (reference) in one CMA module while their roles change in the other. Following, we only describe a single case where the depth feature is the target for ease of explanation.

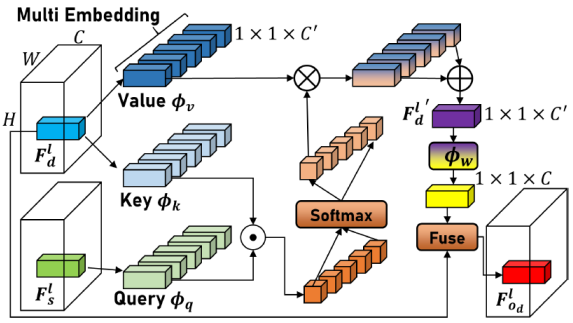


Figure 3. The attention module of FSRE: CMA

Please see Figure 3 for the design of CMA. To generate the refined depth feature $F_d^{l'}(i)$ from original depth feature $F_d^l(i)$, where i is the spatial index of each feature, they only use the semantic feature in the same spatial location ($F_s^l(i)$) as the reference. They adopt multiple linear projections to project original depth and semantic feature vectors onto H different representation subspaces. Each projection transforms the input feature from the original dimension C to C'. The refined depth feature would be the weighted sum of the projections of depth features over the H representation subspace, and each weight is the similarity between projection of depth feature and projection of semantic feature on the corresponding subspace. Here, the similarity is defined as the inner product of two vectors. Formally, the refined depth feature $F_d^l(i)'$ can be computed as follows:

$$F_d^l(i)' = \sum_h {F_d^l}^h(i)',$$

$$where \ \ {F_d^l}^h(i)' = \rho(A^h(i)) \times \phi_v^h(F_d^l(i)),$$

$$where \ \ \rho(A^h(i)) = \frac{e^{A^h(i)}}{\sum_{h' \in H} e^{A^{h'}(i)}},$$

$$and \ \ A^h(i) = \frac{\phi_k^h(F_d^l(i))^T \cdot \phi_q^h(F_s^l(i))}{\sqrt{C'}}$$

where, ϕ_q, ϕ_k and ϕ_v is the linear projection functions for query, key and value, respectively. ρ is the $softmax$ normalized function. Finally, the refined feature map $F_d^{l'}$ is projected to the original dimension C and fused with the initial feature F_d^l to produce the final output.

The above show the computation when where is only one head, in the final implementation, like [8], we will use multi-heads, i.e. query, key and value would be projected into multi representation spaces and do the computation above, finally, the output would be concatenated and fused.

3.2.2 RoiFormer Attention Method

In FSRE, to compute the refined depth feature $F_d^{l'}(i)$, they only use the semantic feature on the same spatial position i as reference. However, it can be image that, usually, the depth information of one spatial position is highly related with the depth information of nearby positions of the same semantic label. Hence, it is naturally to think up an attention method which samples the reference features from other spatial positions.

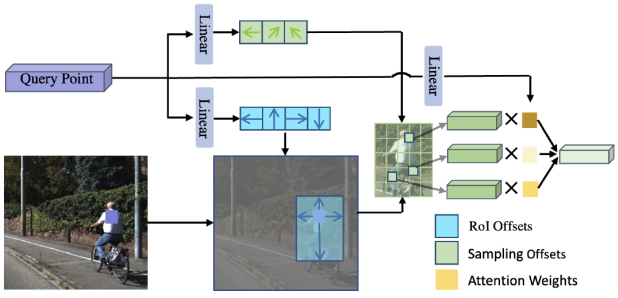


Figure 4. The attention module of RoiFormer (head=1)

One simple way is to define sample space as the whole semantic feature space F^S, as what [12] does. However, the experimental results show that, the whole sampling space F^s is too large for the linear function to find the most relative points (especially for high resolution input), yielding divergence attention and worse performance in high-resolution settings. Inspired by [11], ROIFormer generates the Region of Interests (ROIs) for each query point. Specifically, the ROI for a query point f_i is a bin $b_i = [d_l, d_t, d_r, d_b]$, which represents the distances from f_i to the edges of the ROI box and $[w = d_l + d_r, h = d_t + d_b$ are normalized width and height. Then, the reference features are only sampled form b_i. In their design (see Figure 4), for each quary point f_i, they use a (learnable) linear function to generate the 4 boundaries of bin b_i, and use another (learnable) linear function to generate n_p (a super-parameter) reference spatial locations. Because the linear function output float value, the reference features' value are computed via the interpolation on the learned position. Additionally, the weights assigned to the n_p reference features are also computed by a linear (learnable) function which takes query f_i as input. The final refined feature is the weighted sum of the reference. Formally, the feature fusion within a specific region of interest can be expressed as:

$$Fusion < f_i, b_i >= \sum_{f_j \in \Omega(b_i)} A_{i,j} \phi_v(f_j)$$

, where Ω is sampling function. ϕ_v is linear feature projection function. $A_{i,j}$ is the weighted assigned to f_j, which is computed by a linear function takes query as input. Finally, like FSRE, the refined feature map will be fused with the initial feature to produce the final output.

3.2.3 Our Self-Adaptive Attention

By deeply studied the attention mechanism of ROIFormer, we modify the way of assigning weights to reference features, and design a new attention method which is shown in Figure 5.

First, please note that, in the traditional implementation of Transformer Attention (like [8]), weight assigned to each value feature depends on the similarity of the query feature and the corresponding key feature. The more key feature like query feature, the larger weight that the corresponding value feature would be assigned to, so that the larger proportion the value would account for in the final fused feature. This means the attention is self-adaptive (i.e. auto pay attention to more likely sampled features). But, in ROIFormer, as mentioned before, the weights assigned to the sampled reference features (i.e. value features) are computed only from query point. This design is more computational efficient (just a linear function), but loses the property of self-adaptive. Intuitively, we think self-adaptive should be helpful to attention, hence, we bring this back to ROIFormer and lead to our new attention module, which will assign weights proportional to the similarity of key and query. Here, we define similarity as the inner product of two vectors. Our other parts of design are the same as ROIFormer.

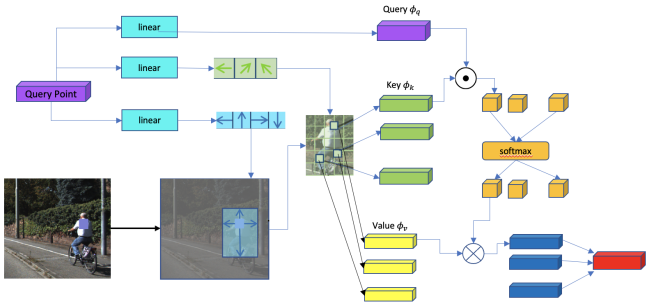


Figure 5. The attention module of adaptive ROIFormer (head=1). The weight assigned to each value feature is proportional to the similarity of key and query. The similarity here is the inner product of two vectors.

3.3. Lightweight and efficient backbone networks

The original backbone of FSRE-depth [5] uses Resnet [4] as encoder. We implemented MobileNetV2 [7] and VoVNet encoder [6]. The latter is much more computation efficient with outstanding performance.

- Resnet [4] is constructed with many residual learning block with feed-forwarded "shortcut connections" to address the problems of vanishing/exploding gradients so as to build deeper neurtal network and ease to train.
- MobileNetV2 [7] is a specifically tailored neural network architecture for mobile and power consumption cautious environment. Its core innovation is the inverted residual with linear bottleneck enabling state of the art performance and small memory footprint to reduce main memory access.
- VoVNet [6], an energy and GPU computation efficient backbone network for real time objection detection, avoids dense connected layers and also aggregates features only once in the last feature maps, One-Shot Aggregation (OSA). That's why it outperforms both Resnet and MobileNetV2 backbone network.

4. Results and Evaluation

4.1. Dataset

We use KITTI depth prediction dataset [1] (Eigen split). The images are captured by driving around the rural areas and on highways, so they mostly contain street, cars, pedestrians, and etc. The depth prediction dataset has more than 90 thousand depth maps with corresponding raw LiDaR scans and RGB images. Besides, since we use semantic information to guide the depth estimation, we also need segmentation data. We adopt the pre-computed segmentation maps (on KITTI images) supplied by [5] as ground truth and train the semantic segmentation branch with cross entropy loss.

4.2. Experiment Details

We resized images to 192×640 and use batch size 24. We use Adam optimizer with learning rate 1.5e-4 and train for 20 epochs. We set our loss weights for smoothness (β), segmentation (γ), and triplet loss (δ) as $\beta = 0.001$, $\gamma = 0.3$, and $\delta = 0.1$. For the triplet loss, we set local patch size $K = 5$, and margin $m = 0.3$. We use 4 Nvidia v100 GPU for parallel training.

4.3. Quantitative Results and Evaluation

The evaluation metrics we use include absolute relative error (Abs Rel), square relative error (Sq Rel), root mean square error (RMSE), log root mean square error (RMSElog), and the ratio of pixels has error smaller than 1.25, 1.25^2, and 1.25^3.

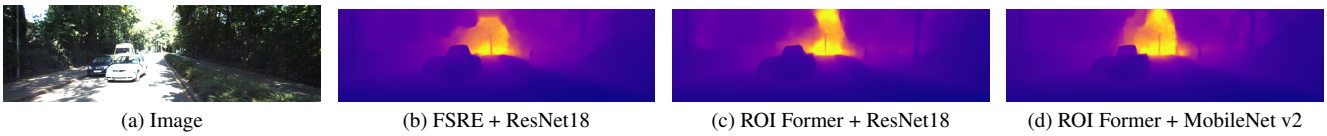
We run several depth estimation experiments with different encoders and attention modules. Please refer to Table 1 for the performance comparison. We found that ResNet50 has the best performance compared to other backbones, while Mobile Net v2 has the wortest. VoVNet has comparable performance with ResNet18 with much less parameter and faster inference time. We found VoVNet very decent backbone, considering its high performance and efficiency. In general ROI Former and ROI Former plus our self-adaptive attention have similar performance, and are better than FSRE.

4.4. Qualitative Results and Evaluation

Figure 6 and figure 7 show some depth predictions of different attention methods and backbones. Figure 8 compares the performance of ROI Former with different backbones, ResNet18, Mobile Net v2, and VoVNet. Both ResNet18 and VoVNet assign the parking sign with same depth, while Mobile Net v2 fails to do that. Figure 9 compares performance of different attention modules, cross-task multi-embedding attention (CMA), ROI Former, and Our self-adaptive attention. We circled the person and the tree truck in the image. We think it is quite hard to recognize these two objects, and the model should rely on semantic information to correctly estimate their depth. Our result shows that ROI Former performs better on estimating the depth of the person and the tree truck than FSRE (which use cross-task multi-embedding attention). ROI Former and ROI Former plus our self-adaptive attention have similar performance. ROI Former plus our self-adaptive attention with VoVNet backbone has clearest boundary of the person and the tree truck.

Attention	Backbone	Lower is better				Higher is better		
		Abs Rel	Sq Rel	RMSE	RMSE Log	< 1.25	$< 1.25^2$	$< 1.25^3$
FSRE	Resnet18	0.1082	0.7495	4.5841	0.1835	0.8833	0.9637	0.9834
FSRE	Resnet50	0.1016	0.6998	4.3752	0.1771	**0.8953**	**0.9668**	0.9842
FSRE	MobileNet v2	0.1117	0.7314	4.5673	0.1849	0.8747	0.9633	0.9841
FSRE	VoVnet	0.1048	0.7046	4.4139	0.1781	0.8895	0.9655	0.9843
ROIFormer	Reset18	0.1071	0.7347	4.512	0.1816	0.8856	0.9650	0.9839
ROIFormer	Resnet50	0.1034	0.6863	4.4146	0.1767	0.8905	0.9660	0.9846
ROIFormer	Mobile Net v2	0.1076	0.7398	4.4912	0.1819	0.8847	0.9645	0.9839
ROIFormer	VoVnet	0.1029	0.6899	**4.3181**	**0.176**	0.8943	0.9666	0.9845
ROIFormer Ours Attn	Resnet18	0.105	0.742	4.5241	0.1806	0.8883	0.9644	0.9839
ROIFormer Ours Attn	Resnet50	0.1034	**0.6765**	4.4187	0.1775	0.8897	0.966	**0.9847**
ROIFormer Ours Attn	VoVnet	**0.1027**	0.6918	4.3438	0.1769	0.8921	0.9661	0.9844

Table 1. Depth estimation results comparison



(a) Image (b) FSRE + ResNet18 (c) ROI Former + ResNet18 (d) ROI Former + MobileNet v2

(e) ROI Former + VoVNet (f) ROI Former Adaptive Attention + ResNet18 (g) ROI Former + Adaptive Attention + VovNet

Figure 6. Depth Prediction Comparision 1

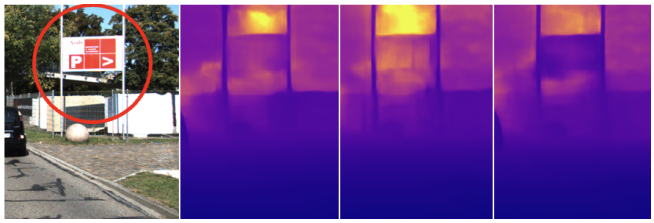


Figure 8. Comparison among different backbones with ROI Former. The predictions images from left to right are generated by backbone Resnet18, MobileNet v2, and VoVNet respectively.

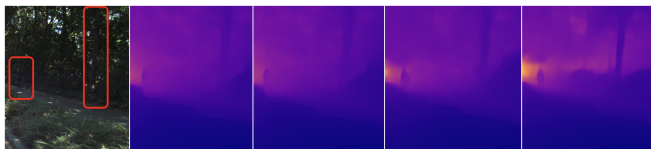


Figure 9. Comparison among different attention modules. The predictions images from left to right used cross-task multi-embedding attention (CMA) + ResNet18, ROI Former + ResNet80, ROI Former Adaptive Attention + ResNet18, ROI Former Adaptive Attention + VoVNet

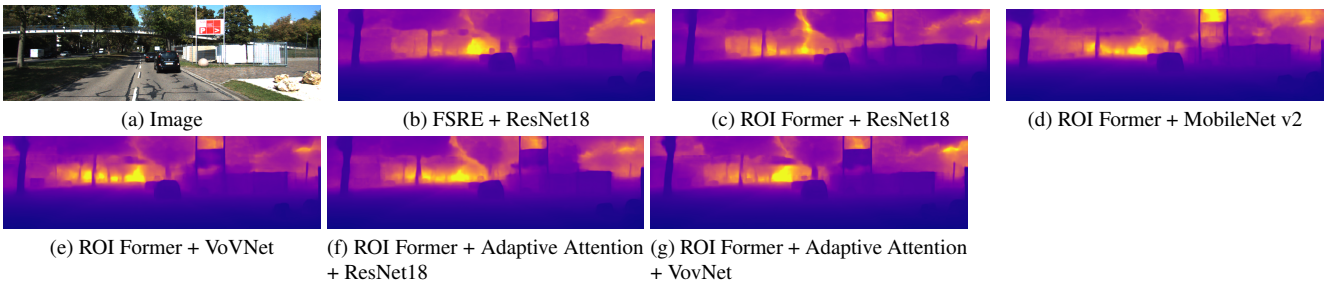
5. Conclusion

We have successfully implemented more accurate depth estimation by fusing semantic information of images, furthermore we adopted lightweight and computation efficient backbone networks to enable depth estimation in embedded platform such as mobile application.

- Semantic information does improve depth prediction accuracy.
- Our adaptive attention method outperforms that of [5], on par with [10].
- VoVNet is computation efficient with great performance.
- Future work can incorporate instance segmentation to further improve depth estimation accuracy.

6. Acknowledgement

We appreciate the teaching staff of Stanford CS231a for their excellent teaching throughout the quarter. We especially thank our mentor Andrey Kurenkov for his support and guidance throughout the project. Additionally,



(a) Image (b) FSRE + ResNet18 (c) ROI Former + ResNet18 (d) ROI Former + MobileNet v2

(e) ROI Former + VoVNet (f) ROI Former + Adaptive Attention + ResNet18 (g) ROI Former + Adaptive Attention + VovNet

Figure 7. Depth Prediction Comparision 2

we want to show greate thanks to Daitao Xing, the 1st authors of [10], because he told us many helpful implementation details when we reproducing the result of ROIFormer.

References

[1] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 1, 5

[2] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency, 2016. 1, 2

[3] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation, 2018. 1, 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 5

[5] Hyunyoung Jung, Eunhyeok Park, and Sungjoo Yoo. Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12642–12652, 2021. 1, 2, 3, 5, 6

[6] Youngwan Lee, Joong won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection, 2019. 1, 5

[7] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2019. 5

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[9] Diana Wofk, Fangchang Ma, Tien-Ju Yang, Sertac Karaman, and Vivienne Sze. FastDepth: Fast Monocular Depth Estimation on Embedded Systems. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019. 2

[10] Daitao Xing, Jinglin Shen, Chiuman Ho, and Anthony Tzes. Roiformer: Semantic-aware region of interest transformer for efficient self-supervised monocular depth estimation, 2022. 1, 2, 3, 6, 7

[11] Tong Yang, Xiangyu Zhang, Zeming Li, Wenqiang Zhang, and Jian Sun. Metaanchor: Learning to detect objects with customized anchors. *Advances in neural information processing systems*, 31, 2018. 4

[12] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 4