CaBins: CLIP-based Adaptive Bins for Monocular Depth Estimation

Eunjin Son   Sang Jun Lee*
Jeonbuk National University, Republic of Korea
{eunjinson, sj.lee}@jbnu.ac.kr

Abstract

Traditional deep-learning models use pre-trained knowledge on large-scale datasets to fine-tune the model. This strategy significantly improves the performance of downstream tasks such as object detection and segmentation. Recently, vision-language (VL) models that jointly train an image encoder and a text encoder have gained attention. Notably, CLIP, which employs contrastive learning for classification, contributed significantly to establishing the foundation for the VL model paradigm. In depth estimation, several CLIP-based models have been proposed that use images and texts called semantic bins. However, it is questionable whether these human-set semantic bins are reasonable. In this work, we propose a network for monocular depth estimation, leveraging CLIP's pre-trained knowledge. Our model employs a regression-classification formulation, predicting depth through a linear combination of depth candidates and a probability map derived from the similarity score between image embedding and text embedding. Unlike previous works relying on human-set semantic bins for the text embedding, our model converts the predicted depth candidates into distance classes using the CaBins module. Moreover, we modify CLIP's image encoder, which is designed for classification, to address the dense prediction task. Experiments were conducted on the NYU-Depth V2 and KITTI datasets. We compared the performance of our model with CLIP-based as well as unimodal monocular depth estimation models. Our proposed model outperformed previous CLIP-based models across all evaluation metrics and showed high-quality boundary predictions on both datasets. Our model is available at https://github.com/EunjinSon1/CaBins.

1. Introduction

Deep-learning models leverage pre-trained knowledge on large-scale datasets to fine-tune the model. This process effectively enhances the model's performance for the majority of downstream tasks. In traditional computer vision task, the network extracts features from an encoder pre-trained on large-scale only image datasets such as ImageNet [10] or MS-COCO [29]. The pre-training strategy of these vision models has demonstrated successful performance improvement across various vision tasks, including object detection [7, 30, 40], segmentation [3, 9, 34, 49], and depth estimation [12, 15, 25, 33].

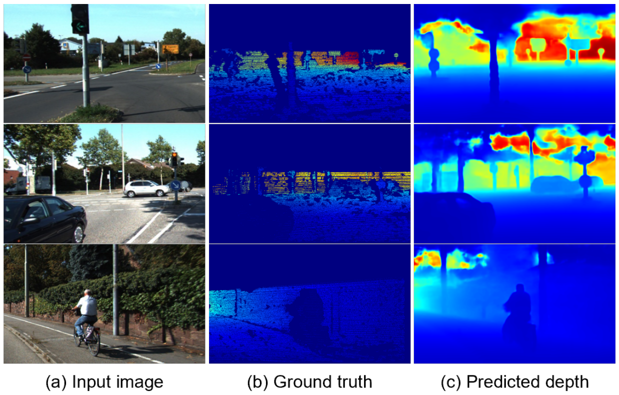

Figure 1. Visualization of the predicted depth map of our network on the KITTI dataset. (a) indicates input images. (b) is ground truths. (c) is the depth maps of our network.

Meanwhile, vision-language (VL) models, which learn vision representations through joint training of an image encoder and a text encoder, have started to attract attention [21, 37]. Notably, CLIP [37] demonstrated impressive classification performance without fine-tuning by pre-training on a dataset of 400 million web-based image-text pairs through contrastive learning. Subsequently, various downstream tasks, such as image generation [16, 47, 48, 51] and video-text retrieval [22, 26, 46, 53], exhibited significant performance improvements by leveraging CLIP's pre-trained knowledge. In particular, these VL models have proven to be effective for zero-shot models because they learn connections between visual and language information [4, 14, 35, 45].

However, CLIP's pre-trained knowledge does not adapt well to all downstream tasks. CLIP is a model specialized for classification problems and its architecture is designed to focus on global information. Therefore, it has not been well explored in tasks requiring dense prediction, where local information is important. Rao *et al*. [39] tackled per-pixel problems such as semantic segmentation and detection by redefining CLIP's image-text matching problem as a pixel-text matching problem. Additionally, CLIP is pre-trained on paired datasets of images and texts such as "`A photo of a [Object].`". The pre-trained knowledge from these datasets has limitations in grasping abstract concepts like depth clues. Consequently, VL models still face challenges in handling complex and abstract tasks, such as depth estimation and counting.

Recently, a zero-shot depth estimation model [52] was proposed, leveraging the pre-trained knowledge of CLIP. DepthCLIP [52] calculates the similarity score between image features and semantic bins' features to predict depth, where semantic bins denote words representing distance scales (e.g., ['giant', 'close', ..., 'unseen']). Their contribution implies the potential for expanding the field of depth estimation, suggesting that VL models can be adapted to monocular depth estimation (MDE) through matching texts for distance scale with images. However, it must utilize predetermined fixed depth values, called quantified bins, to obtain the final depth. Furthermore, there is uncertainty about the appropriateness of the human-set semantic bins and how detailed they should be. Subsequently, several CLIP-based depth estimation works [2, 20, 23] inspired by DepthCLIP have been proposed. However, the fundamental question of semantic bins remains unresolved, leading to a limitation where the estimated depth map appears quite blurry.

In this paper, we propose a novel CLIP-based monocular depth estimation model and compare its performance with other models. Inspired by AdaBins [5], we adopt a regression-classification formulation to predict depth, wherein the depth map is estimated as a linear combination of depth candidates and a probability map. Our work stems from questions about the human-set semantic bins employed in previous works. Instead of using human-set semantic bins, our model uses bin centers, which are depth candidates estimated from an image encoder, as text prompt. We partition all bin centers into a few groups and take one from each group, considering the distribution of bin center values. These obtained values are referred to as CaBins. Furthermore, to address the dense prediction problem, we modify CLIP's image encoder to extract multi-scale features. To demonstrate the effectiveness of our model, it was evaluated on the NYU-Depth V2 [43] and KITTI [18] datasets. Our model outperformed the previous CLIP-based MDE models and achieved comparable results to unimodal MDE models. Additionally, to verify the effectiveness of our method for text prompt, we compare the results across various text prompt. We hope that our work will contribute to the advancement of MDE techniques, especially VL-based MDE.

To summarize, our main contributions are

- We propose a novel model for CLIP-based monocular depth estimation. Unlike previous works that rely on human-set semantic bins, we use estimated bin centers called CaBins as text prompt.
- We modified CLIP's image encoder to extract multi-scale features for addressing the dense prediction task.
- To demonstrate the effectiveness of our model, we conducted experiments on NYU-Depth V2 and KITTI datasets, respectively. Our model outperformed the previous CLIP-based MDE models.

2. Related work

2.1. Monocular Depth Estimation

MDE is the task of estimating pixel-level distance information for a single RGB image. This task is essential for various applications that recognize 3D environments, such as autonomous driving or robotics. There are three major formulations for MDE: Regression, Classification, and Regression-Classification. Regression is a basic framework for MDE, wherein the network is trained through a regression loss function that measures pixel-wise disparity between the predicted depth map and the ground truth. Eigen *et al*. [12] introduced the first deep learning-based depth estimation network and proposed the Scale-Invariant Logarithmic loss, which has been widely adopted as the regression loss function for depth estimation tasks. Lee *et al*. [25] proposed a local planer guidance layer that uses plane coefficients to estimate depth. Ranftl *et al*. [38] proposed a ViT [11]-based depth estimation model instead of the traditional CNN-based backbone. There has been a consistent trend of regression-based depth estimation models [31, 32, 36]. On the other hand, Fu *et al*. [15] introduced a classification approach that predicts optimal depth candidates by dividing the depth range into several candidates on either a uniform or logarithmic scale. While this approach has shown significant performance improvement, it suffers from poor visual quality due to depth discretization. Therefore, Bhat *et al*. [5] aimed to improve the MDE performance by integrating regression and classification approaches. Specifically, they combined a regression method that learns a probability distribution with a classification method that generates N depth candidates. In addition, they proposed an adaptive bins method that takes into account the image-dependent depth distribution, allowing depth candidates to be adaptively predicted for the image instead of uniform division. Building upon the idea of [5], subsequent works [1, 6, 27, 42] have been proposed, including methods for predicting depth candidates at the pixel level [6] and adjusting bin-widths based on uncertainty [42].

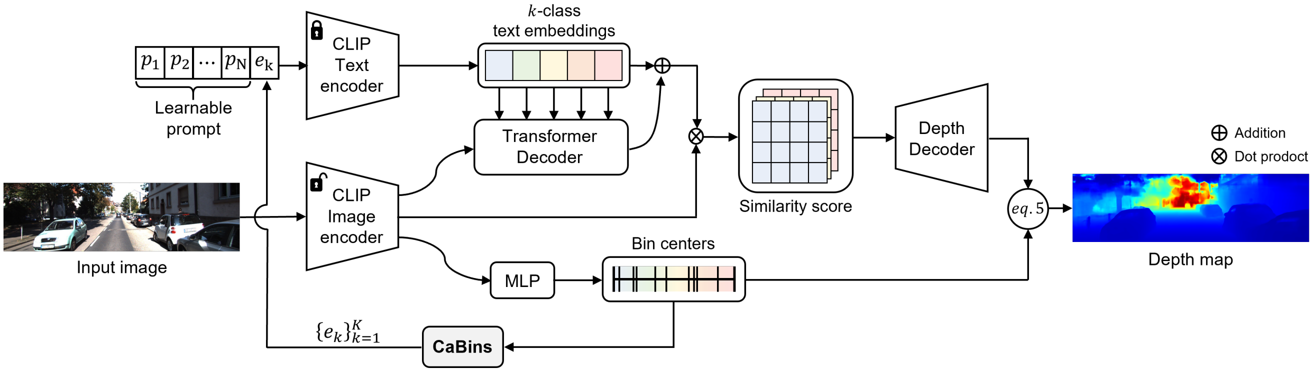

Figure 2. An overview of the architecture of the proposed network. The network takes an RGB image to output an image embedding and bin centers. These bin centers are converted into K CaBins through the CaBins module, where CaBin e_k represents the depth value computed by weighted summation of n_g bin centers in the k-th group. Subsequently, they are concatenated with learnable prompt $[p_1, ..., p_N]$ and fed into the text encoder. The text embedding is updated by the transformer decoder using image embedding. A similarity score is calculated through the dot product between the two embeddings, then fed into the depth decoder to predict the probability map. Finally, the depth map is computed as a linear combination of the bin centers and the probability map.

2.2. Vision-Language Models

The integration of pre-trained VL models into vision tasks has demonstrated successful performance improvements [21, 37]. Particularly, CLIP [37] significantly improved classification performance through the pre-training on a large-scale raw paired dataset using a contrastive learning method, thereby establishing the foundation for the VL pre-training paradigm. They employ a prompt template (e.g., "`A photo of a [Object].`") to enrich contextual information. Zhou *et al.* [54] addressed the constraints of human-designed prompts by introducing prompt learning that utilizes the concatenation of learnable vectors and object labels for text prompt generation. Kwon *et al.* [24] proposed a probabilistic prompt learning method that utilizes multiple attribute prompts for a single image, considering the randomness of the visual-context. Although adapting VL models to vision tasks has led to significant advancements, such works have primarily designed for limited tasks such as classification and segmentation.

2.3. CLIP-based Depth Estimation

Recently, several works have been proposed to adapt VL models in the field of depth estimation [2, 20, 23, 52]. Initially, Zhang *et al.* [52] proposed zero-shot depth estimation model, leveraging CLIP's pre-trained knowledge. They define seven human-set semantic bins (e.g., ['giant', ' extremely close', 'close', 'not in distance', 'a little remote', 'far', 'unseen']) to generate prompts such as "`This object is [semantic bin].`". These prompts are fed into the text encoder to output a text embedding, and the depth weights are calculated through the dot product between it and a image embedding, followed by a softmax operation. Subsequently, the final depth map is predicted through a linear combination of the depth weights and the quantified depth bins, which are fixed depth values set in advance. Hu *et al.* [20] improved DepthCLIP [52] by considering that the depth distribution is different for each scene. They design a learnable depth codebook by training on a single image from each scene category and store the learned quantified depth bins for each scene. Auty [2] utilize learnable tokens instead of semantic bins, but has to rely on predetermined fixed depth values for the final depth prediction. While these works have shown the applicability of VL models in depth estimation, there are still limitations, including fundamental questions about human-set semantic bins and issues with blurry output.

3. Method

In this section, to facilitate a better understanding of our model, we first introduce the architecture of CLIP, specifically its ResNet-based image encoder and text encoder. Subsequently, we present our model and training loss. An overview of the architecture of our proposed network is shown in Figure 2.

3.1. CLIP

CLIP consists of a ResNet [19] or ViT [11]-based image encoder and a Transformer [44]-based text encoder. Each encoder outputs embedding vectors for their corresponding input. For the ResNet-based image encoder, the encoder consists of the standard ResNet architecture and an additional Multi-Head-Attention (MHA) module. Specifically, the input image I is fed into the ResNet architecture, resulting in a feature map $\mathbf{x}$ with a stride of 32 and 2048 channels. This feature map $\mathbf{x}$ is then flattened into a vector and concatenated with a global token $\overline{\mathbf{x}}$ having 2048 channels. The global token $\overline{\mathbf{x}}$ is obtained as the average of $\mathbf{x}$ and serves as a global image representation. Subsequently, the input embedding $[\overline{\mathbf{x}}, \mathbf{x}]$ with the positional embedding added is fed into the MHA module, where an attention mechanism is applied with $\overline{\mathbf{x}}$ as the query and $[\overline{\mathbf{x}}, \mathbf{x}]$ as the key and value. The final output of the image encoder is an embedding vector $\overline{\mathbf{z}}$ with D channels.

For the text encoder, CLIP constructs the text prompts "`A photo of a [Object].`" for K object classes. Subsequently, a tokenizer assigns an integer to each token of the text prompt using a lower-cased byte pair encoding (BPE), resulting in a vector of size $K \times 77$. Where 77 represents the maximum sequence length, including `[SOS]` and `[EOS]` tokens. This vector is converted into an embedding vector of size $K \times 77 \times 512$ through an embedding function. Positional embedding is then added before it is fed into the transformer. The final text embedding $\mathbf{t} \in \mathbb{R}^{K \times D}$ is obtained by projecting only the embedding vector corresponding to the `[EOS]` token of the transformer's output embedding into D dimensions. D is set to 1024 in ResNet-50 and 512 in ResNet-101.

3.2. CaBins

3.2.1 Image Encoder

We adopt ResNet-101 as the image encoder. CLIP concentrates on extracting global information from the image for classification. However, in dense prediction tasks such as MDE, incorporating both global and local information is crucial for achieving accurate depth estimation. Therefore, we make some modifications to the CLIP's ResNet-101-based image encoder, enabling the extraction of multi-scale features. Specifically, for an input image I, the input embedding $[\overline{\mathbf{x}}, \mathbf{x}]$ is obtained by the same process described in Section 3.1. This embedding vector $[\overline{\mathbf{x}}, \mathbf{x}]$ is then fed into the MHA module. Note that, unlike CLIP, the MHA module apply a self-attention mechanism with $[\overline{\mathbf{x}}, \mathbf{x}]$ as a query and generates an output embedding $[\overline{\mathbf{z}}, \mathbf{z}]$ with 512 channels.

Subsequently, an MLP with ReLU activation function is applied to the global token $\overline{\mathbf{z}}$ to generate an n_{bins} dimensional vector $\mathbf{b}'$. This vector $\mathbf{b}'$ is then normalized, summing up to 1, to obtain a bin-width vector $\mathbf{b}$. The final bin center vector $c(\mathbf{b})$ is computed as follows:

$$c(b_i) = d_{min} + (d_{max} - d_{min}) \left(\frac{b_i}{2} + \sum_{j=1}^{i-1} b_j \right), \quad i \in \{1, ..., n_{bins}\}, \tag{1}$$

where d_{min} and d_{max} denote the minimum and maximum values of the depth range, respectively, and n_{bins} is the number of bins, which is set to 256 as in AdaBins [5]. Moreover, we utilize the encoded features $\{\mathbf{x}_i\}_{i=1}^{4}$ with strides of 4, 8, 16, and 32 from each stage of ResNet-101 in the depth decoder.

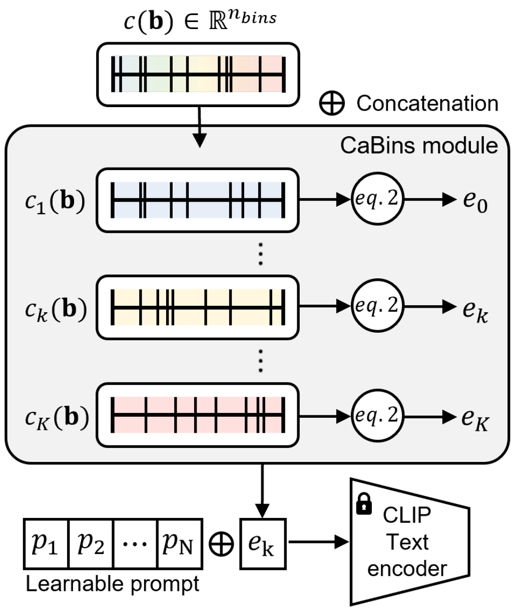

Figure 3. Our proposed CaBins module. Given n_{bins} bin centers, the CaBins module partitions them into K groups. In the k-th group, the CaBin e_k is computed as a weighted summation. The weight is defined by applying the softmax operation to the inverse of the deviation from the mean, as shown in Equation 2. Finally, the CaBins are concatenated with the learnable prompts and fed into the text encoder.

3.2.2 CaBins

We construct the text prompt based on the predicted bin centers. As shown in Figure 3, the bin center vector $c(\mathbf{b})$ from the image encoder is grouped into adjacent bins to generate K groups $\{c_k(\mathbf{b})\}_{k=1}^{K}$, where K is set to 8. Subsequently, we take one value from each group to serve as the distance class e_k for the text prompt. To assign greater weight to bin center values that are closer to the mean in the k-th group, the weight w and distance class e_k are defined as follows:

$$\begin{gathered} e_k = \sum_{i}^{n_g} w_i c_k(b_i), \\ w_i = \text{softmax}\left(\frac{1}{|c_k(b_i) - m_k| + \epsilon} \right), \end{gathered} \tag{2}$$

where n_g and m_k represent the number and mean of bin centers $c_k(b_i)$ in the k-th group. e_k is the k-th distance class computed by the weighted summation of n_g bin centers, and the resulting K distance classes are referred to as K CaBins. ϵ is set to $1e-4$, a small positive number to ensure numerical stability. To prove the efficiency of our method, we conducted experiments using various methods, which are covered in Section 4.6.

3.2.3 Text Encoder

We use CLIP's pre-trained transformer-based text encoder. The text encoder is frozen during training. Unlike previous works [20, 52] that use seven semantic bins, we employ bin centers estimated from the image encoder for text prompt. To construct text prompts, the n_{bins} bin centers are passed into the CaBins module to generate K CaBins, as detailed in Section 3.2.2. These resulting CaBins $\{e_k\}_{k=1}^K$ are concatenated with the learnable prompts $\mathbf{p} \in \mathbb{R}^{N\times 512}$ introduced by CoOp [54] and then fed into the text encoder. Here, N represents the context length of the text prompt, which is set to 8 in our case. Following the same process as CLIP, the text encoder outputs a text embedding $\mathbf{t} \in \mathbb{R}^{K\times 512}$. Subsequently, inspired by DenseCLIP [39], the text embedding $\mathbf{t}$ is updated through a transformer decoder using $[\bar{\mathbf{z}}, \mathbf{z}]$ as key and value, with $\mathbf{t}$ as query. This process ensures a more comprehensive integration of visual context into the text embedding:

$$\begin{aligned} \mathbf{t} &\leftarrow \mathbf{t} + \gamma \mathbf{v_{post}}, \\ \mathbf{v_{post}} &= \text{TransDecoder}(\mathbf{t}, [\bar{\mathbf{z}}, \mathbf{z}]), \end{aligned} \tag{3}$$

where $\gamma \in \mathbb{R}^{512}$ is a learnable parameter, initialized to 10^{-4}.

3.2.4 Depth Prediction

To match the visual representation with the linguistic representation for the depth value, a similarity score between the image and text embedding is calculated through the dot product:

$$\mathbf{s} = \hat{\mathbf{z}} \cdot \hat{\mathbf{t}}, \tag{4}$$

where $\hat{\mathbf{z}}$ and $\hat{\mathbf{t}}$ mean $\mathbf{z}$ and $\mathbf{t}$ normalized in the channel direction, respectively. The concatenation of $\mathbf{s}$ and $\mathbf{x}_4$ is fed into a depth decoder and used to predict a probability map. The depth decoder consists of simple upsampling layers and channel reduction layers. At each stage, the encoded features $\{\mathbf{x}_i\}_{i=1}^3$ are used as skip-connections to generate a feature map with a shape of $H/2 \times W/2 \times n_{bins}$. The probability map $\mathbf{p}$ is predicted through a 1×1 convolution followed by a softmax function on this feature map. The depth d_i at pixel i is predicted through a linear combination of the bin center vector $c\,(\mathbf{b})$ and the probability map $\mathbf{p}$:

$$d_i = \sum_{k=1}^{n_{bins}} c\,(b_k)\, p_{ik}, \tag{5}$$

where $c\,(b_k)$ is the k-th bin center, and p_{ik} is the probability for the k-th bin center at pixel i. Finally, our depth map is obtained by upsampling to the original resolution through bilinear interpolation.

3.3. Training Loss

Pixel-level loss. Following previous works [5, 25], we use the Scale-Invariant Logarithmic (SILog) loss introduced by Eigen *et al.* [12] to train the network:

$$\mathcal{L}_{pixel} = \alpha \sqrt{\frac{1}{n}\sum_i g_i^2 - \frac{\lambda}{n^2}\left(\sum_i g_i\right)^2}\,, \tag{6}$$

where $g_i = \log d_i - \log d_i^*$ is the logarithmic distance between the predicted depth map d and the ground truth d^* at the i-th pixel and n is the number of valid pixels in the ground truth. We set $\lambda = 0.85$ and $\alpha = 10$ in all experiments, respectively.

Bins loss. Following [5], the adaptive bins are supervised using bi-directional Chamfer loss [13], which encourages the two sets of values to be similar:

$$\mathcal{L}_{bins} = chamfer\,(X, c\,(\mathbf{b})) + chamfer\,(c\,(\mathbf{b})\,, X)\,, \tag{7}$$

where X is a set of ground truth depth values and $c\,(\mathbf{b})$ is a set of predicted bin center values.

Total loss. Finally, the total loss is defined as:

$$\mathcal{L}_{total} = \mathcal{L}_{pixel} + \beta \mathcal{L}_{bins}, \tag{8}$$

where we set $\beta = 0.1$ for all experiments in this work.

4. Experiments

4.1. Implementation Details

The proposed algorithm was implemented using a Nvidia GeForce RTX 3090 GPU hardware environment and PyTorch framework. For training, we adopt the AdamW optimizer with a weight decay of 0.1 and a 1-cycle policy with the maximum learning rate of $3.57e-5$. The network is trained for 25 epochs with a batch size of 8 in all experiments. We evaluate the proposed model on the NYU-Depth V2 indoor dataset and the KITTI outdoor dataset, respectively.

4.2. Datasets

NYU-Depth V2 dataset. The NYU-Depth V2 dataset is a dataset composed of video sequences captured from 464 different indoor scenes using RGBD camera. It consists of a total of 120K RGB image and depth pairs with a resolution of 640×480. Depth values range up to 10 meters. Following previous work [25], we use 24,231 images for training and 654 images for testing. Additionally, we apply a predefined cropping by Eigen *et al.* [12].

KITTI dataset. The KITTI dataset is an outdoor road driving dataset consisting of stereo images and corresponding

Method	Approach	AbsRel↓	RMSE↓	log 10↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Make3D [41]	Unimodal	0.349	1.214	-	0.447	0.745	0.897
DORN [15]		0.115	0.509	0.051	0.828	0.965	0.992
ASTransformer [8]		0.103	0.374	0.044	0.902	0.985	0.997
DepthFormer [28]		0.096	0.339	0.041	0.921	0.989	0.998
NeWCRFs [50]		0.095	0.334	0.041	0.922	0.992	0.998
DepthCLIP [52]	CLIP-based	0.388	1.167	0.156	0.394	0.683	0.851
Hu *et al.* [20]		0.347	1.049	0.140	0.428	0.732	0.898
Auty† [2]		0.324	0.961	0.127	0.473	0.779	0.921
Ours		**0.120**	**0.401**	**0.050**	**0.866**	**0.978**	**0.996**

Table 1. Performance comparison on NYU-Depth V2 dataset. A unimodal approach refers to training the model only on the image dataset. ↓ denotes that lower values are preferable, while ↑ denotes that higher values are preferable. The symbol † means reimplementation.

Method	Approach	AbsRel↓	SqRel↓	RMSE↓	RMSE log↓	log 10↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
DORN [15]	Unimodal	0.072	0.307	2.727	0.120	-	0.932	0.984	0.994
ASTransformer [8]		0.058	-	2.685	0.089	-	0.963	0.995	0.999
DepthFormer [28]		0.052	0.158	2.143	0.079	-	0.975	0.997	0.999
NeWCRFs [50]		0.052	0.155	2.129	0.079	-	0.974	0.997	0.999
DepthCLIP [52]	CLIP-based	0.473	6.007	12.958	-	0.680	0.281	0.531	0.696
Hu *et al.* [20]		0.384	4.661	12.290	-	0.632	0.312	0.569	0.739
Auty† [2]		0.307	2.197	6.405	0.121	0.113	0.548	0.826	0.935
CLIP2Depth [23]		0.074	0.303	2.948	-	0.032	0.938	0.990	0.998
Ours		**0.057**	**0.186**	**2.322**	**0.088**	**0.025**	**0.964**	**0.995**	**0.999**

Table 2. Performance comparison on the Eigen split of KITTI dataset. A unimodal approach refers to training the model only on the image dataset. ↓ denotes that lower values are preferable, while ↑ denotes that higher values are preferable. The symbol † means reimplementation.

Velodyne LiDAR scans. The RGB images have an average resolution of 1241×376 pixels. Depth maps are created by projecting LiDAR points and have a maximum depth value of 80 meters. Following the standard Eigen split [12], we use 23,158 images for training and 697 images for testing. We apply a pre-defined cropping proposed by Garg *et al.* [17].

4.3. Evaluation Metrics

We evaluate the quantitative performance of the proposed model using standard metrics used in previous works [25, 50]. For the accuracy metrics, the ratio of pixels with relative errors lower than a threshold value is used: $\%$ of d_i s.t. $\max(\frac{d_i}{d_i^*}, \frac{d_i^*}{d_i}) = \delta < thr$, for $thr = 1.25, 1.25^2, 1.25^3$. For the error metrics, the following metrics are used: absolute relative error (AbsRel): $\frac{1}{n}\sum_{d\in n}|d - d^*|/d^*$; squared relative error (SqRel): $\frac{1}{n}\sum_{d\in n}\|d - d^*\|^2/d^*$; root-mean-squared error (RMSE): $\sqrt{\frac{1}{n}\sum_{d\in n}\|d - d^*\|^2}$; root-mean-squared logarithmic error (RMSE log): $\sqrt{\frac{1}{n}\sum_{d\in n}\|\log d - \log d^*\|^2}$; average log error (log 10): $\frac{1}{n}\sum_{d\in n}|\log_{10} d - \log_{10} d^*|$, where d is the predicted depth map, d^* is the ground truth, and n denotes a total number of available pixels in the ground truth.

4.4. Experimental Results on NYU-Depth V2

Table 1 and Figure 4 show the quantitative and qualitative results on the NYU-Depth V2 dataset, respectively. As shown in Table 1, our model exhibited improved performance over CLIP-based depth models. Specifically, it achieved reductions of 62.96% and 58.27% in the AbsRel and RMSE error metrics, respectively, compared to Auty [2]. In Figure 4, it demonstrated outstanding results in predicting object boundaries and capturing small objects such as the faucet. However, our model tends to underestimate the depth range.

4.5. Experimental Results on KITTI

Table 2 and Figure 5 show the quantitative and qualitative results on the KITTI dataset, respectively. As shown in Table 2, our model outperforms all previous CLIP-based models by a significant margin across all evaluation met-

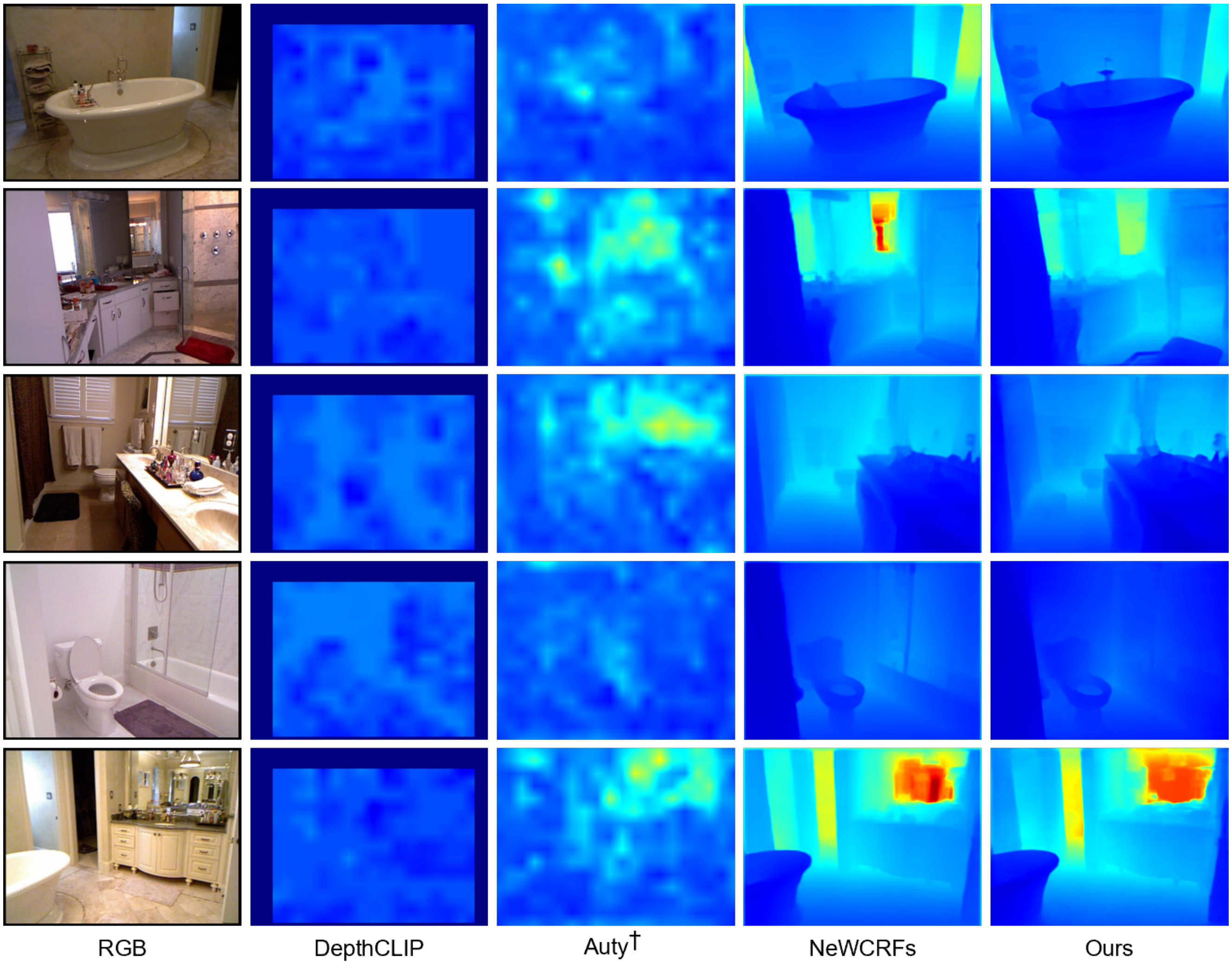

Figure 4. Qualitative comparison on the NYU-Depth V2 dataset. The symbol † means reimplementation.

rics. Specifically, it improved CLIP2Depth [23] by 22.97% in terms of the AbsRel metric. Additionally, despite the limitation that image features and text features from the VL models are not well aligned for abstract tasks such as depth estimation, our model showed comparable performance to existing unimodal vision models. Figure 5 shows the qualitative results of the estimated depth maps. Our model yielded high-quality boundary predictions and effectively captured small objects such as car side mirrors. Additionally, it adequately covered the entire depth range for the KITTI dataset.

4.6. Ablation Study

In this section, we conduct an ablation study to demonstrate the effectiveness of the proposed CaBins module. Experiments were performed using a ResNet-50 based image encoder and evaluated on the KITTI dataset. Moreover, we compare the results of different image encoders, including ResNet-50, ResNet-101, and ViT-B/16.

CaBins. The main contribution of our work is to use predicted depth values for text prompt instead of the semantic bins. In our CaBins module, we utilize a weighted summation with deviation to extract CaBins $\{e_k\}_{k=1}^{K}$ from all bin center values. To demonstrate this effect, we conducted experiments using semantic bins and various methods. For semantic bins, the word 'farthest' was added to the semantic bins used in [52] to match the number of text prompt. That is, the semantic bins used in the experiment are as follows: ['giant', ' extremely close', 'close', 'not in distance', 'a little remote', 'far', 'farthest', 'unseen'].

As shown in Table 3, our method reduced the RMSE error metric from 2.363 to 2.339 compared to the semantic bin-based method. These results support our proposal that utilizing predicted depth values for text prompt is more

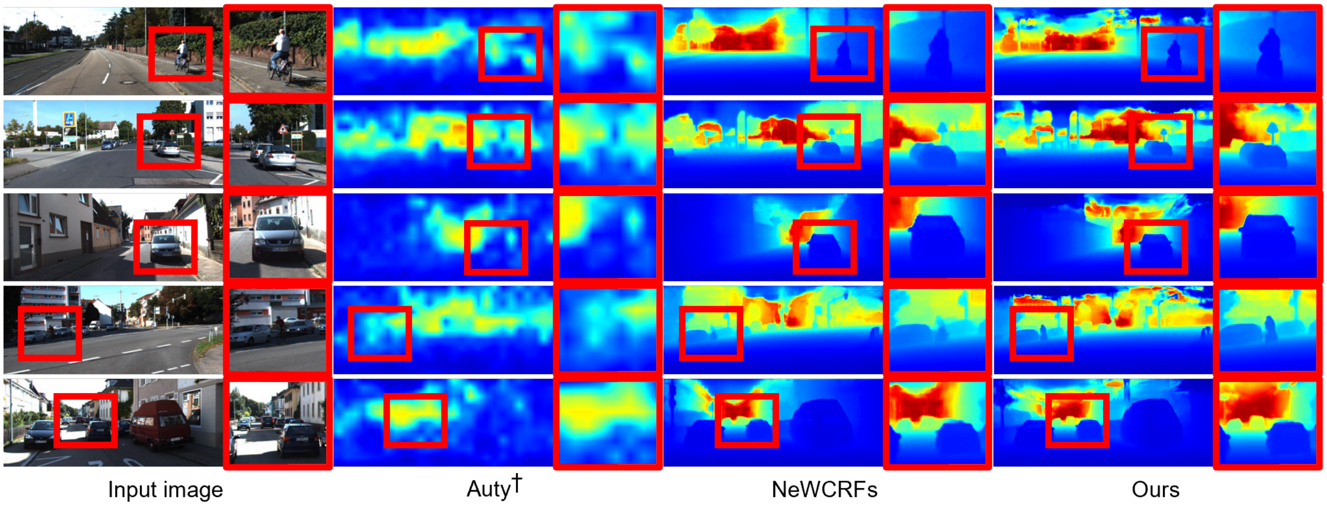

Figure 5. Qualitative comparison on the Eigen split of KITTI dataset. The symbol † means reimplementation.

practical than relying on semantic bins. Additionally, instead of merely selecting median values (denoted *"Median"*) or using simple average values (denoted *"Mean"*), utilizing values that account for the distribution of predicted bin centers showed improved performance.

Backbone. CLIP provides various image encoders based on ResNet and ViT. We compared the performance of the models on ResNet-50, ResNet-101 and ViT-B/16 backbones. For the ViT-B/16 backbone, we modified the architecture following DPT [38] to extract multi-scale features. Table 4 shows the results for different backbones. Regardless of the type of backbone, our model outperforms previous CLIP-based models by a large margin. However, we observed a significant performance gap between ResNet and ViT-based networks. We presume that this gap is caused by differences in the process of extracting multi-scale feature maps between two backbones. Specifically, the ResNet-based image encoder conducts stepwise downsampling to extract feature maps with strides of 4, 8, 16, and 32. On the other hand, the ViT-based image encoder generates multi-scale feature maps by applying upsampling or downsampling on the feature map with a stride of 16. We suppose that the lost information, which cannot be recovered during upsampling, may have impacted the depth estimation performance, where local information is important.

Distance class	AbsRel↓	SqRel↓	RMSE↓	log 10↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Semantic bin	0.059	0.196	2.363	0.026	0.959	0.995	0.999
Median	0.059	0.193	2.346	0.026	0.961	0.995	0.999
Mean	0.059	0.198	2.383	0.026	0.960	0.995	0.999
Ours	0.059	0.191	2.339	0.026	0.960	0.995	0.999

Table 3. Ablation experiments to demonstrate the proposed CaBins module. For this experiment, the ResNet-50 is adopted as an image encoder. Our method showed improved performance in terms of SqRel and RMSE metrics compared to other methods.

Backbone	AbsRel↓	SqRel↓	RMSE↓	log 10↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
RN50	0.059	0.191	2.339	0.026	0.960	0.995	0.999
RN101	0.057	0.186	2.322	0.025	0.964	0.995	0.999
ViT-16/B	0.070	0.246	2.542	0.030	0.952	0.994	0.999

Table 4. Ablation experiments on the different backbones.

5. Conclusion

In this paper, we propose a novel monocular depth estimation network, which leverages CLIP's pre-trained knowledge. Our network consists of CLIP encoders for image and text, a CaBins module, and a depth decoder. Unlike previous CLIP-based MDE models that rely on human-set semantic bins, we use the bin centers estimated from the image encoder as distance classes, called CaBins, for text prompts. CaBins are obtained through a weighted sum of the bin centers, with the weights defined based on the deviation from the mean of the bin centers. Experiments were conducted on the NYU-Depth V2 and KITTI datasets. The experimental results demonstrate the effectiveness of our proposed method. Especially, it outperforms previous CLIP-based depth models by a significant margin. We hope that our work will contribute to the advancement of the depth estimation techniques.

Acknowledgement

This research was supported by Samsung Electronics Co., Ltd.

References

[1] Ashutosh Agarwal and Chetan Arora. Attention attention everywhere: Monocular depth prediction with skip attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5861–5870, 2023. 2

[2] Dylan Auty and Krystian Mikolajczyk. Learning to prompt clip for monocular depth estimation: Exploring the limits of human language. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2039–2047, 2023. 2, 3, 6

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 1

[4] Alberto Baldrati, Lorenzo Agnolucci, Marco Bertini, and Alberto Del Bimbo. Zero-shot composed image retrieval with textual inversion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15338–15347, 2023. 1

[5] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4009–4018, 2021. 2, 4, 5

[6] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Localbins: Improving depth estimation by learning local distributions. In *European Conference on Computer Vision*, pages 480–496. Springer, 2022. 2

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1

[8] Wenjie Chang, Yueyi Zhang, and Zhiwei Xiong. Transformer-based monocular depth estimation with attention supervision. In *BMVC*, page 7, 2021. 6

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3

[12] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 1, 2, 5, 6

[13] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 5

[14] Junjie Fei, Teng Wang, Jinrui Zhang, Zhenyu He, Chengjie Wang, and Feng Zheng. Transferable decoding with visual entities for zero-shot image captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3136–3146, 2023. 1

[15] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018. 1, 2, 6

[16] Rao Fu, Xiao Zhan, Yiwen Chen, Daniel Ritchie, and Srinath Sridhar. Shapecrafter: A recursive text-conditioned 3d shape generation model. *Advances in Neural Information Processing Systems*, 35:8882–8895, 2022. 1

[17] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 740–756. Springer, 2016. 6

[18] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237, 2013. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[20] Xueting Hu, Ce Zhang, Yi Zhang, Bowen Hai, Ke Yu, and Zhihai He. Learning to adapt clip for few-shot monocular depth estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5594–5603, 2024. 2, 3, 5, 6

[21] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 1, 3

[22] Peng Jin, Hao Li, Zesen Cheng, Kehan Li, Xiangyang Ji, Chang Liu, Li Yuan, and Jie Chen. Diffusionret: Generative text-video retrieval with diffusion model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2470–2481, 2023. 1

[23] Dunam Kim and Seokju Lee. Clip can understand depth. *arXiv preprint arXiv:2402.03251*, 2024. 2, 3, 6, 7

[24] Hyeongjun Kwon, Taeyong Song, Somi Jeong, Jin Kim, Jinhyun Jang, and Kwanghoon Sohn. Probabilistic prompt learning for dense prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6768–6777, 2023. 3

[25] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 1, 2, 5, 6

[26] Pandeng Li, Chen-Wei Xie, Liming Zhao, Hongtao Xie, Jiannan Ge, Yun Zheng, Deli Zhao, and Yongdong Zhang. Progressive spatio-temporal prototype matching for text-video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4100–4110, 2023. 1

[27] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *arXiv preprint arXiv:2204.00987*, 2022. 2

[28] Zhenyu Li, Zehui Chen, Xianming Liu, and Junjun Jiang. Depthformer: Exploiting long-range correlation and local information for accurate monocular depth estimation. *Machine Intelligence Research*, 20(6):837–854, 2023. 6

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 1

[30] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1

[31] Ce Liu, Suryansh Kumar, Shuhang Gu, Radu Timofte, and Luc Van Gool. Single image depth prediction made better: A multivariate gaussian take. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17346–17356, 2023. 2

[32] Ce Liu, Suryansh Kumar, Shuhang Gu, Radu Timofte, and Luc Van Gool. Va-depthnet: A variational approach to single image depth prediction. *arXiv preprint arXiv:2302.06556*, 2023. 2

[33] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2015. 1

[34] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[35] Timo Lüddecke and Alexander S Ecker. Image segmentation using text and image prompts. in 2022 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7076–7086, 2021. 1

[36] Luigi Piccinelli, Christos Sakaridis, and Fisher Yu. idisc: Internal discretization for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21477–21487, 2023. 2

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 3

[38] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 2, 8

[39] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18082–18091, 2022. 2, 5

[40] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[41] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2008. 6

[42] Shuwei Shao, Zhongcai Pei, Xingming Wu, Zhong Liu, Weihai Chen, and Zhengguo Li. Iebins: Iterative elastic bins for monocular depth estimation. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[43] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12*, pages 746–760. Springer, 2012. 2

[44] 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. 3

[45] Hualiang Wang, Yi Li, Huifeng Yao, and Xiaomeng Li. Clipn for zero-shot ood detection: Teaching clip to say no. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1802–1812, 2023. 1

[46] Ziyang Wang, Yi-Lin Sung, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Unified coarse-to-fine alignment for video-text retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2816–2827, 2023. 1

[47] Tianyi Wei, Dongdong Chen, Wenbo Zhou, Jing Liao, Weiming Zhang, Gang Hua, and Nenghai Yu. Hairclipv2: Unifying hair editing via proxy feature blending. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23589–23599, 2023. 1

[48] Yuxiang Wei, Yabo Zhang, Zhilong Ji, Jinfeng Bai, Lei Zhang, and Wangmeng Zuo. Elite: Encoding visual concepts into textual embeddings for customized text-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15943–15953, 2023. 1

[49] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021. 1

[50] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3916–3925, 2022. 6

[51] Cheng Zhang, Xuanbai Chen, Siqi Chai, Chen Henry Wu, Dmitry Lagun, Thabo Beeler, and Fernando De la Torre. Iti-gen: Inclusive text-to-image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3969–3980, 2023. 1

[52] Renrui Zhang, Ziyao Zeng, Ziyu Guo, and Yafeng Li. Can language understand depth? In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6868–6874, 2022. 2, 3, 5, 6, 7

[53] Shuai Zhao, Linchao Zhu, Xiaohan Wang, and Yi Yang. Centerclip: Token clustering for efficient text-video retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 970–981, 2022. 1

[54] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9): 2337–2348, 2022. 3, 5