MonoFormer: Towards Generalization of self-supervised monocular depth estimation with Transformers

Jinwoo Bae, Sungho Moon, and Sunghoon Im
Department of Electrical Engineering and Computer Science, DGIST, Daegu, Korea
{sjg02122, byeol3325, sunghoonim}@dgist.ac.kr

Abstract

Self-supervised monocular depth estimation has been widely studied recently. Most of the work has focused on improving performance on benchmark datasets, such as KITTI, but has offered a few experiments on generalization performance. In this paper, we investigate the backbone networks (e.g., CNNs, Transformers, and CNN-Transformer hybrid models) toward the generalization of monocular depth estimation. We first evaluate state-of-the-art models on diverse public datasets, which have never been seen during the network training. Next, we investigate the effects of texture-biased and shape-biased representations using the various texture-shifted datasets that we generated. We observe that Transformers exhibit a strong shape bias and CNNs do a strong texture-bias. We also find that shape-biased models show better generalization performance for monocular depth estimation compared to texture-biased models. Based on these observations, we newly design a CNN-Transformer hybrid network with a multi-level adaptive feature fusion module, called MonoFormer. The design intuition behind MonoFormer is to increase shape bias by employing Transformers while compensating for the weak locality bias of Transformers by adaptively fusing multi-level representations. Extensive experiments show that the proposed method achieves state-of-the-art performance with various public datasets. Our method also shows the best generalization ability among the competitive methods.

1 Introduction

How do humans efficiently extract and recognize essential information from complex scenes? The biological vision system treats the object’s shape as the single most crucial vision cue, compared with other cues like texture or color [1]. This enables humans, even small children, to easily recognize an object from a line drawing or a silhouette image. It is widely known that convolutional neural networks (CNNs) are designed with inspiration from the biological neural networks in living organisms [2-4]. CNNs extract the simple patterns (e.g., edges) and then build complex patterns by successively composing early neural responses. However, in contrast to human visual representation, recent researches [5-7] have revealed that CNNs are strongly biased towards recognizing textures rather than shapes. CNN-based models rationally classify labels even in images with disrupted shape structures [8,9]. On the other hand, CNN models fail to predict labels correctly in a texture-removed image whose shape is well-preserved [10].

Then, how does this observation affect the monocular depth estimation task? Over the past decade, monocular depth estimation has made significant progress using CNNs [11-16]. These works show the remarkable performance on the KITTI datasets [17] even with the model trained in a self-supervised manner. However, the experiments have been conducted on only a few driving scenes, mostly KITTI datasets, so the generality of these methods has not been closely studied. In this paper,
we study the generalization performance of the state-of-the-art methods and investigate how texture-biased representation from CNNs affects monocular depth estimation. We evaluate state-of-the-art models trained on KITTI using six public depth datasets (SUN3D [18], RGBD [19], MVS [20], Scenes11 [20], ETH3D [21], and Oxford Robotcar [22]). We also conduct experiments on three different texture-shifted datasets including texture-smoothed (Watercolor), textureless (Pencil-sketch), and texture-transferred (Style-transfer) images. Through these extensive experiments, we determine that texture-biased models are vulnerable to generality in monocular depth estimation. Recently, Transformers [23] have received a surge of attention for their outstanding performance in the field of computer vision [23, 24], despite the lack of a spatial locality bias. Moreover, several works [25, 6, 26] show that Transformers have a strong shape bias, unlike CNNs. We also investigate the Transformers, similar to the experiments conducted for CNNs, and observe that shape bias is key to generalize depth estimation. Thus, we propose a CNN-Transformer hybrid network, called MonoFormer, which are highly complementary to each other. The design intuition behind MonoFormer is to take the strong shape bias of Transformers and the spatial locality bias of low-level Transformers features projected from CNN features. To do so, we design a layer-wise Attention Connection Module (ACM) and a Feature Fusion Decoder (FFD). The ACM measures the importance of shape bias representation and the local details, and then the FFD adaptively fuses them for depth prediction. The detailed ablation studies show that the shape-biased features are mostly extracted from high-level Transformers and the local details are captured at low layers.

To verify the generality, we evaluate our KITTI-trained model on the six out-of-distribution datasets. These experiments show MonoFormer achieves performance improvement of up to more than 30% over other CNN-based state-of-the-art models [14, 13, 15], 7% over a Transformer-based model [23], and 15% over a conventional hybrid model [27]. Our model shows strong robustness and generality regardless of the testing distributions. By investigating the network structures, we observe that the CNNs mostly learn texture-based representation while Transformers nearly learn shape-based representation. We also reveal that the shape-biased models achieve superior generalization ability compared with texture-biased models on out-of-distribution training datasets. Our contributions can be summarized as follows:

- We investigate the representation learned by CNNs, Transformers, and hybrid models for monocular depth estimation using various public datasets and stylized datasets.
- We propose a CNN-Transformer hybrid network with multi-level feature aggregation, which complements the shape bias and spatial locality bias toward the generalization of monocular depth estimation.
- Extensive experiments demonstrate the effectiveness of the proposed method, and our method achieves state-of-the-art performance on KITTI datasets, diverse out-of-distribution datasets, and texture-shifted datasets.

2 Related Work

2.1 Self-supervised monocular depth estimation

Self-supervised depth estimation methods [28, 14, 15, 29, 30, 11] simultaneously train depth and motion network by imposing photometric consistency loss between target and source images warped by the predicted depth and motion. SfMLearner [28] first proposes a depth and ego-motion estimation pipeline without the ground truth depth and motion. Monodepth2 [14] presents a minimum reprojection loss to handle occlusions, a full-resolution multi-scale sampling method to reduce visual artifacts, and an auto-masking loss to ignore outlier pixels. PackNet-SfM [15] introduces packing and unpacking blocks that leveraged 3D convolutions to learn the dense appearance and geometric information in real-time. HR-Depth [29] analyzes the reason for the inaccurate depth prediction in large gradient regions and designed a skip connection to extract representative features in high resolution.

2.2 Vision Transformers

Recently, Transformers [31] start to show promises for solving computer vision tasks such as image classification [23, 32], object detection [24], and dense prediction. [33, 34, 27]. ViT [23] employs Transformers architecture on fixed-size image patches for image classification for the first time.
Figure 1: Overall Architecture. We design an encoder-decoder structure with a multi-level feature fusion module. The encoder is composed of a CNN and Transformers. The ACM learns the channel and position attentions. The FFD adaptively fuses the encoder features using the attention maps.

DeiT [32] utilizes Knowledge distinction on ViT architecture, showing good performance only with the ImageNet dataset. DETR [24] propose the direct set prediction approach, which simplifies the object detection pipeline, based on a CNN-Transformer network and bipartite matching. Some works [34,27] have employed Transformers for monocular depth estimation in a supervised manner. DPT [34] introduces a dense prediction using a Transformer as the basic computational building block of the encoder. TransDepth [27] utilizes multi-scale information to capture local level details. These works show the generalized performance, but they require a large number of training datasets captured in diverse environments with ground truth depth maps.

2.3 Out-of-distribution generalization

Out-of-distribution (OOD) generalization is one of the core problems in machine learning where the testing distribution is unknown and different from the training [35,36]. Some works [37–39] learn how to disentangle the informative and distinct components of the data, which is considered a good representation for out-of-distribution generalization. β-VAE [40] can learn the representation more effectively by adding additional hyperparameter beta into the VAE objective function. Other works [41,42,5] generate a variety of training datasets by applying filtering methods and a style-transfer algorithm [43] to improve the generalization of their networks. In addition, some recent works [25,7,26] aim to improve generalization by analyzing several properties of Transformers or CNNs. These works focus on a simple computer vision task, classification, but the detailed analysis of dense prediction, especially monocular depth estimation, has not been explored.

3 Method

This section describes MonoFormer, our self-supervised monocular depth estimation network consisting of a CNN-Transformer encoder described in Sec. 3.1, an Attention Connection Module (ACM) in Sec. 3.2, and a Feature Fusion Decoder (FFD) in Sec. 3.3. The overall pipeline is illustrated in Fig. 1.

3.1 CNN-Transformer Encoder

The encoder consists of a CNN and Transformers. We use ResNet50 [44] as the CNN backbone, and L number of Transformers. In this work, we set the L as 4. An input image I passes through the CNN encoder to extract a feature map \( F \in \mathbb{R}^{C \times H \times W} \), then the map is divided into \( N \) (= \( \frac{H}{16} \times \frac{W}{16} \)) number of patches \( p_n \in \mathbb{R}^{C \times 16 \times 16} \), which is utilized as the input of the first Transformer layer. We additionally
use a special token $t_s$ following the work [34]. We input the patch tokens $p_n$, $n \in \{1, ..., N\}$ and the special token $t_s$ with a learnable linear projection layer $E$ as follows:

$$Z_0 = [t_s; p_1E; p_2E; \ldots; p_NE],$$

where $Z_0$ is the latent embedding vector. The Transformer encoder consists of a Multi-head Self-Attention (MSA) layer, a Multi-Layer Perceptron (MLP) layer, and Layer Norm (LN) layers. The MLP is built with GELU non-linearity [45]. The LN is applied before every block and residual connections apply after every block. Self-Attention (SA) at each layer $l \in \{1, ..., L\}$ is processed with the learnable parameters $W_Q^m, W_K^m, W_V^m \in \mathbb{R}^{C \times d}$ of {query, key, value} weight matrices, given the embedding vector $Z_l \in \mathbb{R}^{N \times C}$ as follows:

$$\text{MSA}_{l-1} = Z_{l-1} + \text{concat}(\text{SA}_{1-1}; \text{SA}_{2-1}; \ldots; \text{SA}_{M-1})W,$$

$$Z_l = \text{MLP}((\text{LN}(\text{MSA}_{l-1}))) + \text{MSA}_{l-1}.$$ 

This Transformer layer is repeated $L$ times with unique learnable parameters. The outputs of the Transformers $\{Z_1, ..., Z_L\}$ are utilized as the input of the following layers ACM and FFD.

### 3.2 Attention Connection Module (ACM)

We design a new skip connection method, ACM, which produces the attention of global context and a semantic presentation of the feature given the features $Z_l$, $l \in \{1, ..., L\}$. The skip connection is widely utilized for the dense prediction tasks [46] because it helps to keep the fine detail by directly transferring the spatial information to the decoder. However, it has been observed that in the naïve skip connection method, concatenating each feature is too simple to preserve local detail, such as object boundaries [47]. To tackle the problem, we introduce an ACM that produces attention weight by capturing long-range context and exploiting the inter-dependencies between each channel map, respectively. These two attention maps are utilized in the following section, which highlights the importance of the features.

The position attention module produces a position attention map $A^p_l \in \mathbb{R}^{C \times N}$ as follows:

$$A^p_l = \text{softmax}(Q^p_l (K^p_l)^T)V^p_l,$$

where $Q^p_l, K^p_l$ and $V^p_l$ are the query, key, and value matrices computed by passing $Z_l$ through a single convolutional layer. The channel attention module directly calculate the channel attention map $A^c_l \in \mathbb{R}^{C \times N}$ by computing the gram matrix of $Z_l$ as follows:

$$A^c_l = \text{softmax}(Z_lZ_l^T).$$

The position attention map $A^p_l$ and channel attention map $A^c_l$ enhance the feature representation by capturing long-range context and exploiting the inter-dependencies between each channel map, respectively. These two attention maps are utilized in the following section, which highlights the importance of the features.

### 3.3 Feature Fusion Decoder (FFD)

The FFD gets the encoder features $Z_l$, the attention maps $A^p_l, A^c_l$, and the output feature $X_l$ of the last Transformer layer passed through a Residual convolutional layer. The decoder fuses the feature $X_{L-l+1}$, $l \in \{1, ..., L\}$ through a single Convolutional layer (Conv) and Channel Normalization (CN) with learnable parameters $\alpha, \beta$ and $\gamma$ as follows:

$$X_{L-l} = \hat{X}_{L-l}[1 + \tanh(\gamma(CN(\alpha||\hat{X}_{L-l}||_2 + \beta))],$$

$$\hat{X}_{L-l} = \text{Conv}(w_pA^p_lZ_l + w_cA^c_lZ_l + Z_l) + X_{L-l+1},$$

where $w_p$ and $w_c$ are the learnable parameters that determine the importance of the position and channel attentions [49]. The parameter $\alpha$ works so that each channel can learn about each other individually, and $\gamma$ and $\beta$ control the activation channel-wisely following the work in [50]. Through this process, the FFD is able to produce a depth map from the fused features that preserve local detailed semantic representation while maintaining the global context of features.
Figure 2: Qualitative comparison to state-of-the-arts. We use KITTI for training and testing.

Table 1: Quantitative comparison to state-of-the-arts. We evaluate models trained on KITTI (K) and KITTI pre-trained on CityScapes [51] (K+CS). We only use monocular images (M) for supervision. Bold and Underline are the best and the second performance, respectively.

4 Experiments

4.1 Training loss and implementation detail

We train both depth and motion networks using photometric consistency (L2 loss and SSIM loss) and edge-aware smoothness losses following the best practices of self-supervised monocular depth estimation [28, 14, 15]. We set the weight for SSIM, L2 photometric, and smoothness losses as 0.85, 1.15 and 0.001, respectively. We use 7 convolution layers for 6DoF camera pose estimation following the work in [28]. We implement our framework on PyTorch and train it on 4 Titan RTX GPUs. We use the Adam optimizer [52] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Our model is trained for 30 epochs with a batch size of 8. The learning rates for depth and pose network are $2 \times 10^{-5}$ and $5 \times 10^{-4}$, respectively. We will release the source code, the trained weights and the datasets once the paper is accepted.

4.2 Comparison on KITTI datasets

We compare our method with state-of-the-art methods, SGCDepth [11], GeoNet [12], Struct2depth [16], Monodepth2 [14], PackNet-SfM [15], SGCDepth [30], R-MSFM [13] in Tab. 1. We use the KITTI Eigen split [53, 54] consisting of 39,810 training, and 4,424 validation and 697 test data. We additionally sample data about 5% of the total data with infinite-depth problems that mostly occur in dynamic scenes, following the work in [25]. We implement our framework on PyTorch and train it on 4 Titan RTX GPUs. We use the Adam optimizer [52] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Our model is trained for 30 epochs with a batch size of 8. The learning rates for depth and pose network are $2 \times 10^{-5}$ and $5 \times 10^{-4}$, respectively. We will release the source code, the trained weights and the datasets once the paper is accepted.

The quantitative results show that the proposed method outperforms other models. The qualitative results in Fig. 2 show that our method precisely preserves object boundaries. This demonstrates that the encoder captures both global context and informative local features and transfers them to the decoder for the pixel-wise prediction.
Figure 3: Depth map results on texture-shifted datasets. We test our hybrid/ViT models and the competitive models trained on KITTI using watercolor, pencil-sketch, and style-transfer images (Top to Bottom). Note that the Ours-Hybrid is equivalent to MonoFormer.

| Datasets    | Model       | Abs Rel ↓ | Sq Rel ↓ | RMSE ↓ | RMSElog ↓ | δ < 1.25 ↑ | δ < 1.25↑ | δ < 1.25↑ |
|-------------|-------------|-----------|----------|--------|-----------|------------|-----------|------------|
| Water-color | Monodepth2  | 0.170     | 1.345    | 6.175  | 0.263     | 0.730      | 0.909     | 0.960      |
|             | PackNet-SfM | 0.174     | 1.364    | 6.334  | 0.264     | 0.742      | 0.906     | 0.961      |
|             | R-MSFM6     | 0.194     | 1.613    | 7.173  | 0.302     | 0.696      | 0.876     | 0.943      |
|             | Ours-ViT    | 0.152     | 1.196    | 5.668  | 0.232     | 0.799      | 0.932     | 0.973      |
|             | Ours-Hybrid | **0.147** | **1.098** | **5.891** | **0.231** | **0.798** | **0.930** | **0.973** |
| Pencil-sketch| Monodepth2  | 0.196     | 1.522    | 6.232  | 0.276     | 0.691      | 0.898     | 0.962      |
|             | PackNet-SfM | 0.204     | 1.569    | 6.568  | 0.290     | 0.670      | 0.888     | 0.957      |
|             | R-MSFM6     | 0.217     | 1.698    | 6.719  | 0.301     | 0.647      | 0.872     | 0.951      |
|             | Ours-ViT    | 0.174     | 1.311    | 5.770  | 0.248     | 0.756      | 0.920     | 0.967      |
|             | Ours-Hybrid | **0.151** | **1.064** | **5.685** | **0.228** | **0.782** | **0.932** | **0.976** |
| Style-transfer| Monodepth2 | 0.435     | 6.107    | 10.891 | 0.309     | 0.579      | 0.800     | 0.821      |
|             | PackNet-SfM | 0.379     | 4.462    | 9.834  | 0.470     | 0.418      | 0.708     | 0.855      |
|             | R-MSFM6     | 0.394     | 4.667    | 10.214 | 0.490     | 0.399      | 0.680     | 0.837      |
|             | Ours-ViT    | 0.378     | 4.854    | 9.869  | 0.449     | **0.447**  | 0.739     | 0.869      |
|             | Ours-Hybrid | **0.355** | **3.891** | **9.489** | **0.441** | **0.441**  | **0.732** | **0.871** |

Table 2: Quantitative comparison on texture-shifted datasets.

4.3 Analysis of texture-/shape-bias on CNN and Transformer

Generally, the texture represents a spatial color or pattern of pixel intensity in an image [56]. To examine the influence of textures on the inference process in detail, we apply three different texture modification strategies including texture-smoothing (Watercolor), texture removal (Pencil-sketch), and texture-transfer (Style-transfer). The following is a summary of the image generation:

- **Watercolor**: We smooth the texture details from original images while preserving the color cues using `cv2.stylization`. The image looks like a watercolor picture.
- **Pencil-sketch**: We remove both textures and color from original images using `cv2.pencilSketch`. The image seems like a sketch drawn with pencils.
- **Style-transfer**: We apply a new texture to the original image (context) by utilizing other images (style) using a style transfer algorithm [43]. The textures of the original images are changed.

The generated images and the correspondence results are shown in Fig. 3. The first two images are watercolors, the middle two images and the last two images are pencil-sketch and style transferred images, respectively. We also conduct the quantitative evaluations in Tab. 2 using all of the KITTI test data (697 images). In this experiment, we compare the performance of CNN-based models (Monodepth2, PackNet-SfM, R-MSFM6), a Transformer-based model (Ours-ViT), and a hybrid (Ours-Hybrid) model. We note that Ours-Hybrid is equivalent to MonoFormer and Ours-ViT employs the ViT [23] encoder structure. Both qualitative and quantitative results of the watercolor data show that both the CNN-based and Transformer-based models produce plausible depth maps. However, the CNN-based model tends to lose more details of the object boundaries and has higher errors than the Transformer-based models. The experiments with the pencil-sketch data and the style-transfer data show that the Transformer-based models distinguish objects (e.g., pedestrians and cars) and stuff (e.g., walls and roads) better than the CNN-based models. Specifically, the CNN-based models produce unrecognizable depth maps on style-transfer data due to the loss of original texture information. These experiments demonstrate our two observations. One is that CNNs have a strong texture bias while Transformers have a strong shape bias. The other is that models with shape bias representation provide better generalization performance for monocular depth estimation compared to models.
with texture bias. Of particular note, MonoFormer (Ours-Hybrid) more precisely preserves object boundaries than Transformer-based model (Ours-ViT). Ours-ViT also generally produces reliable depth thanks to the shape bias of Transformers, but fails to recover details such as a pedestrians. We believe that the proposed multi-level feature fusion module captures both shape bias and the spatial locality bias.

4.4 Generalization performance of CNN-based, Transformer-based, and hybrid models

We compare the generalized performance of ours and all the competitive models trained on the KITTI datasets [53, 54]. We test the models using public depth datasets consisting of indoor scenes (SUN3D [18], RGBD [19]), synthetic scenes from graphics tools (Scenes11 [20]), outdoor building-focused scenes (MVS [20]), and night driving scenes (Oxford Robotcar [22]). We also use ETH3D [21] containing both indoor and outdoor scenes. The results in Fig. 4 show that the CNN-based models fail to estimate depth even though the scenes from the training and test sets share the stuff (e.g. road and sky) and things (e.g. cars), while the Transformer-based model keeps the details of object and scene. We observe that the texture shifts caused by illumination changes confuses the CNN-based model to estimate accurate depth. The test results on the other scene environments in Fig. 5 and Fig. 6 also show aspects similar to the results in Fig. 4. The transformers-based models recover scene depth even in the complex scenes containing the things and stuff which never been seen during training. However, the CNN-based models estimate unreliable depth maps, which keep the infinity depth mostly seen in KITTI datasets and loss the depth boundaries of objects. Of particular note,
Datasets Model Abs Rel Sq Rel RMSE δ < 1.25↑ δ < 1.25↓ δ < 1.25↑
MVS Monodepth2 0.471 0.407 0.503 0.408 0.661 0.623 0.806
PackNet-SfM 0.449 0.295 0.429 0.397 0.670 0.562 0.731
R-MSFM6 0.550 0.603 0.583 0.352 0.591 0.471 0.627
Ours-ViT 0.260 0.102 0.257 0.611 0.877 0.701 0.846
Ours-Hybrid 0.246 0.088 0.248 0.661 0.880 0.701 0.846

RGBD Monodepth2 0.610 0.508 0.488 0.292 0.520 0.471 0.627
PackNet-SfM 0.593 0.416 0.460 0.318 0.562 0.494 0.636
R-MSFM6 0.695 0.553 0.490 0.261 0.471 0.471 0.627
Ours-ViT 0.383 0.185 0.284 0.487 0.701 0.575 0.713
Ours-Hybrid 0.365 0.137 0.282 0.496 0.742 0.615 0.739

Scenes11 Monodepth2 1.647 0.763 0.356 0.312 0.529 0.471 0.627
PackNet-SfM 2.065 0.837 0.330 0.310 0.530 0.494 0.636
R-MSFM6 1.727 0.726 0.361 0.280 0.494 0.471 0.627
Ours-ViT 1.671 0.657 0.268 0.355 0.575 0.575 0.713
Ours-Hybrid 1.538 0.404 0.255 0.388 0.615 0.615 0.739

SUN3D Monodepth2 0.554 0.535 0.576 0.324 0.556 0.556 0.718
PackNet-SfM 0.523 0.406 0.506 0.310 0.530 0.530 0.721
R-MSFM6 0.599 0.163 0.554 0.814 0.814 0.910
Ours-Hybrid 0.247 0.088 0.257 0.579 0.869 0.910

ETH3D Monodepth2 1.007 0.780 0.396 0.318 0.536 0.536 0.687
PackNet-SfM 0.802 0.401 0.268 0.378 0.639 0.639 0.809
R-MSFM6 0.943 0.632 0.366 0.330 0.541 0.541 0.686
Ours-ViT 0.701 0.312 0.217 0.473 0.760 0.760 0.890
Ours-Hybrid 0.670 0.294 0.188 0.529 0.819 0.819 0.925

Table 3: Comparison results. Evaluation of KITTI-trained model on diverse public datasets.

MonoFormer (Ours-Hybrid) produces more accurate depth maps which preserve the fine structures compared with Ours-ViT. The quantitative evaluations in Tab. 3 show that ours outperforms all competitive methods for all datasets and all measurements. Our MonoFormer achieves performance improvement of up to more than 30% over other CNN-based state-of-the-art models and 7% over a Transformer-based model (Ours-ViT) on average in Abs Rel. We believe that our network efficiently combines the local region information from the proposed module while keeping the shape bias representation from Transformers.

4.5 Comparison to the conventional hybrid models

We compare our model with existing CNN-Transformer hybrid models, TransDepth [27] and DPT [34]. The original TransDepth model is trained with a large number of various datasets in a supervised manner. For a fair comparison, we train the author-provided TransDepth and DPT models with KITTI eigen split in a self-supervised manner. We conduct the quantitative comparison using the five out-of-distribution datasets as well as the KITTI datasets. The results in Tab. 4 show the performance improvement ratio from TransDepth to MonoFormer. The experiments show that the proposed method achieves performance improvement around 15% on average in Abs Rel over TransDepth. We also compare our model with another hybrid model, DPT [34] using KITTI datasets in Tab. 5. These results show that MonoFormer outperforms all the conventional hybrid models.

4.6 Ablation study

In this section, all experiments are conducted using KITTI datasets for training and testing same as the experiments performed in Sec. 4.2. Effectiveness of the proposed modules. We conduct an ablation study to demonstrate the effectiveness of the proposed modules, ACM and FFD in Tab. 5. The baseline is DPT [34]. The models with only the ACM module or FFD module marginally improve the depth estimation performance, due to
Table 5: Ablation study on ACM and FFD. Both is with ACM and FFD.

Table 6: Ablation study on the number of encoder and decoder layers.

Figure 7: Visualization of attention map and feature map. We visualize the self-attention map of the patch on the upper left corner of the image $I$. The left column from the second row is the attention map from shallow layers, whereas the right is the map from deep layers.

Figure 8: Visualization of results with/without ACM and FFD.

The number of encoder and decoder layers We compare the performance of our model according to the number of encoder and decoder layers in Tab. 6. We find out that the model with four transformer layers achieves the best performance. Thus, we set $L$ as four for our MonoFormer.

Visualization of attention maps. We visualize the attention maps from the lower to higher layers of Transformers. As shown in Fig. 7, the encoder in the shallow layer extracts local region features. The deeper the layer, the more global shape contexts are extracted. Another observation is that ACM captures more detailed attention at different depths of the encoder features. FFD enhances the encoder features by fusing them with the attention map from ACM. The fused feature captures features from coarse to fine details. These experiments show that our model is capable of accurate pixel-wise prediction as it secures adequate local details.

5 Conclusion

In this paper, we provide three important observations for the self-supervised monocular depth estimation task: 1) CNN-based models rely heavily on textures, while Transformer-based models rely on shapes for a monocular depth estimation task. 2) Texture-based representations lead to poor generalization performance with texture-shift such as scene changes, illumination changes, and style changes. 3) Shape-based representations are more helpful for a generalized monocular depth model than texture-based representations. Based on these observations, we propose a CNN-Transformer hybrid network, called MonoFormer, which incorporates both shape bias and spatial locality bias. The proposed model achieves the best performance among various competitive methods on diverse unseen datasets as well as KITTI datasets, by a high margin. The extensive experiments demonstrate that our MonoFormer has superior generalization ability. We believe that the performance improvement comes from the design of strong shape-biased models, and this observation can be a useful insight to better understanding of monocular depth estimation. There are some limitations and we plan to explore further in future works. In this paper, we only consider the problem for self-supervised monocular depth estimation. This can be further studied for supervised methods with a large amount of diverse datasets, stereo depth estimation, or multi-view stereo matching. The other important and interesting topic is to reduce model size and memory consumption of the hybrid models.
References

[1] Barbara Landau, Linda B Smith, and Susan S Jones. The importance of shape in early lexical learning. *Cognitive development*, 1988.

[2] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat’s striate cortex. *The Journal of physiology*, 1959.

[3] Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1988.

[4] Nikolaus Kriegeskorte. Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annual review of vision science*, 2015.

[5] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019.

[6] Katelyn Morrison, Benjamin Gilby, Colton Lipchak, Adam Mattioli, and Adriana Kovashka. Exploring corruption robustness: Inductive biases in vision transformers and mlp-mixers. *arXiv preprint arXiv:2106.13122*, 2021.

[7] Shikhar Tuli, Ishita Dasgupta, Erin Grant, and Thomas L Griffiths. Are convolutional neural networks or transformers more like human vision? In *CogSci*, 2021.

[8] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Texture and art with deep neural networks. *Current opinion in neurobiology*, 2017.

[9] Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. In *ICLR*, 2019.

[10] Pedro Ballester and Ricardo Matsumura Araujo. On the performance of googlenet and alexnet applied to sketches. In *AAAI*, 2016.

[11] Mingkang Xiong, Zhenghong Zhang, Weilin Zhong, Jinsheng Ji, Jiyuan Liu, and Huilin Xiong. Self-supervised monocular depth and visual odometry learning with scale-consistent geometric constraints. In *IJCAI*, 2021.

[12] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018.

[13] Zhongkai Zhou, Xinnan Fan, Pengfei Shi, and Yuanxue Xin. R-msfm: Recurrent multi-scale feature modulation for monocular depth estimating. In *CVPR*, 2021.

[14] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, 2019.

[15] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *CVPR*, 2020.

[16] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI*, 2019.

[17] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013.

[18] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *ICCV*, 2013.

[19] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IROS*, 2012.

[20] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *CVPR*, 2017.
[21] Thomas Schops, Johannes L Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In CVPR, 2017.

[22] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. IJRR, 2016.

[23] 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. In ICLR, 2020.

[24] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In ECCV, 2020.

[25] Chongzhi Zhang, Mingyuan Zhang, Shanghang Zhang, Daisheng Jin, Qiang Zhou, Zhongang Cai, Haiyu Zhao, Shuai Yi, Xianglong Liu, and Ziwei Liu. Delving deep into the generalization of vision transformers under distribution shifts. In CVPR, 2022.

[26] Namuk Park and Songkuk Kim. How do vision transformers work? In ICLR, 2022.

[27] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformer-based attention networks for continuous pixel-wise prediction. In CVPR, 2021.

[28] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In CVPR, 2017.

[29] Xiaoyang Lyu, Liang Liu, Mengmeng Wang, Xin Kong, Lina Liu, Yong Liu, Xinxin Chen, and Yi Yuan. Hr-depth: high resolution self-supervised monocular depth estimation. In AAAI, 2021.

[30] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In ECCV, 2020.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NIPS, 2017.

[32] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In ICML, 2021.

[33] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zeikun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In CVPR, 2021.

[34] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In CVPR, 2021.

[35] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Wenjun Zeng, and Tao Qin. Generalizing to unseen domains: A survey on domain generalization. In IJCAI, 2021.

[36] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. arXiv preprint arXiv:2108.13624, 2021.

[37] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. Proceedings of the IEEE, 2021.

[38] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In ICML, 2019.

[39] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In ICLR, 2017.
[40] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *PAMI*, 2013.

[41] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

[42] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021.

[43] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016.

[44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[45] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[46] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015.

[47] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. U-Net++: A nested u-net architecture for medical image segmentation. In *DLMIA*. 2018.

[48] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, 2019.

[49] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019.

[50] Zongxin Yang, Linchao Zhu, Yu Wu, and Yi Yang. Gated channel transformation for visual recognition. In *CVPR*, 2020.

[51] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[52] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[53] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013.

[54] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[55] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. In *ICLR*, 2020.

[56] Laleh Armi and Shervan Fekri-Ershad. Texture image analysis and texture classification methods-a review. *arXiv preprint arXiv:1904.06554*, 2019.