Lightweight Transformer Architectures for Monocular Depth Estimation

NYU MSCS Deep Learning – Final Project

Weiming Lei[1]

[1]Department of Electronic and Computer Engineering, New York University
wl3328@nyu.edu

Abstract

This project focus on the challenge of monocular depth estimation, targeting the computational demands associated with state-of-the-art methods that utilize Vision Transformers (ViTs). The primary goal was to thoroughly evaluate the performance-efficiency trade-off for lightweight models. Our approach involved implementing and comparing a lightweight hierarchical Transformer, Swin-Tiny, against a Convolutional Neural Network (CNN) baseline built on a ResNet-34 encoder. Both models shared a common depth decoder and were trained and evaluated on the $NYUDepth-V2-dataset$. The key findings revealed that the ResNet-34 CNN baseline achieved superior depth prediction accuracy, registering a final AbsRel of 0.1259 and a $\delta 1$ accuracy of 0.8642. In contrast, the Swin-Tiny model, despite its architectural novelty, yielded an AbsRel of 0.1523 and a $\delta 1$ accuracy of 0.8122. We conclude that while lightweight Transformers are promising, optimized CNN-based solutions remain a strong, high-performance option for resource-constrained monocular depth estimation without extensive pre-training.

Keywords: deep learning, Monocular depth estimation, lightweight Transformer

1 Introduction

The field of computer vision relies heavily on monocular depth estimation, the task of inferring a dense, per-pixel depth map from a single 2D RGB image. The domain has been revolutionized by Vision Transformers (ViTs), with models like DPT achieving state-of-the-art performance.

This project focuses on evaluating lightweight models to find an optimal balance between prediction performance and efficiency. Our contributions are summarized as follows:

- We investigate the performance-efficiency trade-off of lightweight Transformer architectures for monocular depth estimation.
- We implement and compare the lightweight Transformer, Swin-Tiny, against a traditional CNN baseline utilizing a ResNet-34 encoder.
- We provide a analysis to evluate if small, optimized Transformers can achieve competitive depth estimation performance without requiring massive, large-scale pre-training.

2 Background and Related Work

2.1 Monocular Depth Estimation and Deep Learning

Monocular depth estimation (MDE), the process of inferring a dense, per-pixel depth map from a single 2D image, is a foundational task in computer vision with direct applications in robotics, autonomous systems, and augmented reality (AR/VR). Our work uses the NYU Depth V2 dataset [8], a standard benchmark for indoor MDE. Performance is quantified using metrics such as Absolute Relative Error (AbsRel) and δ thresholds.

2.2 CNNs as Baselines and Innovations in Transformers

Convolutional Neural Networks (CNNs), particularly the ResNet architecture [3] with its crucial residual connections, have long served as powerful and efficient MDE encoders. The work of Saxena et al. [7] established classical CNN and UNet-based models as strong lightweight references, a paradigm our ResNet-34 baseline follows.

More recently, the advent of Vision Transformers (ViT) [2] shifted the state-of-the-art. Models like the Dense Prediction Transformer (DPT) [6] achieve superior accuracy by leveraging the ViT's global self-attention mechanism, often requiring extensive pre-training and large model sizes (e.g., ViT-L/16). Intermediate methods, such as AdaBins [1], introduced a key innovation by using Transformers to predict adaptive depth bin centers, significantly improving accuracy while still maintaining a moderately complex structure.

2.3 The Lightweight Efficiency Challenge

The high computational cost associated with the large ViT backbones used in state-of-the-art models like DPT presents a major barrier to real-world deployment on resource-constrained platforms.

This project focuses on evaluating the challenge of efficiency by comparing an optimized CNN approach with a computationally reduced Transformer:

- Optimized CNN Baseline: We employ a ResNet-34 encoder paired with a UNet-style decoder, representing a mature and effective lightweight benchmark [7].
- Swin-Tiny: We utilize the Swin Transformer [4], which achieves linear computational complexity by replacing global attention with shifted window attention and incorporating a hierarchical architecture. The Swin-Tiny variant is specifically chosen to test the limits of lightweight Transformers when trained without massive, external pre-training.

Our contribution is a focused, empirical analysis on the performance-efficiency trade-off between these two resource-conscious architectures for MDE, providing insight into a better choices for constrained environments.

3 Problem Statement and Goals

3.1 Problem Description

The core technical task of this project is monocular depth estimation (MDE).

Task Definition

Given a single 2D color image (RGB), the task is to infer and predict a dense, corresponding depth map where each pixel value represents the distance of the scene point from the camera.

Input: A single RGB image, processed and resized to (224×224) pixels, standardized using ImageNet mean and variance.

Output: A dense depth map of the same spatial resolution, with depth values clipped to a valid range (e.g., 0.001-10 meters).

The main question we seek to answer is: Can a lightweight, computationally efficient Transformer architecture (Swin-Tiny) achieve competitive monocular depth estimation performance compared to a optimized CNN baseline (ResNet-34), when both are trained under comparable, resource-constrained conditions.

3.2 Objectives and Scope

The project's objectives are defined by a focused, empirical comparison, and its scope is strictly limited to indoor scenes and supervised learning.

Objectives

Implement and train two distinct lightweight encoder architectures: the ResNet-34 encoder (as the CNN baseline) and the Swin-Tiny Transformer (as the lightweight Transformer).

Evaluate both models under controlled conditions by employing a shared depth decoder (e.g., DPT-like multi-scale fusion) and training them on the same split of the NYU Depth V2 dataset. Quantify and compare the performance of both models using standard MDE metrics (AbsRel, RMSE, $\delta 1$,$\delta 2$,$\delta 3$ thresholds) to assess accuracy. Analyze the comparative performance (accuracy) versus the potential computational efficiency (model size/theoretical speed) trade-off.

Scope and Limitations

The project is limited to the NYU Depth V2 dataset [5], meaning all scenes are indoor environments (kitchen, living room, office). The findings are not expected to generalize directly to complex outdoor environments or disparate datasets. The study is restricted to the supervised learning paradigm using the labeled subset of the NYU Depth V2 dataset. Self-supervised or unsupervised methods are outside the current scope. The focus is exclusively on lightweight models (ResNet-34 and Swin-Tiny). We do not aim to implement or against the full, computationally heavy state-of-the-art models (like the full DPT or ViT-L/16).

4 Approach

4.1 Overall Design

The method utilizes a standard encoder-decoder structure, where the key is the swapping of the encoder backbone while maintaining a shared decoder to ensure a fair comparison of the extracted feature quality.

The main components and their interaction are as follows:

$Input$: RGB Image (resized to 224×224).

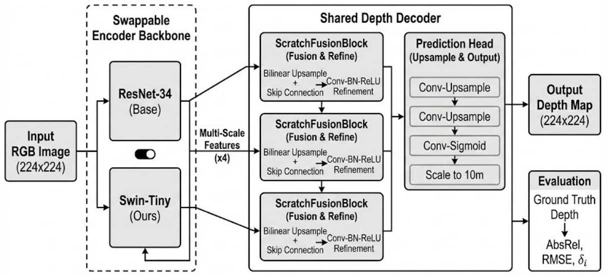

Figure 1: Overview of the proposed lightweight monocular depth estimation approach

$EncoderBackbone$: Either ResNet-34 (CNN Baseline) or Swin-Tiny (Lightweight Transformer) extracts multi-scale features.

$DepthDecoder$: A unified, DPT-like multi-scale fusion decoder receives the features and aggregates them to generate a high-resolution depth map.

$OutputandEvaluation$: The final dense depth map is generated and evaluated against the ground truth on the NYU Depth V2 test set.

4.2 Models, Methods, or System Components

Encoder Architectures

CNN Baseline (ResNet-34): The standard 34-layer Residual Network (ResNet-34) [3] is used, instantiated via the *timm* library. It provides four hierarchical feature maps from its stages, which serve as the multi-scale inputs to the decoder. ResNet-34 is selected as a well-established, computationally efficient benchmark for comparison against the novel Transformer architecture [7].

Lightweight Transformer (Swin-Tiny): The Swin-Tiny variant of the Swin Transformer [4] is implemented.Similar to ResNet-34, the Swin-Tiny backbone is configured to output four multi-scale feature maps from its stages.

Four multi-scale feature maps from the encoder are individually passed through 1×1 convolutional layers to project their channel dimensions into a uniform embedding size. Three stacked ScratchFusionBlock modules iteratively fuse the features, starting from the lowest resolution. Each block performs 2× Bilinear Upsampling of the current aggregated feature map and adds it element-wise to the corresponding higher-resolution feature map from the encoder (the skip connection). The combined feature map is then refined using a sequence of two 3×3 Convolutional layers, each followed by Batch Normalization and ReLU activation. The final aggregated feature map is processed by a sequence of upsampling and convolution layers, terminating with a Sigmoid activation to produce a normalized depth map. This is then scaled by the maximum predicted depth.

Loss Function and Metrics

Loss Function: The primary optimization objective is the Scale-Invariant Logarithmic Loss (SILogLoss) [7], which is a robust metric for MDE that minimizes the variance of the log-depth ratio.

$$\mathcal{L}_{\text{SILog}} = 10 \cdot \sqrt{\text{Var}(g) + 0.15 \cdot \text{Mean}(g)^2}$$

where $g = \log(\hat{y}) - \log(y)$, y is the predicted depth, and y is the ground truth depth.

Optimization: The AdamW optimizer is utilized, alongside an aggressive training schedule with a Cosine Annealing Learning Rate Scheduler to facilitate convergence.

Metrics: Model performance is assessed using standard MDE metrics:

- AbsRel (Absolute Relative Error) (Lower is better).
- RMSE (Root Mean Squared Error) (Lower is better).
- Accuracy Thresholds $\delta 1, \delta 2, \delta 3$: Percentage of pixels whose predicted depth is close to the ground truth $\max\left(\frac{y}{\hat{y}}, \frac{\hat{y}}{y}\right) < 1.25^i$ (Higher is better).

4.3 Implementation Details

Frameworks and Libraries: The project was entirely implemented using the PyTorch framework, leveraging the timm library for pre-trained backbone access and the datasets library for handling the NYU Depth V2 data.

Dataset and Preprocessing: Images were resized to 224×224 and normalized with ImageNet mean/variance. Depth targets were clamped between 0.001 and 10 meters.

Table 1: Training and Fine-tuning Hyperparameters

Parameter	**Initial Training**	**Fine-tuning**
Optimizer	AdamW	AdamW
Learning Rate (LR)	1×10^{-3} (Cosine Annealing)	1×10^{-5} (Cosine Annealing)
Batch Size	128	16
Epochs	40	10

Training utilized Automatic Mixed Precision (AMP) with torch.bfloat16 for faster and more memory-efficient training, a Gradient Scaler, and Gradient Clipping (max-norm=1.0).

Hardware: All training was performed on a high-performance GPU (specifically, an NVIDIA A100 GPU available through the Colab Pro environment).

5 Data and Experimental Setup

5.1 Datasets or Data Sources

The project utilizes the NYU Depth V2 dataset [5], a standard benchmark for monocular depth estimation in indoor scenes.

Data Source and Statistics: The dataset consists of RGB images and corresponding depth maps captured by a Microsoft Kinect sensor. The models were trained and evaluated on two different partitions of the NYU Depth V2 dataset using the Hugging Face datasets library for convenient access.

- Initial Training: Used the sayakpaul/nyu-depth-v2 partition, leveraging its predefined large splits for robust initial training.
- Fine-Tuning: Used the 0jl/NYUv2 partition to adhere to a common academic split, specifically 795 images for training and 654 images for testing.

Preprocessing and Standardization:

- Image Input: All RGB images were resized to a fixed resolution of (224×224) pixels using bilinear interpolation. They were then standardized using the mean and standard deviation derived from the ImageNet dataset.
- Depth Target: Ground truth depth maps were also resized to (224×224) using nearest neighbor interpolation to preserve boundary information. Depth values were converted to meters and clamped to the valid range of 0.001 to 10.0 meters.

5.2 Baselines or Comparison Points

Standard Baseline Model

ResNet-34 [3] is selected as the standard, lightweight CNN baseline. It represents a highly optimized and widely used feature extractor within the traditional CNN architecture.

Existing Systems and Prior Results from the Literature

Performance leaders like the Dense Prediction Transformer (DPT) [6] rely on large-scale pre-trained Vision Transformers (e.g., ViT-L), which require massive pre-training on external datasets (e.g., MiDaS Mix5/6). This project compares Swin-Tiny against these results, assessing its competitiveness in a resource-constrained and lightweight setting, without dependence on extensive external pre-training.The performance of the Swin-Tiny [4] is compared with published results from other competitive lightweight MDE models in the literature, such as AdaBins [1], to position the efficacy of the lightweight Transformer backbone.

Ablated Versions and Control of the Own Method

The central comparison requires isolating the performance contribution of the encoder. Therefore, both the ResNet-34 and Swin-Tiny models utilize an identical Lightweight Multi-Scale Fusion Decoder.

5.3 Evaluation Protocol

Model performance is comprehensively evaluated using a set of established monocular depth estimation metrics, focusing on both error minimization and accuracy thresholds: Error Metrics:

AbsRel (Absolute Relative Error)

RMSE (Root Mean Squared Error): Measures the standard deviation of the prediction errors.

Accuracy Thresholds:

δi: Calculates the percentage of pixels where the ratio between the predicted depth and the ground truth is less than 1.25^i.

6 Results

Table 2: Quantitative Results

Metric	ResNet-34 (Base)	Swin-Tiny (Transformer)
AbsRel (↓)	0.1259	0.1523
RMSE (↓)	0.5146	0.5876
δ_1 (↑)	0.8642	0.8122
δ_2 (↑)	0.9745	0.9616
δ_3 (↑)	0.9936	0.9903
Parameters (M)	22.39	28.51
G-MACs (Giga)	5.42	10.05

Visualizations and Error Cases

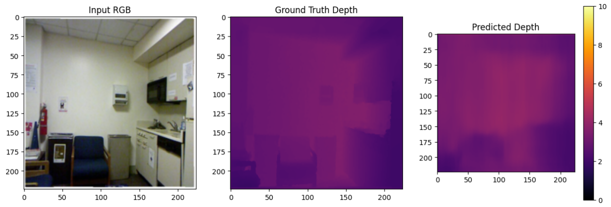

Figure 2: Pretrain Swin-Tiny Example

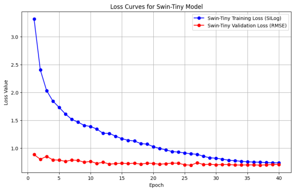

Figure 3: Pretrain Swin-Tiny Loss Curve

(Pretrain ResNet34 Loss Curve)

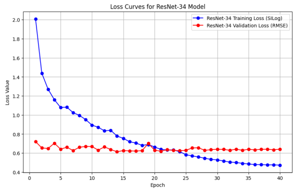

Figure 4: Pretrain ResNet34 Loss Curve

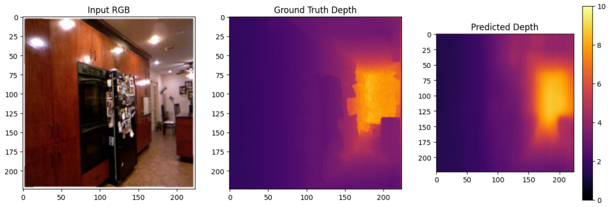

Figure 5: Finetuned Swin-Tiny Example

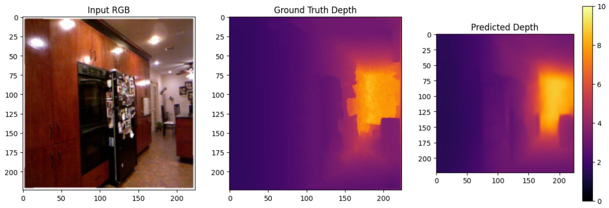

Figure 6: Finetuned ResNet34 Example

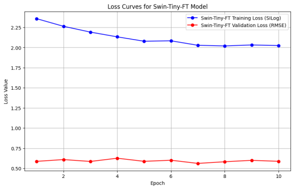

Figure 7: Finetuned Swin-Tiny Loss Curve

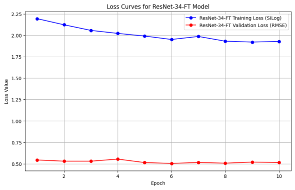

Figure 8: Finetuned ResNet34 Loss Curve

In this specific experimental setup, the ResNet-34 model emerged as the superior choice across all measured metrics, both in terms of accuracy and efficiency.

- Accuracy: ResNet-34 significantly outperformed Swin-Tiny on all depth prediction metrics, especially the crucial AbsRel (0.1259 vs 0.1523) and δi accuracy (0.8642 vs 0.8122). This indicates that the features extracted by the ResNet-34 encoder, when processed by the shared fusion decoder, resulted in more reliable and accurate final depth predictions.
- Trade-offs: This experiment revealed that ResNet-34 was superior in every measured aspect. ResNet-34 was fast in inference speed (5.42 G-MACs vs. 10.05 G-MACs) and required less memory (22.39 M Parameters vs. 28.51 M Parameters). the specific implementation of Swin-Tiny feature projection and fusion within the depth decoder resulted in a larger computational overhead compared to the ResNet-34 path.

Regardless of the backbone architecture—whether the traditional Convolutional Neural Network (ResNet-34) or the novel Vision Transformer (Swin-Tiny)—employing the low-learning-rate fine-tuning phase significantly improved the model accuracy on the standard NYU Depth V2 test set.

7 Analysis and Discussion

Our models operate in a highly restricted efficiency niche (5 to 10 G-MACs). Compared to state-of-the-art systems like AdaBins (AbsRel 0.111 at 50 G-MACs) and DPT-L (AbsRel 0.088 at 250 G-MACs), our models are orders of magnitude more efficient, accepting a modest accuracy drop.

Table 3: Placeholder Caption

Model	AbsRel (NYUv2 ↓)	G-MACs (↓)
DPT-Hybrid-L	≈ 0.088	≈ 250
AdaBins	≈ 0.111	≈ 50
Our ResNet-34	0.1259	5.42
Our Swin-Tiny	0.1523	10.05

The ResNet-34 prediction maps display sharper, cleaner object boundaries (e.g., table edges, cabinet transitions).Swin-Tiny's predictions suffer from noticeable ”depth smearing” or blurring in areas of depth discontinuity and poor definition of fine structures. This qualitative error directly explains its higher AbsRel, demonstrating that its features lack the necessary local precision for accurate geometric reconstruction.

8 Limitations and Ethical Considerations

Feature-Decoder Mismatch: The primary issue was using a shared CNN-optimized decoder. The advanced features from the Swin-Tiny encoder were not effectively processed, resulting in poor edge fidelity (depth smearing) and lower accuracy compared to the ResNet-34.

Inefficient Complexity: The Swin-Tiny's higher 10.05 G-MACs workload proved inefficient, failing to yield a better accuracy/cost ratio than the simpler ResNet-34.

Data Constraint: The lack of massive-scale external pretraining limited the Swin-Tiny's ability to learn robust, generalizable features, which are crucial for Transformer performance.

Possible Misuse: The ability to quickly and efficiently estimate 3D geometry from 2D images, especially using lightweight models, poses risks related to unauthorized 3D reconstruction of private spaces, enhancing surveillance capabilities, or assisting in object tracking and autonomous intrusion systems.

9 Conclusion and Future Work

This project evaluated ResNet-34 (CNN) and Swin-Tiny (Transformer) for Monocular Depth Estimation (MDE) under ultra-low computational constraints (targeting edge deployment).The ResNet-34 architecture proved optimal, achieving the highest accuracy (AbsRel 0.1259) with the lowest cost (5.42 G-MACs). This confirms the superior efficiency trade-off of the optimized CNN for resource-constrained MDE.The two-stage training strategy was highly effective, yielding a significant 16.2% error reduction for ResNet-34.Swin-Tiny underperformed due to a feature-decoder mismatch with the shared CNN-optimized decoder, resulting in poor edge fidelity and depth smearing.

Future efforts will focus on addressing the Swin-Tiny's limitations and further optimizing the MDE pipeline: Design a Transformer-native decoder to effectively process Swin-Tiny's hierarchical features and restore boundary precision.Conduct deployment and generalization tests on edge devices and outdoor datasets (e.g., KITTI) to measure real latency and model transferability.

Reproducibility and Artifacts

Code Resources can be download at github

Models can be download at Google Drive

References

[1] Shariq Bhat, Ibraheem Alhashim, and Helmut Wonka. Adabins: Depth estimation using adaptive binning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 4009–4019, 2021.

[2] 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. *International Conference on Learning Representations (ICLR)*, 2021.

[3] 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 (CVPR)*, pages 770–778, 2016.

[4] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baofeng Dai. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021.

[5] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[6] René Ranftl, Alexey Bochkovskiy, and Michael Wimmer. Vision transformers for dense prediction. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12179–12189, 2021.

[7] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Monocular depth estimation using relative depth, geometry and context cues. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1161–1168, 2027.

[8] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision (ECCV)*, pages 746–760, 2012.