Towards Low-Latency Monocular Depth Estimation

Mihir Deshmukh [1] **Sujit Shankar Mohite** [1] **Reese Haly** [1] **Jackson Martinez Balcazar** [1]

Abstract

The goal of this project was to reduce inference latency for Monocular Depth Estimation (MDE) while maintaining and/or improving accuracy. We enhanced the PixelFormer architecture for monocular depth estimation by integrating efficient attention mechanisms, leading to a reduction in inference latency. Comparative tests confirmed that our modifications outperformed the base PixelFormer in terms of loss and computational efficiency.

1. Introduction

Monocular Depth Estimation (MDE), the ability to perceive depth from a single image, is a skill that poses various challenges. In the realm of autonomous driving, accurate and efficient MDE is essential for safely navigating streets. In robotics, it enables machines to better understand and interact with their surroundings. Similarly, augmented and virtual reality rely heavily on quality performing MDE's to seamlessly blend virtual components with the real world. Traditional approaches to depth estimation, like stereo vision, often use dual cameras, whereas single camera-based techniques use specialized structured light sources. Each of these approaches possesses distinct merits: stereo vision demonstrates adaptability, while structured light excels in diverse environmental conditions.

State-of-the-art (SOTA) techniques for MDE are based on encoder-decoder style transformer architectures, like PixelFormer (Agarwal & Arora, 2023), and Convolutions Neural Network (CNN) architectures. While various state-of-the-art architectures demonstrate impressive performance on depth estimation tasks, they aren't without limitations. These mechanisms often have high computational complexities with respect to their sequence length. Moreover, they often struggle to capture dependencies over long distances within the sequence, often resulting in a loss of context and relevant information (Agarwal & Arora, 2023).

Our project aims to improve computational efficiency by introducing Efficient and Performer attention mechanisms into the decoder of the PixelFormer model. These mechanisms are designed to provide a global context with linear computational complexity, potentially improving the efficiency as well as the performance of MDE models.

1.1. Research contributions

This project showcases the effectiveness of Efficient and Fast attention in transformer models for Monocular Depth Estimation (MDE). The integration of these attention variants not only enhances latency but also improves accuracy compared to the base PixelFormer model. Importantly, these attention variants are adaptable across domains, allowing for a broader exploration of their impact on computational efficiency and model accuracy, contributing significantly to the field of transformer-driven architectures.

2. Related Work

In 2021, Dosovitskiy et al. introduced transformers into the field of Computer Vision with their Vision Transformer (ViT) model(Dosovitskiy et al., 2021). ViT splits and inputs into a series of patches then embeds and performs self-attention on the embedded tokens. By taking transformers, which were previously reserved for natural language processing tasks, and replacing convolutions with self-attention, the authors were able to achieve competitive performance when compared to state-of-the-art CNNs while also opening the door to experimentation on this new type of model.

Building on top of this model, Liu et al. presented Swin Transformers to "serve as a general-purpose backbone for computer vision"(Liu et al., 2021). They implemented a hierarchical transformer attention with the use of shifted windows. During self-attention, the input tokens are split into smaller local windows of patches in an attempt to model local spatial contexts while limiting the computational complexity of self-attention to scale with the size of the local window instead of the size of the input image. The use of shifted windows in self-attention not only al-

[1]Worcester Polytechnic Institute. Correspondence to: Mihir Deshmukh <c.mpdeshmukh@wpi.edu>, Sujit Shankar Mohite <c.smohite@wpi.edu>, Reese Haly <c.rjhaly@wpi.edu>, Jackson Martinez Balcazar <c.jmartinezbalcaza@wpi.edu>.

CS/DS 541 Deep Learning Final Project, Worcester Polytechnic Institute.

lows the Swin Transformer to model fine-grained details but also promotes its scalability, making it a solid backbone for computer vision tasks.

Agarwal et al. adapted the Swin Transformer architecture as an encoder for use in MDE in their model, PixelFormer (Agarwal & Arora, 2023). PixelFormer poses MDE as a pixel query refinement problem, in which coarser, more general features are used to initialize the pixel-level queries, and then these queries are refined to a higher resolution by utilizing Skip Attention Modules (SAM). SAMs use self-similarity between the pixel queries and the ensuing encoder embeddings to reintroduce fine-grained details back into the input. Through this method, PixelFormer is able to fuse global and local contexts throughout inference, allowing it to successfully model accurate depth information from a singular input stream.

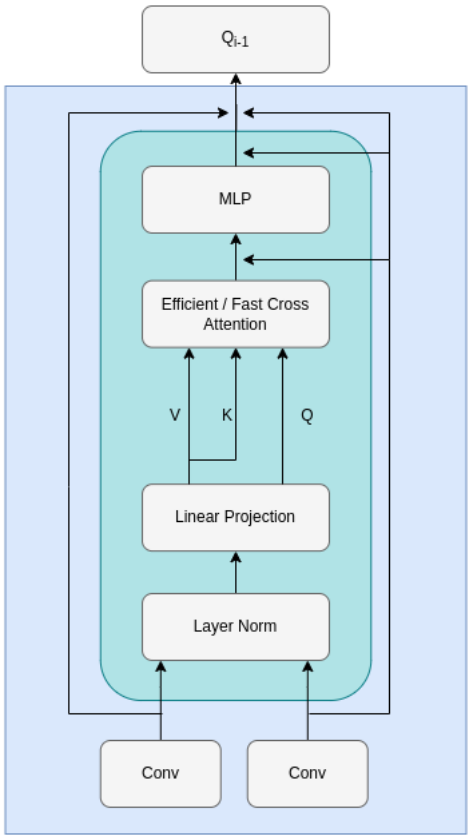

Figure 1. Architecture of the Skip Attention Module(SAM)

3. Proposed Method

In an effort to reduce the latency of MDE while maintaining accuracy, we decided to replace the attention mechanism in the PixelFormer architecture's SAMs with more efficient attention mechanisms. PixelFormer uses window attention to calculate the attention matrix(Agarwal & Arora, 2023). Window attention works under the assumption that tokens in the embedding will most likely be more related to their neighbors than tokens further away. To accomplish this, window attention uses a sliding window to limit calculating attention only to a token's closest neighbors. While this both decreases the time and space complexity of calculating attention, it is still less time and space efficient than other attention mechanisms and decreases the amount of information available during attention, thus reducing accuracy. These problems are addressed by the following two attention functions: efficient attention and fast attention via Fast Attention Via Positive Orthogonal Random Features(FAVOR+). Replacing the default attention mechanism in the PixelFormer architecture with these attention mechanisms should allow the model to compute inference quicker while also providing more information during attention, thus improving accuracy

$$\boldsymbol{E}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \rho_q(\boldsymbol{Q})\left(\rho_k(\boldsymbol{K})^{\mathsf{T}}\boldsymbol{V}\right)$$

Figure 2. Equation used to calculate efficient attention

Efficient attention and FAVOR+ work in a similar way to reduce the computational and spatial complexity of calculating attention. Efficient attention works by calculating softmax on queries and keys individually before multiplying the matrices together(Shen et al., 2020). On the other hand, FAVOR+ uses individual row-vectors from the queries and keys and the equations below to approximate softmax(Choromanski et al., 2022). Both of these attention mechanisms relieve the key issue with standard attention, which is that the attention matrix grows quadratically with respect to the input. Using efficient attention and FAVOR+ removes this necessity, allowing the estimated attention to grow linearly. As a result, the time and space complexity of our PixelFormer model's attention mechanism will also grow linearly, which will decrease inference latency while additionally utilizing all of the input to calculate attention. By implementing these attention mechanisms into a PixelFormer model and retraining the model, we hope to achieve faster inference while at least maintaining accuracy.

$$\mathrm{SM}(\mathbf{x},\mathbf{y}) = \mathbb{E}_{\omega\sim\mathcal{N}(0,\mathbf{I}_d)}\left[\exp\left(\omega^\top\mathbf{x}-\frac{\|\mathbf{x}\|^2}{2}\right)\exp\left(\omega^\top\mathbf{y}-\frac{\|\mathbf{y}\|^2}{2}\right)\right]$$

Figure 3. Equation used to calculate FAVOR+

We also propose to use a different alignment function, specifically cosine similarity in window attention, to identify the impact of a different alignment function that is less sensitive to magnitude than dot product attention.

Modified SAM: As seen in figure 1, we replaced the window cross attention in SAM propose by (Agarwal & Arora, 2023) with Fast and Efficient Attention. We also changed the key dimension to match the input encoder feature dimensions. The key and values are supplied by the encoder, while the queries are provided by the decoder, which is kept from the base PixelFormer. The subsequent SAM modules take the corresponding encoder features and the previous SAM output to generate a feature map.

Another variant of the PixelFormer is the addition of cosine similarity in the existing window attention block in SAM. We propose this modification to identify a well-suited alignment function.

4. Experiments

NYU Depth V2: We utilized the NYU Depth V2 dataset for our experiments. This indoor dataset comprises 120,000 RGB images and corresponding depth pairs, each with dimensions of 480x640. The data is captured through video sequences in 464 distinct indoor scenes, utilizing the Microsoft Kinect sensor. We adhered to the official training/testing split established by the 'Attention Attention Everywhere' paper, which guides our baseline comparison. The training set comprises 36,253 images from 249 scenes, while the test dataset encompasses 654 images derived from 215 scenes. The input image size was 480x640. The proposed network outputs depth predictions with a resolution of 120x160, which we upsampled by 4 to match the ground truth for training and testing data.

Implementation Details: We implemented three new variants for the base PixelFormer model. The first variant added Fast Attention, the second was Efficient Attention, and the other was Cosine similarity window attention. The proposed variants are implemented in PyTorch. We employ the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 1×10^{-2}. To expedite training, we initialize the encoder weights with pre-trained weights from the Swin Transformer-L and keep them frozen throughout the training process, reducing overall training time. After training each variant of the PixelFormer model, we observed that the Root Mean Squared Error(RMSE) loss saturates within 15 epochs for each variant. The learning rate is initially set to 4×10^{-5}, and we linearly decrease it to 4×10^{-6} across training iterations. To enhance the model's robustness, we incorporate various data augmentation techniques, including random rotation, horizontal flipping, and adjustments to image brightness.

The validation of each PixelFormer variant implementation is done on Google Colab Pro. The training is conducted on 2 NVIDIA A100 (40GB) GPUs in data parallel mode on the WPI Turing cluster. Base PixelFormer model and proposed variants requires about 19.80 hours to train on 2 NVIDIA A100 GPUs for 20 epochs. After training the base model and its variants with a batch size of 4, 8, and 16, we found 8 to be the optimum batch size for all variants, which yields the best results.

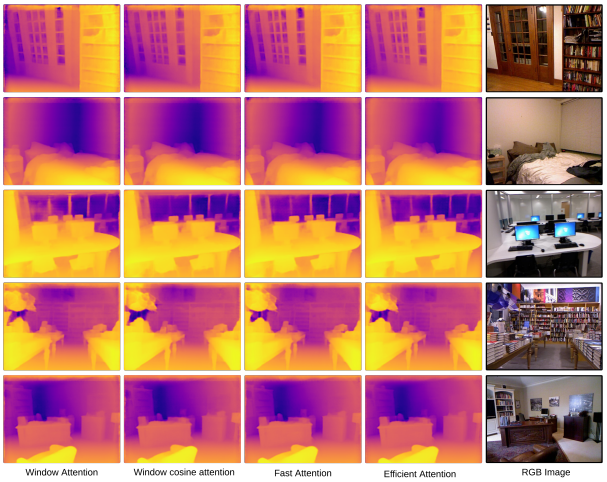

Figure 4. Comparison of model outputs

5. Results

We performed metric-based quantitative and image-based qualitative comparisons for the base PixelFormer model with our proposed three variants.

5.1. Effectiveness of Efficient and Fast attention:

Table 1 shows the quantitative comparison of RMSE loss for the base PixelFormer model and proposed three variants. All proposed variants yielded lower RMSE errors than the base PixelFormer model. The efficient attention model performed the best out of all variants, outperforming the base PixelFormer model by 1.24% in RMSE loss.

Model Variant	PixelFormer Base	Cosine Similarity	Efficient Attention	Fast Attention
RMSE	0.322	0.319	**0.318**	0.320

Table 1. Quantitative comparison of RMSE on NYUV2 dataset

We validated the improvement in RMSE loss by performing inference on images from the test set selected from various scenes. As seen in Figure 4 we can see that the window attention has an aliasing effect around the edges. This is seen clearly in the bookshelf image, where we see a severe fringing effect on the image borders in the base model. This is mitigated by global attention mechanisms like FAVOR+ and Efficient attention, as observed in the output images.

Model Variant	PixelFormer Base	Cosine Similarity	Efficient Attention	Fast Attention
FLOP's	10.079	10.082	**9.787**	16.908

Table 2. Quantitative comparison of Computational Efficiency

Model Variant	PixelFormer Base	Cosine Similarity	Efficient Attention	Fast Attention
Inf. Time	196.47	197.57	**189.83**	209.52

Table 3. Quantitative comparison of Inference Time

These results help validate our approach to add global context, which we can clearly say helps in improving the model output.

Further, we also observe cosine similarity with window attention performs much better than the baseline dot product attention consistently across all the outputs. This implies the appropriateness of the choice of cosine similarity over the dot product for the task of combining features for monocular depth estimation.

5.2. Impact on Computational Efficiency

We assessed the influence of our proposed variants on inference time and computational efficiency, which is crucial for achieving low-latency monocular depth estimation. To quantify the computational load, we computed the Floating Point Operations (FLOPs) of the SAM module for each proposed variant and the base model. We determined that the FLOPs for the encoder, which were consistent across all four variants, amounted to 0.227 TFLOPs. Table 2 shows the impact on the computational efficiency of our proposed variants of the SAM block against the base PixelFormer SAM block. We found that the proposed SAM block with Efficient Attention has the highest computational efficiency among all other proposed variants and the base PixelFormer SAM block. FAVOR+ only has 1.5 times FLOPS but includes global context instead of the small 7*7 window; hence, it is still an impressive result.

Subsequently, we calculated the inference time for each model to generate a depth map for an image to determine the latency of the proposed variants. Table 3 shows the findings that were consistent with our computation efficiency calculations, with the Efficient Attention model demonstrating the shortest inference time. Notably, the Efficient model exhibited a 3.38% improvement in speed compared to the base PixelFormer model. Moreover, we conducted an evaluation of the influence of Automatic Mixed Precision mode on the inference time, focusing on the fastest model. Remarkably, the inference time for the Efficient Attention model was reduced to 85.58 seconds with an RMSE of 0.323.

6. Discussion

With the introduction of the proposed three attention mechanisms, we were able to improve the performance while reducing the number of FLOPS. This validates the effectiveness of our proposed approach. Introducing (Shen et al., 2020) and (Choromanski et al., 2022) helps add global context for fusing the encoder and decoder features. While effective, this approach has a small impact on the inference time, as about 80 percent of the FLOPS are required for the encoder output. To have a significant impact, we need to change the attention mechanism inside the multi-stage SWIN encoder used here. Another approach could be to make use of a more efficient multi-stage transformer instead, like (Wu et al., 2021).

Further, as seen in Figure 4 we can clearly infer that cosine similarity is a better alignment function compared to dot product attention for fusing encoder and decoder features. Even the windowed cosine similarity is able to perform similarly to global attention(Efficient & Fast). We believe this can be attributed to the cosine similarity being sensitive to only the directions of the vectors and not the magnitude. The magnitude doesn't carry information about the similarity of the features, and hence, having an alignment function that is less sensitive to the magnitude performs better. One potential approach would be to calculate cosine similarity in the global attention approaches with the help of normalizing the input features before feeding them to the attention mechanism. This might help mitigate the fringing effect around the edges as well as the grid artifacts that we observe in windowed cosine attention in the background windows in the computer image in Fig 4.

7. Conclusions and Future Work

In this paper, we investigated the effects of several attention functions on the accuracy and latency of a popular monocular depth estimation model, PixelFormer. Our results suggest that using Efficient attention and FAVOR+ in the PixelFormer cross-attention function leads to an increase in accuracy due to more information during attention. Efficient attention also has a decrease in floating-point operations(FLOPs) even while having a global context. We also conclude that cosine similarity is a better-suited alignment function for combining encoder and decoder features in cross-attention. We noticed that most FLOPS were computed during the model's encoder. Therefore, to decrease latency further, we suggest future work to investigate replacing the attention mechanism in PixelFormer's encoder.

References

Agarwal, Ashutosh and Arora, Chetan. Attention attention everywhere: Monocular depth prediction with skip attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 5861–5870, January 2023.

Choromanski, Krzysztof, Likhosherstov, Valerii, Dohan, David, Song, Xingyou, Gane, Andreea, Sarlos, Tamas, Hawkins, Peter, Davis, Jared, Mohiuddin, Afroz, Kaiser, Lukasz, Belanger, David, Colwell, Lucy, and Weller, Adrian. Rethinking attention with performers, 2022.

Dosovitskiy, Alexey, Beyer, Lucas, Kolesnikov, Alexander, Weissenborn, Dirk, Zhai, Xiaohua, Unterthiner, Thomas, Dehghani, Mostafa, Minderer, Matthias, Heigold, Georg, Gelly, Sylvain, Uszkoreit, Jakob, and Houlsby, Neil. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

Liu, Ze, Lin, Yutong, Cao, Yue, Hu, Han, Wei, Yixuan, Zhang, Zheng, Lin, Stephen, and Guo, Baining. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

Shen, Zhuoran, Zhang, Mingyuan, Zhao, Haiyu, Yi, Shuai, and Li, Hongsheng. Efficient attention: Attention with linear complexities, 2020.

Wu, Haiping, Xiao, Bin, Codella, Noel C. F., Liu, Mengchen, Dai, Xiyang, Yuan, Lu, and Zhang, Lei. Cvt: Introducing convolutions to vision transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22–31, 2021. URL https://api.semanticscholar.org/CorpusID:232417787.