Abstract—Depth information is the foundation of perception, essential for autonomous driving, robotics, and other source-constrained applications. Promptly obtaining accurate and efficient depth information allows for a rapid response in dynamic environments. Sensor-based methods using LIDAR and RADAR obtain high precision at the cost of high power consumption, price, and volume. While due to advances in deep learning, vision-based approaches have recently received much attention and can overcome these drawbacks. In this work, we explore an extreme scenario in vision-based settings: estimate a depth map from one monocular image severely plagued by grid artifacts and blurry edges. To address this scenario, We first design a convolutional attention mechanism block (CAMB) which consists of channel attention and spatial attention sequentially and insert these CAMBs into skip connections. As a result, our novel approach can find the focus of current image with minimal overhead and avoid losses of depth features. Next, by combining the depth value, the gradients of X axis, Y axis and diagonal directions, and the structural similarity index measure (SSIM), we propose our novel loss function. Moreover, we utilize pixel blocks to accelerate the computation of the loss function. Finally, we show, through comprehensive experiments on two large-scale image datasets, i.e. KITTI and NYU-V2, that our method outperforms several representative baselines.

Index Terms—computer vision, deep learning, monocular depth estimation, encoder-decoder, attention-based

I. INTRODUCTION

Perception is one of the key technologies in many areas, such as autonomous driving, virtual reality, and robotics [1], which helps to detect, understand, and interpret the surrounding environments, including dynamic and static obstacles. The performance of perception usually relies on the accuracy of depth information estimation [2]. For example, autonomous driving requires to estimate the inter-vehicle distance and warn potential rear-end collisions [3], robotic arms cannot grasp the target without accurate depth information [4], and so on.

There exist many strategies to infer depth information. In general, these strategies can be classified into two categories: sensor-based methods and image-based methods [3], [5]. Sensor-based strategies, such as utilizing like LIDAR, RGB-D camera, and other active sensors [6], are able to collect depth information accurately. However, this type of methods usually places heavy burdens on manpower and computation [7]. In addition, there could be strict conditions when applying these methods. For instance, LIDAR estimates depth accurately only at sparse locations [8] and RGB-D camera suffers from its limited measurement range and outdoor sunlight sensitivity [9]. Alternatively, image-based methods can overcome these issues and be applied in a wide range of applications. The conventional image-based depth estimation methods heavily rely on multi-view geometry [10], such as stereo images [11] and consecutive frames. Nevertheless, it introduces issues such as calibration drift over time [2], [8] as well as high demands on computational resources and memory [12]. Therefore, using a monocular camera becomes an alternative low-cost, efficient, and attractive solution with light maintenance requirements for autonomous driving, robotics, and other resource-constrained applications [13].

This paper studies the extreme case in monocular depth estimation, which is to estimate the depth map from one image. This could be an ill-posed problem as there is an ambiguity in the scale of the depth [12]. Owing to the release of publicly available datasets and the advancement of
Convolutional Neural Networks (CNNs), Eigen et al. [16] first prove that the scale information can be learned by properly designing the network structure [16]. After this, there has been a lot of work along this direction [17], [18]. Despite their success, there are still some critical issues to be addressed:

- Many methods do not consider the contextual information and treat all pixels equally. It may result in the grid artifacts problem [19] and the edges in depth maps may be distorted or blurry [20], as shown in Figure 1.
- Depth estimation is often deeply integrated with industrial applications, which require real-time operation with limited computational resources. The conflict between real-time requirements and expensive computational overhead should be mitigated urgently.
- For traditional CNN architecture, such as fully connected network (FCN), after multiple layers of information processing, the depth features could be severely lost, which may lead to low accuracy and cannot meet the requirements in practice [5].

To alleviate these issues, this paper presents a new approach for depth monocular estimation from a single image. The main contributions are summarized as follows:

- We propose an encoder-decoder attention based network to effectively generate corresponding depth map from a single image and avoid grid artifacts with least possible overhead. To leverage the contextual information and find focuses of images, we design a convolutional attention mechanism block insert these CBAMs into the skip connections.
- We design a novel loss function by combining the depth value, the gradients of three dimensions (i.e. X-axis, Y-axis and diagonal direction) and structural similarity index measure (SSIM). In addition, we introduce pixel blocks, instead of single pixel, to save computational resources when calculating the loss.
- We conduct comprehensive experiments on two large-scale datasets, i.e. KITTI and NYU-V2. It is shown that our approach outperforms several representative baseline methods, which verify the effectiveness of our approach.

II. RELATED WORK

Since there is only one single image need to be calculated, depth estimation from one image can effectively reduce the computational complexity and memory overhead [8]. Numerous methods have been proposed for estimating depth information from one image in recent years. Herein, we briefly review the relevant studies.

This problem was firstly studied by Eigen et al. [16]. They regard this problem as a regression problem and propose a CNNs architecture which is composed of global coarse-scale network and local fine-scale network to generate depth maps. By taking advantage of the 3D geometric constraints, Yin et al. [17] implement ‘virtual norm’ constraints [12] and proposed a supervised framework to obtain a high-quality depth estimation. Praful et al. [21] utilize UW-GAN to estimate depth information, their network includes two modules: the generator predicts depth maps, and the discriminator determines the quality of the maps. Fu et al. [22] introduce a spacing-increasing discretization (SID) strategy to discretize depth and recast depth network learning as an ordinal regression problem to generate depth maps. Xu et al. [18] propose a conditional random field (CRF) based model for the multi-scale features to estimate the fine-grained depth maps. Although these fully connected network (FCN) based methods have achieved great success, there still exists some critical limitations, such as inconsistent labeling, losing or smoothing object details and requiring extra memory for holding a large part of parameters.

III. PROPOSED METHOD

We introduce the architecture of our attention-based encoder-decoder network and the design of loss function for the monocular depth estimation in this section.

A. Network Architecture

For an input RGB image $I_i$, our method is able to generate the corresponding depth map $D_i$ in an end-to-end fashion. As shown in Figure 2, our network mainly consists of encoder and decoder. The corresponding layers of encoder and decoder are connected by skip connections with CAMBs. We use DesNet-169 [23] without the last classification layer as encoder, which extracts high-resolution features and downsamples the input image. Our encoder is pretrained on ImageNet dataset [24]. The decoder in our network contains a straightforward up-scaling scheme, which simply upsamples the output of the previous layer to the same size as the output of the corresponding encoder layer after CAMB, then concatenate these two output feature together and performs a convolution operation.

B. Attention Module

Based on CBAM [25] whose lightweight property has been well proved, we design a more lightweight and effective
convolution attention mechanism block (CAMB) by and insert CAMBs into our model. Different from CBAM, our attention module utilize global power average pooling [26], which is formulated as (1) where \( R \) stands for the current feature map and \( p \) is the hyperparameter, and more simplified operations

\[
\hat{a} = \sqrt[p]{\sum_{i \in R} a_i^p},
\]

It should be pointed out that when we set \( p = 1 \) or \( \infty \), equation (1) actually represents sum pooling, which is proportional to average pooling, and max pooling. Specifically, we do not directly pass the output feature \( F_n \) of the encoder’s nth layer to the corresponding layer of the decoder through skip connections. Instead, we first feed \( F_n \) to CAMB, which has channel attention and spatial attention two sequential sub-modules. The former performs global power average pooling operations on \( F_n \), and passes the pooling results to a global shared three-layer fully connected DNN. Then execute the sigmoid function \( s(\cdot) \) to get the channel feature map \( F_{CAM_n} \). The latter uses the product of \( F_{CA_n} \) and \( F_n \) as input and performs power average pooling along the channel axis. Then we execute a convolution operation \( c(\cdot) \) with kernel size of \( 7 \times 7 \), which is experimentally determined, to generate a feature map and pass it to the sigmoid function \( s(\cdot) \) to get the final spatial attention map \( F_{SA_n} \). In the end, we multiply \( F_{CA_n} \) and \( F_{SA_n} \) to get the final attention feature \( F_{AM_n} \).

In order to add attention features on the basis of retaining the original features, we merge the final attention feature \( F_{AM_n} \) of CAMB with \( F_n \) using element-wise summation. The detail of this process is shown in Figure 3.

C. Loss Function

As illustrated in Figure, we design a novel loss function \( \mathcal{L} \), which consists of two main components

\[
\mathcal{L} = \lambda (\alpha \mathcal{L}_{\text{depth}} + \beta \mathcal{L}_{\text{grad}})
\]

where \( \mathcal{L}_{\text{depth}} \) is a variation of \( L_1 \) norm of the difference between ground truth and depth estimation, \( \mathcal{L}_{\text{grad}} \) stands for the gradient of adjacent pixel blocks, \( \lambda \) is a function of SSIM, \( \alpha \) and \( \beta \) are the hyperparameters.

\( L_1 \) norm of the difference between ground truth depth map \( y \) and the prediction \( \hat{y} \) is the most standard loss function for depth regression problems

\[
\mathcal{L}_1 = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|.
\]

where \( N \) stands for the total number of pixel. However, an issue of directly using \( L_1 \) norm is that the difference of each pixel, that is, \( |y_i - \hat{y}_i| \) for each \( i \), has an equal contribution to \( \mathcal{L}_1 \) between distant and nearby pixel [27]. For example, the error of 10cm should mean differently for objects at a distance of 1 meter and 10 meters. Follow [20], [28], we use a logarithmic variation of \( \mathcal{L}_1 \) to alleviate this issue

\[
\mathcal{L}_{\text{depth}} = \frac{1}{N} \sum_{i=1}^{N} F(|y_i - \hat{y}_i|)
\]

where \( F(x) = \ln(x + \theta) \), \( \theta \) is a hyperparameter. There are other methods to deal with this issue, such as using reciprocal of depth [29] and depth-balanced Euclidean loss [27].

In addition, in order to track the depth changing between adjacent pixels, we design \( \mathcal{L}_{\text{grad}} \)

\[
\mathcal{L}_{\text{grad}} = \frac{1}{N} \sum_{i=1}^{N} [F(\delta_x(y_i) - \delta_x(\hat{y}_i)) + F(\delta_y(y_i) - \delta_y(\hat{y}_i))

+ F(\delta_{\text{diag}}(y_i) - \delta_{\text{diag}}(\hat{y}_i))].
\]

where \( \delta_x \), \( \delta_y \) and \( \delta_{\text{diag}} \) stands for the mean of the adjacent pixel’ gradient of one image in \( x \), \( y \) and diagonal directions, respectively. Specifically, for each direction, the gradient is calculated in the same way, that is, for each pixel \( i \), calculate the difference between the value of the next pixel in the current direction and \( i \)’s value and then the gradient of this direction is the mean of all the differences. Different from previous work [28], [29], we originally incorporate the diagonal components into the gradient calculation. Considering the complexity of the shape of real-world objects, our novel \( \mathcal{L}_{\text{grad}} \) is able to further penalize small structural errors and improve fine details of depth maps. Apparently, this kind of loss is able to effectively reduce grid artifacts and blurry edges. However, one more dimension of computing increases the requirements for computing resources, which has a huge impact on resource-constrained areas. Therefore, we propose a trade-off by computing the gradient between adjacent pixel blocks, rather than a single pixel, for each pixel blocks \( b \times b \), we use the mean of each channel of all pixels in it to represent, where size \( b \) is a hyperparameter. Figure. 4 shows the detail of this strategy. It should be pointed out that [20], [28] also consider to further improve details, however, their methods rely on the expensive inner product of vectors, which is unaffordable for resource-constrained areas.

Lastly, we add a coefficient \( \lambda = 1 - \text{SSIM}(y, \hat{y}) \) to our overall loss function. The SSIM is a well-known quality metric used to measure the similarity between two images and is considered to be correlated with the quality perception of the human visual system [30]. SSIM(x, y) is a real number in the unit interval, and the larger its value is, the more similar the two images are. Therefore, we use \( \lambda \), instead of SSIM(x, y), in our loss function.

D. Data Augmentation

For better generalization in computer vision related tasks, data augmentation is necessary, which is an effective strategy to reduce over-fitting of CNNs [31]. Vertical flip and horizontal flip are most common strategies of data augmentation. Therefore, we execute vertical and horizontal flipping on
Fig. 3. The pipeline of CAMB attention module. For the output feature $F_n$ of $n$th layer of encoder, we perform channel attention operation and spatial attention operation in sequence, and then add the calculation result $F_{AM_n}$ to $F_n$ to obtain the final feature $F_{out_n}$ that combines the normal feature with attention feature.

Fig. 4. The strategy of calculating $L_{grad}$. (a) stands for the overall process of gradient calculation for one image: pixel block $b \times b$ slides from left to right, then from top to bottom and the step size is 1. The blue block and red block stands for the starting position and end position respectively. For each step, the calculation of each step is shown as (b). After computing all $\delta_x$, $\delta_y$, and $\delta_{diag}$, we calculate the mean of these three sets of gradient values. (c) is the traditional gradient calculation algorithm, that is, only considering the gradients in $x$ and $y$ directions and unit is fixed to one single pixel.

images at a probability of $\zeta$ and $\eta$ respectively, where $\zeta$ and $\eta$ are hyperparameters. As described in [29], despite image rotations and distortions are also common data augmentation methods, they introduce useless information for the ground-truth depth, such as unnecessary geometric interpretations and invalid data. Therefore, we do not include these two methods in our approach.

IV. EXPERIMENTS

We test the effectiveness of our approach separately on different datasets, and compare it with several representative baseline methods, which can represent the STOA. Then we provide the ablation study that evaluates the contribution of each component described in Section IV.

A. Datasets

We evaluate our method on KITTI and NYU-v2 datasets, which are the most commonly used datasets for monocular depth estimation in computer vision.

KITTI is an outdoor dataset for monocular deep estimation and object detection and tracking based on deep learning, which is captured through a car equipped with 2 high-resolution color cameras, 2 gray-scale cameras, laser scanner and global positioning system (GPS) and contains 93,000 training samples. The original image size is around 1,242 \times 375, and its ground-truth depth maps are sparse with a lot of missing data. Therefore, we execute inpainting method to fill the missing parts [32]. We use the training/testing sets split of Eigen et al. [16], which is the most standard method for KITTI splitting.

NYU-v2 focuses on the indoor scenes, which contains about 120K frames of RGB-D image pairs captured by a RGB camera and the Microsoft Kinect depth camera to simultaneously collect the RGB and depth information. The original image size is 640 \times 480. Similar with KITTI, we also execute inpainting method to fill missing depth values.
We follow the official training/testing split, which uses 249 scenes for training and 215 scenes (654 images) for testing. From the total 120K image-depth pairs, we train our model on a 50K subset as [29].

**B. Hyperparameters**

In our experiments, we set $\zeta = 0.3$ and $\eta = 0.3$ for the probabilities of vertical and horizontal flipping in data augmentation. We use the ADAM optimizer with learning rate $0.0001$ and the batch size is set to 8. For effectively combining augmentation, we use the ADAM optimizer with learning rate $\beta$ and $\delta$ used in Equation 2, we experimentally set $\alpha = 1$ and $\beta = 0.8$. We set $\theta = 0.5$ for the logarithmic function $F(\cdot)$ used in Equation 4. As for the block size $b$, we set it to 2 according to comprehensive experiments. We set $p = 3$ for power average pooling.

**C. Analysis**

**Quantitative comparison.** In Table I and Table II, we compare the proposed algorithm with the recent SOTA algorithms [14], [18], [22], [29], [33], [34] and pioneering work [16] in monocular depth estimation on NYU-V2 and KITTI dataset quantitatively, which is able to fully demonstrate the effectiveness of our method.

Specifically, compared with [16], our approach achieves significant improvement in all metrics on both KITTI and NYU-V2 datasets. Especially for KITTI, the error metrics is almost halved and RMSE is even reduced by 4.508. As compared with other methods, the overall performance of our approach is still superior. For NYU-V2 dataset, our approach is first place except $\delta_1$ and abs.rel are second place. As for KITTI, our approach shows the best performance in $\delta_1$, $\delta_3$, RMSE, log.rel and abs.sql. In $\delta_2$ and sqrl.rel, although our method only won the second place, the difference between the first place is tiny (0.001 in $\delta_2$ and 0.006 in sqrl.rel).

**Qualitative comparison.** Figure 5 compares depth maps qualitatively. For better visualizations, we transfer original depth maps to color map by calling a toolbox in matplotliblib. It is observed that our proposed approach estimates the depth maps reliably and accurately and also reduce grid and blurry edging artifacts in comparison with the other approaches.

**V. CONCLUSION**

This paper explores the problem of monocular depth estimation based on single image, which is the most extreme but alluring case in vision-related depth estimation. We propose an attention-based encoder-decoder network with a novel loss function to effectively address this problem. Specifically, we insert lightweight attention module CAMB into skip connections between encoder and decoder to find focuses for reducing grid artifacts and blurry edges with least possible overhead. In addition, our loss function is designed by combining the difference of depth values, gradient in three directions ($x$, $y$ and diagonal) and SSIM to further improve fine details of depth maps and penalize small structural errors. For speeding up loss computing, we utilize flexible pixel blocks as units of computation instead of single pixel. Comprehensive experiments on two large-scale datasets show that our approach outperforms several representative baseline methods.

**REFERENCES**

[1] J. Van Brummelen, M. O’Brien, D. Grayer, and H. Najjaran, “Autonomous vehicle perception: The technology of today and tomorrow,” Transportation research part C: emerging technologies, vol. 89, pp. 384–406, 2018.

[2] F. El Jamiy and R. Marsh, “Survey on depth perception in head mounted displays: distance estimation in virtual reality, augmented reality, and mixed reality,” IET Image Processing, vol. 13, no. 5, pp. 707–712, 2019.

[3] T. Zhe, L. Huang, Q. Wu, J. Zhang, C. Pei, and L. Li, “Inter-vehicle distance estimation method based on monocular vision using 3d detection,” IEEE transactions on vehicular technology, vol. 69, no. 5, pp. 4907–4919, 2020.

[4] M. Hwang, D. Seita, B. Thananjeyan, J. Ichinowski, S. Paradis, D. Fer, T. Low, and K. Goldberg, “Applying depth-sensing to automated surgical manipulation with a da vinci robot,” in 2020 International Symposium on Medical Robotics (ISMIR). IEEE, 2020, pp. 22–29.

[5] Y. Ming, X. Meng, C. Fan, and H. Yu, “Deep learning for monocular depth estimation: A review,” Neurocomputing, vol. 438, pp. 14–33, 2021.

[6] R. Daniels, E. R. Yeh, and R. W. Heath, “Forward collision vehicular radar with ieee 802.11: Feasibility demonstration through measurements,” IEEE Transactions on Vehicular Technology, vol. 67, no. 2, pp. 1404–1416, 2017.

[7] J. Zhang, Q. Su, C. Wang, and H. Gu, “Monocular 3d vehicle detection with multi-instance depth and geometry reasoning for autonomous driving,” Neurocomputing, vol. 403, pp. 182–192, 2020.

[8] A. Badki, O. Gallo, J. Kautz, and P. Sen, “Binary ttc: A temporal geofence for autonomous navigation,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12 946-12 955.

[9] C. Zhao, Q. Sun, C. Zhang, Y. Tang, and F. Qian, “Monocular depth estimation based on deep learning: An overview,” Science China Technological Sciences, pp. 1–16, 2020.

[10] J. L. Schonberger and J.-M. Frahm, “Structure-from-motion revisited,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4104–4113.

Fig. 5. Qualitative comparison of estimated depth maps. We select three representative scenarios (office, classroom with students and kitchen) from the testing set of NYU-V2 for intuitively reflecting the effectiveness of our approach.
TABLE I

COMPARISONS OF DIFFERENT METHODS ON NYU. IN EACH COLUMN WE BOLD THE BEST PERFORMING METHOD AND UNDERLINE THE SECOND-BEST.

| Methods          | $\delta_1$ | $\delta_2$ | $\delta_3$ | RMSE | log. rel | abs. rel |
|------------------|------------|------------|------------|------|----------|----------|
| Eigen at al. [16] | 0.786      | 0.930      | 0.988      | 0.641| 0.158    |          |
| Fu et al. [22]   | 0.828      | 0.965      | 0.992      | 0.569| 0.115    |          |
| Alhashim et al. [29] | 0.846      | 0.974      | 0.990      | 0.465| 0.053    | 0.123    |
| Laina et al. [33] | 0.811      | 0.953      | 0.988      | 0.573| 0.055    | 0.127    |
| Hao et al. [34]  | 0.841      | 0.966      | 0.991      | 0.555| 0.053    | 0.127    |
| MS-CRF et al. [18] | 0.811      | 0.954      | 0.987      | 0.586| 0.052    | 0.121    |
| Ours             | 0.855      | 0.980      | 0.994      | 0.441| 0.047    | 0.107    |

TABLE II

COMPARISONS OF DIFFERENT METHODS ON KITTI. IN EACH COLUMN WE BOLD THE BEST PERFORMING METHOD AND UNDERLINE THE SECOND-BEST.

| Methods          | $\delta_1$ | $\delta_2$ | $\delta_3$ | RMSE | log. rel | abs. rel | sq. rel |
|------------------|------------|------------|------------|------|----------|----------|--------|
| Godat et al. [14] | 0.861      | 0.949      | 0.976      | 4.935| 0.206    | 0.114    | 0.898  |
| Eigen at al. [16] | 0.692      | 0.899      | 0.967      | 7.156| 0.270    | 0.190    | 1.515  |
| Kuznetsova et al. [35] | 0.862      | 0.960      | 0.986      | 4.621| 0.189    | 0.113    | 0.741  |
| Alhashim et al. [29] | 0.886      | 0.965      | 0.986      | 4.170| 0.171    | 0.093    | 0.589  |
| Fu et al. [22]   | 0.952      | 0.984      | 0.994      | 2.727| 0.120    | 0.072    | 0.307  |
| Ours             | 0.947      | 0.989      | 0.996      | 2.548| 0.113    | 0.061    | 0.297  |

[11] D. Scharstein and C. Pal, “Learning conditional random fields for stereo,” in 2007 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2007, pp. 1–8.
[12] F. Khan, S. Salahuddin, and H. Javidnia, “Deep learning-based monocular depth estimation methods—a state-of-the-art review,” Sensors, vol. 20, no. 8, p. 2270, 2020.
[13] A. Sabnis and L. Vachhani, “Single image based depth estimation for robotic applications,” in 2011 IEEE Recent Advances in Intelligent Computational Systems. IEEE, 2011, pp. 102–106.
[14] 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, 2017, pp. 270–279.
[15] Z. Yin and J. Shi, “Geonet: Unsupervised learning of dense depth, optical flow and camera pose,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1983–1992.
[16] D. Eigen, C. Puhrsch, and R. Fergus, “Depth map prediction from a single image using a multi-scale deep network,” Advances in neural information processing systems, vol. 27, 2014.
[17] W. Yin, Y. Liu, C. Shen, and Y. Yan, “Enforcing geometric constraints of virtual normal for depth prediction,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 5684–5693.
[18] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe, “Multi-scale continuous crfs as sequential deep networks for monocular depth estimation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 5354–5362.
[19] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” arXiv preprint arXiv:1706.05587, 2017.
[20] J. Hu, M. Oztay, Y. Zhang, and T. Okatani, “Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries,” in 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2019, pp. 1043–1051.
[21] P. Huber, S. Murala, and A. Dhali, “Uw-gan: Single-image depth estimation and image enhancement for underwater images,” IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1–12, 2021.
[22] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, “Deep ordinal regression network for monocular depth estimation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2002–2011.
[23] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.
[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, pp. 248–255.
[25] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “Cham: Convolutional block attention module,” in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 3–19.
[26] J. B. Estrach, A. Szlam, and Y. LeCun, “Signal recovery from pooling representations,” in International conference on machine learning, PMLR, 2014, pp. 307–315.
[27] J.-H. Lee, M. Heo, K.-R. Kim, and C.-S. Kim, “Single-image depth estimation based on fourier domain analysis,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 330–339.
[28] L. Huyhn, P. Nguyen-Ha, J. Matas, E. Rahtu, and J. Heikkilä, “Guiding monocular depth estimation using depth-attention volume,” in European Conference on Computer Vision. Springer, 2020, pp. 581–597.
[29] I. Alhashim and P. Wonka, “High quality monocular depth estimation via transfer learning,” arXiv preprint arXiv:1812.11941, 2018.
[30] Y. Chen, H. Zhao, Z. Hu, and J. Peng, “Attention-based context aggregation network for monocular depth estimation,” International Journal of Machine Learning and Cybernetics, vol. 12, no. 6, pp. 1583–1596, 2021.
[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, vol. 25, 2012.
[32] 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. IEEE, 2012, pp. 3354–3361.
[33] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” in 2016 Fourth international conference on 3D vision (3DV). IEEE, 2016, pp. 239–248.
[34] 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). IEEE, 2018, pp. 304–313.
[35] Y. Kuznetsova, J. Stuckler, and B. Leibe, “Semi-supervised deep learning for monocular depth map prediction,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 6647–6655.