Monocular Depth Estimation using Deep Edge Intelligence

Irfan Ali Sadab, Md Arafat Islam, Rashik Iram Chowdhury, and Md. Ishan Arefin Hossain
Department of Computer Science and Engineering, North South University, Bangladesh
Email: irfan.sadab@northsouth.edu, arafat.islam21@northsouth.edu, rashik.chowdhury@northsouth.edu, ishan.hossain@northsouth.edu

***Abstract*—Monocular depth estimation, a crucial challenge in computer vision, has significant applications across various domains, including robotics, augmented reality, and autonomous systems. This study explores the efficacy of multiple Convolutional Neural Network (CNN) backbones as encoders within a U-Net architecture, including DenseNet121, InceptionV3, EfficientNetV2, MobileNetV2, and ResNet50, each paired with custom decoders incorporating squeeze-and-excitation blocks to enhance depth prediction accuracy. We introduce U-ResNet50, a model that combines ResNet-50 with an optimized custom decoder, designed to achieve high depth estimation accuracy. Trained on the NYU Depth V2 dataset with Structural Similarity Index Measure (SSIM) as the primary loss function, the models were evaluated on metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and SSIM accuracy. U-ResNet50 outperformed the others, achieving 94.62% SSIM accuracy and an RMSE of 0.0474, balancing efficiency and precision. To further enhance its practical application, we quantized the U-ResNet50 model, reducing its size and computational requirements while maintaining performance. The quantized model was deployed via a Streamlit web interface, demonstrating its potential for edge applications in robotics and augmented reality.**

***Index Terms*—Monocular Depth Estimation, Deep Learning, Quantization, Edge Intelligence**

I. Introduction

Monocular depth estimation—the task of predicting depth information from a single RGB image—is increasingly crucial in computer vision, especially as applications grow in fields such as robotics, augmented reality, autonomous driving, and medical imaging. Unlike stereo vision or LiDAR systems, monocular depth estimation requires only a single camera, making it a cost-effective and versatile solution that aligns with the principles of Sustainable Technology by reducing reliance on expensive and resource-intensive hardware. However, deriving accurate 3D information from 2D images poses significant challenges due to inherent ambiguities.

Recent advancements in deep learning, particularly convolutional neural networks (CNNs), have greatly improved monocular depth estimation, offering potential applications in environments where efficient resource utilization is prioritized. CNNs can extract complex features, crucial for interpreting depth cues in images. This paper investigates the efficacy of several advanced CNN architectures—DenseNet121, InceptionV3, EfficientNetV2, MobileNetV2, and ResNet50—as backbones for monocular depth estimation. Each architecture provides unique benefits in terms of computational efficiency, model complexity, and feature extraction, aspects that are fundamental to developing sustainable AI solutions.

Our approach includes a custom decoder with squeeze-and-excitation blocks, designed to refine feature representations and improve depth map predictions with minimal computational overhead. By enhancing model efficiency and accuracy, this work contributes to the development of AI systems that meet Industry 5.0's emphasis on resource efficiency and sustainable scalability. The models are trained on the NYU Depth V2 dataset, employing the structural similarity index (SSIM) as the primary loss function due to its ability to measure perceptual similarity. Additionally, we use metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and SSIM accuracy to comprehensively evaluate model performance.

A key contribution of this work is the quantization of the best-performing model, significantly reducing its size and computational requirements while maintaining a reasonable trade-off in accuracy.. This quantized model is deployed in a Streamlit web interface, demonstrating a real-world solution that is not only accessible but also efficient, aligning with sustainable technology goals. The deployment showcases the model's potential in applications such as autonomous navigation and augmented reality.

The primary contributions of this paper are as follows:

- **Architecture Exploration**: A comparative analysis of different CNN backbones for monocular depth estimation, highlighting trade-offs between accuracy and computational efficiency, contributing to resource-efficient solutions.
- **Custom Decoder Design**: Introduction of a novel decoder architecture with squeeze-and-excitation blocks, enhancing depth detail capture without extensive computational demands.
- **Comprehensive Evaluation**: Extensive evaluation of the models using multiple metrics, ensuring robust performance assessment that supports the development of sustainable AI-driven systems.
- **Practical Deployment**: Quantization and deployment of the best-performing model in a Streamlit web interface, demonstrating practical application, accessibility, and the potential for energy-efficient use in Industry 5.0 contexts.

The remainder of this paper is organized as follows: Section II reviews related work in monocular depth estimation. Section III details the proposed methodology, including the network architecture and training procedures. Section IV presents the results. Section V discusses the limitations of the current approach, while Section VI concludes the paper and explores potential future work.

II. Literature Review

Monocular depth estimation is an evolving field focused on improving depth accuracy, scale awareness, and robustness in diverse environments. This review covers recent advancements in model architecture, training strategies, and adaptability.

Y. Zhang et al. [1] introduced a parallel decoder architecture that tackles spatial information loss in CNNs by combining global and local depth predictions from an EfficientNet-B7 encoder. Through multi-scale feature extraction and a depth interval-based loss, the model achieves state-of-the-art performance on NYU Depth v2 and KITTI datasets, with high accuracy and low computational demands—useful for 3D reconstruction and scene understanding applications.

V. Guizilini et al. [2] addressed the challenge of scale variations in novel environments with a scale-aware model that uses domain adaptation techniques, achieving effective zero-shot depth estimation without extensive labeled data.

DepthFM, proposed by *M. Gui et al.* [3], incorporates optical flow-based alignment for temporal consistency in video-based depth estimation, making it suitable for applications like autonomous driving that require coherent depth information across sequences.

URCDC-Depth [4] combines residual learning with cross-dimensional feature extraction, balancing accuracy and computational efficiency to support depth estimation across diverse scenarios, showing potential for real-time deployment.

SwiftDepth by *Luginov et al.* [5] uses a hybrid CNN and Vision Transformer to create a self-supervised model that captures both local and global features to deliver efficient, robust depth estimation on mobile devices.

CATNet by *Tang et al.* [6] integrates Convolutional Attention and Transformer modules, improving depth estimation in complex scenes. The model's Multi-Dimensional Convolutional Attention and Dual Attention Transformer modules enhance performance on datasets like KITTI and NYUv2.

The DisDepth model, introduced by *Zheng et al.* [7], employs cross-architecture knowledge distillation, transferring knowledge from a Transformer-based teacher to a CNN-based student model, effectively improving performance in resource-constrained environments.

Tadepalli et al. [8] leverage EfficientNet-B0 within an encoder-decoder framework, achieving competitive metrics such as F1-score, Jaccard score, and Mean Absolute Error, making it effective for applications requiring accuracy and computational efficiency.

Lite-Mono by *Zhang et al.* [9] introduces a self-supervised hybrid CNN and Transformer architecture that combines Consecutive Dilated Convolutions and Local-Global Features Interaction modules, achieving high accuracy with minimal parameters on KITTI and Make3D datasets.

Saxena et al. [10] proposed DepthGen, which uses diffusion models to handle noisy or incomplete data.The model captures multimodal depth distributions and effectively handles ambiguous surfaces, such as transparent and reflective materials, achieving state-of-the-art performance on the NYU dataset and competitive results.

Together, these works demonstrate significant progress in monocular depth estimation through innovative architectures, training strategies, and efficient models, emphasizing the field's focus on accuracy and adaptability for real-world applications. However, despite these advancements, the trade-off between efficiency and accuracy still remains a challenge.

III. Methodology

In this work, we propose a customized model for monocular depth estimation based on the U-Net architecture, utilizing ResNet-50 as the backbone, referred to as U-ResNet50. Additionally, we experimented with U-Net architectures using other backbones, including DenseNet121, InceptionV3, EfficientNetV2, and MobileNetV2, to explore their effectiveness in monocular depth estimation. To further optimize U-ResNet50 for practical deployment, we quantized the model and deployed it in a web interface, enhancing accessibility and efficiency. We illustrate the comprehensive workflow from data processing to model deployment in Fig. 1.

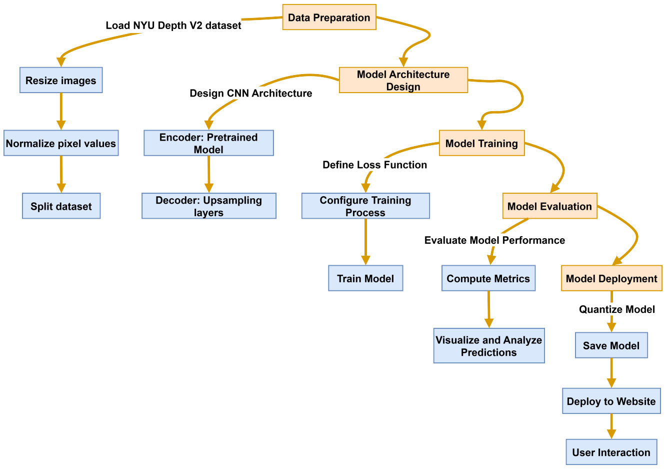


Fig. 1: From Data to Deployment: The Project Workflow

Data Preprocessing: The dataset used for this study is the NYU Depth Dataset V2, which includes RGB images and corresponding depth maps. Preprocessing steps are as follows:

- **Data Loading and Shuffling**: The dataset was loaded and shuffled to ensure randomness in the training process.
- **Data Sampling**: Due to computational constraints, the dataset was reduced to 5,000 samples, though the original dataset contains significantly more data.
- **Data Splitting**: We split the dataset into training, validation, and test sets with proportions of 80%, 10%,

and 10%, respectively, to facilitate model training and validation while reserving a portion for final evaluation.

Data Generation: A custom data generator was implemented using TensorFlow's `Sequence` utility for efficient batch processing and data augmentation. Key features of the data generator include:

- **Batch Processing**: The generator yields batches of preprocessed image-depth pairs, ensuring efficient memory usage during training.
- **Data Augmentation**: Random horizontal flipping was applied to both RGB images and depth maps to enhance the model's generalization capabilities.
- **Resizing**: All images and depth maps were resized to 224x224 pixels to fit the input requirements of the neural network.

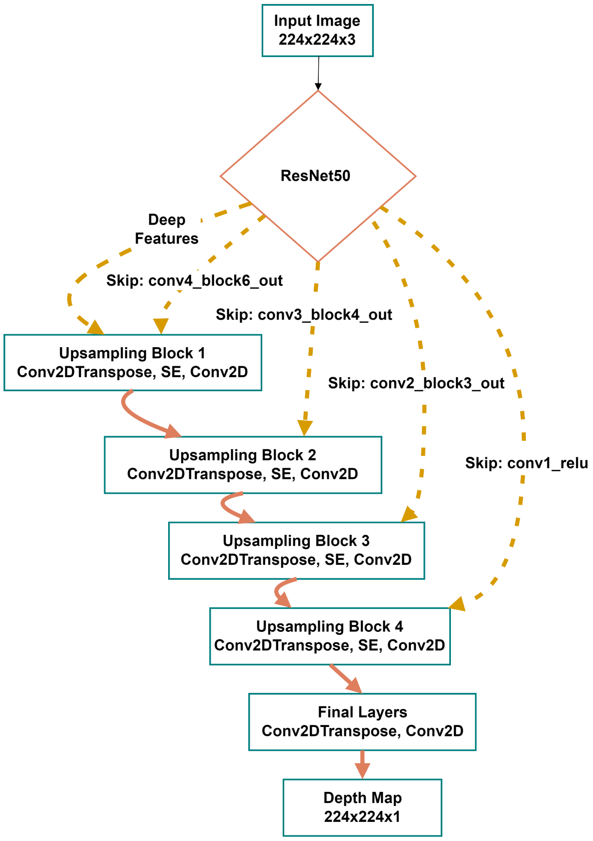


Fig. 2: Proposed U-ResNet50 Model Architecture

Model Architecture: The U-ResNet50 architecture (Fig. 2) combines ResNet-50 as the encoder with a custom-designed decoder, optimized specifically for depth estimation. The ResNet-50 backbone, pretrained on ImageNet dataset, provides robust feature extraction, while the decoder refines and upsamples these features to produce enhanced depth predictions. The key components of this architecture are described below:

Backbone (ResNet-50): ResNet-50, known for its strong feature extraction capabilities, serves as the encoder in our model. By using a pretrained ResNet-50, the model can efficiently capture hierarchical features essential for depth estimation.

Skip Connections: We integrate skip connections from four key layers of the ResNet-50 backbone (`conv1_relu`, `conv2_block3_out`, `conv3_block4_out`, and `conv4_block6_out`). These connections provide spatial information at different scales, which is crucial for reconstructing fine-grained details in the depth map.

Custom Decoder: The custom decoder in U-ResNet50 is designed to balance computational efficiency with depth estimation accuracy. It includes several upsampling blocks, each of which utilizes:

- **Transposed Convolutions for Upsampling**: To progressively increase the spatial resolution, we use `Conv2DTranspose` layers at each stage. These layers upsample the low-resolution feature maps from the encoder, essential for depth prediction.
- **Concatenation with Skip Connections**: At each upsampling block, the model concatenates the upsampled features with corresponding skip connections, allowing for richer feature representations by combining high-level and low-level information.
- **Squeeze-and-Excitation (SE) Blocks**: The custom decoder integrates squeeze-and-excitation blocks. Each SE block adaptively recalibrates the channel-wise feature responses by globally pooling each feature map and generating a channel-wise weight vector. This vector highlights important features while reducing noise, thereby enhancing depth cues. The SE blocks thus play a critical role in making depth predictions more precise and reliable without increasing the model's complexity.
- **Depthwise Convolutions and Batch Normalization**: Depthwise convolutions are used to reduce parameter count, enhancing computational efficiency. Batch normalization layers follow these convolutions to stabilize learning and accelerate convergence, particularly beneficial when training on smaller datasets.
- **Leaky ReLU Activations and Dropout**: Leaky ReLU activations introduce non-linearity, helping the model learn intricate depth relationships, while dropout (set to 0.3) mitigates overfitting, contributing to improved generalization.

Final Layers: The final layers of the decoder include a transposed convolution layer with 32 filters, a 3x3 kernel, and a stride of 2 for upsampling. This is followed by a 1x1 convolution layer with a sigmoid activation, producing a single-channel depth map with normalized pixel values.

Loss Function: The primary loss function is the Structural Similarity Index Measure (SSIM), selected for its ability to capture structural information essential in depth estimation. The SSIM loss is defined as:

$$\text{SSIM Loss}(x, y) = 1 - \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

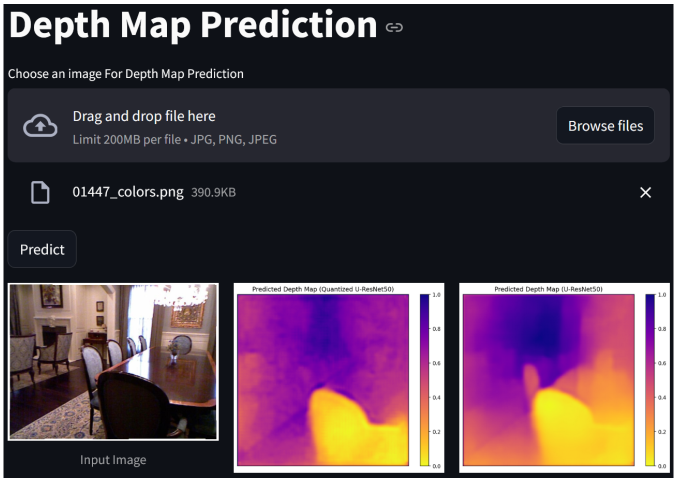
where x and y are the images being compared, μ, σ, and C are constants, means, and variances of x and y, respectively.

Optimizer and Metrics: The Adam optimizer was used with a learning rate of $1e^{-4}$. Model performance was evaluated using SSIM, Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

Early Stopping: To prevent overfitting, early stopping was implemented with a patience of 10 epochs, based on validation loss.

Quantization: To enhance efficiency, we applied post-training dynamic range quantization to U-ResNet50, where the model parameters were converted from `float32` to `int8`. The `tf.lite.Optimize.DEFAULT` optimization setting enabled TensorFlow Lite to analyze model activation ranges and apply quantization. This process significantly reduced the model size and computational requirements while maintaining performance, making the model more suitable for deployment on edge devices.

Deployment: For accessible inference, both the original and quantized versions of U-ResNet50 were deployed in a Streamlit web interface. The quantized model, converted to TensorFlow Lite (TFLite) format, optimizes inference on resource-constrained devices. Users can upload RGB images, which are preprocessed to 224x224 pixels. The application displays the original image and the depth maps generated by the models, using a reverse plasma colormap for enhanced visualization as highlighted in Fig. 3.



Depth Map Prediction

Choose an image For Depth Map Prediction

Drag and drop file here

Limit 200MB per file • JPG, PNG, JPEG

01447_colors.png 390.9KB

Input Image

Browse files
Predict
Predicted Depth Map (Quantized U-ResNet50)
Predicted Depth Map (U-ResNet50)
1.0
0.8
0.6
0.4
0.2
0.0

Fig. 3: Web interface displaying depth map predictions generated by both the quantized and baseline U-ResNet50 models.

IV. Results

To evaluate the performance of the models, following metrics were used: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Structural Similarity Index Measure (SSIM). These metrics provide a comprehensive assessment of the model's accuracy and structural similarity.

A. Mean Absolute Error (MAE)

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It is calculated as follows:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{1}$$

where n is the number of samples, y_i is the ground truth value, and $\hat{y}_i$ is the predicted value.

B. Mean Squared Error (MSE)

MSE measures the average of the squares of the errors. It is more sensitive to large errors compared to MAE. The equation for MSE is:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{2}$$

C. Root Mean Squared Error (RMSE)

RMSE is the square root of MSE and provides an error metric in the same units as the original data. It is given by:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{3}$$

D. Structural Similarity Index Measure (SSIM)

SSIM is a perceptual metric that quantifies the image quality degradation caused by processing such as data compression or transmission losses. It is particularly useful for depth estimation as it considers changes in structural information. SSIM is calculated using the following equation:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{4}$$

where μ_x and μ_y are the average of x and y, σ_x^2 and σ_y^2 are the variance of x and y, σ_{xy} is the covariance of x and y, and C_1, C_2 are constants to stabilize the division.

These metrics collectively provide a robust evaluation of the depth estimation models' performance, covering both error magnitude and structural similarity.

TABLE I: Models' Performance Metrics

Model	SSIM L.	MAE	MSE	RMSE	SSIM Acc.
U-DenseNet121	0.0775	0.0446	0.0066	0.0812	0.9225
U-InceptionV3	0.1392	0.0684	0.0105	0.1025	0.8608
U-EfficientNetV2	0.0730	0.0420	0.0046	0.0676	0.9270
U-MobileNetV2	0.0618	0.0312	0.0027	0.0522	0.9382
U-ResNet50	**0.0538**	**0.0285**	**0.0023**	**0.0474**	**0.9462**
Metric3Dv2 [11]	-	-	-	0.183	-
UniDepth (Zero-shot) [12]	-	-	-	0.201	-
HybridDepth [13]	-	-	-	0.128	-
Depth Anything [14]	-	-	-	0.206	-
ECoDepth [15]	-	-	-	0.218	-

Note: Values are rounded to four decimal places.

In analyzing the performance metrics of U-Net architecture-based depth estimation models using different backbone models, U-ResNet50 stands out with the lowest values in Mean Absolute Error (MAE) (0.0285), Mean Squared Error (MSE) (0.0023), and Root Mean Squared Error (RMSE) (0.0474),indicating superior accuracy and error reduction as shown in Table I. It also achieves the highest Structural Similarity Index Measure (SSIM) accuracy of 0.9462, suggesting better structural similarity (Fig. 4) and perceptual quality in depth estimation. U-MobileNetV2 follows closely, with competitive metrics but slightly higher error values. U-EfficientNetV2, U-DenseNet121, and U-InceptionV3 demonstrate progressively higher error rates and lower SSIM accuracies, with U-InceptionV3 being the least effective among the five models.

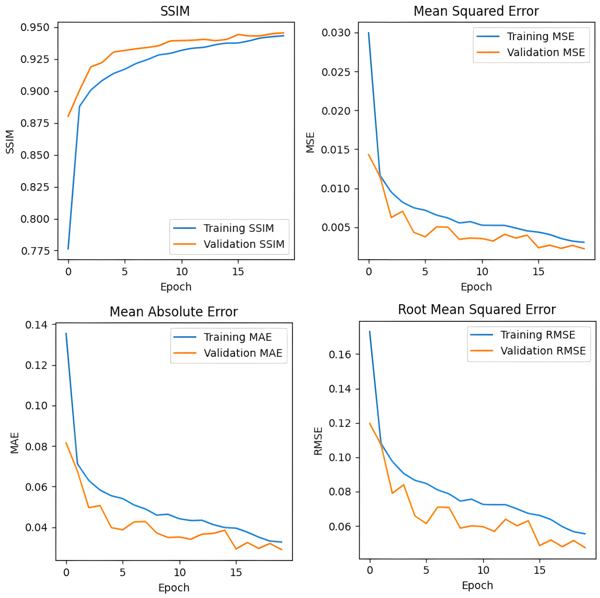


Fig. 4: Performance metrics of the Proposed U-ResNet50 Model

When comparing these results to recent work in the field, our proposed models, particularly U-ResNet50, demonstrate superior performance. The RMSE values reported in recent papers highlight this advantage. For instance, Metric3Dv2 achieves an RMSE of 0.183 [11], while UniDepth (Zero-shot) reports an RMSE of 0.201 [12], both significantly higher than the 0.0474 RMSE achieved by U-ResNet50. Similarly, HybridDepth records an RMSE of 0.128 [13], and Depth Anything shows an RMSE of 0.206 [14], further underscoring the superior accuracy of our model. Even ECoDepth, with an RMSE of 0.218 [15], cannot match the performance of U-ResNet50, in terms of RMSE.

These comparisons indicate that the U-Net architecture models, particularly the proposed U-ResNet50, outperform recent state-of-the-art models in terms of RMSE, demonstrating better error reduction and accuracy. The lower RMSE values achieved by our models indicate more precise depth estimation, which is crucial for applications requiring high accuracy and reliability. This improvement can be attributed to the effective use of the ResNet50 backbone and the custom decoder architecture employed in our models. Overall, the integration of these advanced backbone networks and customized decoder designs contributes to the robustness and high performance of our proposed models in monocular depth estimation (Fig. 5).

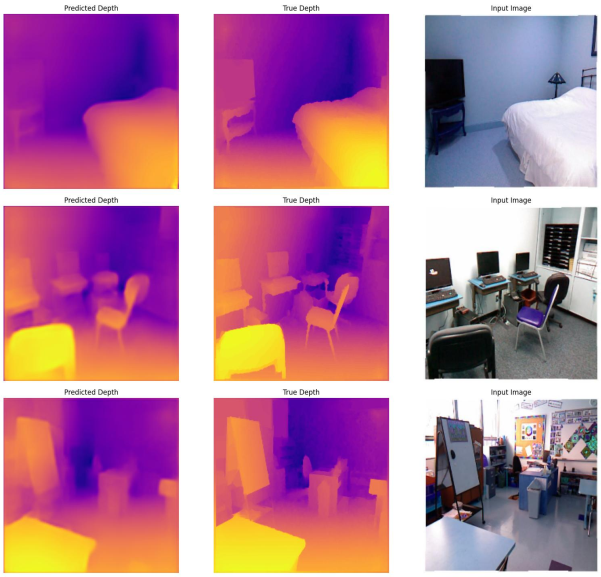


Fig. 5: Visualizing Depth Prediction Accuracy of the Proposed U-ResNet50 model: Comparing predicted depth maps with true depth maps and original input images

Quantization:

TABLE II: Model Size

Original(MB)	Quantized(MB)
148.740	13.005

Note: Values are rounded to three decimal places.

The size of the U-Net architecture-based U-ResNet50 model has been significantly reduced by the quantization process, making the model more lightweight and efficient. The original model size of 148.740 MB was reduced to 13.005 MB after quantization (Table II). This reduction, approximately 91.18%, allows for the deployment of the model in resource-constrained environments with a reasonable tradeoff in performance. The quantized model has been deployed into a Streamlit web interface, demonstrating its practicality and effectiveness in real-world applications.

V. Limitations

While our study has advanced monocular depth estimation via deep learning, certain limitations warrant further exploration.

Resource Constraints and Impacts: Due to computational constraints, we used a 5,000-sample subset of the NYU

Depth V2 dataset. This limitation may affect the comparability of our results with models trained and evaluated on the full dataset. Nevertheless, our models achieved competitive accuracy, demonstrating the effectiveness of our approach. Training and evaluating these models on the full dataset could potentially improve performance, particularly in more diverse scenes, by capturing a wider range of visual contexts.

Generalization Limits: Although our models performed well on NYU Depth V2, their adaptability to other datasets, like KITTI, and real-world variability (e.g., lighting, weather, object geometry) is uncertain. Testing on diverse datasets is essential for confirming robustness.

Quantization Trade-offs: Quantization reduced model size for edge deployment but introduced slight accuracy trade-offs. Advanced strategies, like mixed-precision quantization, could further minimize accuracy loss while improving efficiency.

Augmentation and Training Scope: We limited data augmentation to horizontal flipping; additional techniques (e.g., rotation, scaling) and semi-supervised learning could enhance robustness.

Interpretability Challenges: The complexity of deep learning architectures, like ResNet50, makes model interpretability challenging. Future work could explore feature visualization to clarify the decision-making process, supporting safer application deployment.

VI. Conclusion

In conclusion, this research successfully addresses the challenge of monocular depth estimation by developing a robust and versatile model that integrates advanced deep learning techniques. The proposed solution effectively overcomes the limitations of traditional methods, offering significant improvements in both efficiency and accuracy. Key methodologies employed include batch normalization, dropout layers, and efficient convolutional techniques, all carefully designed to optimize depth feature extraction from single images. Despite the resource constraints that prevented training on the full NYU Depth Dataset V2, which contains over 50,000 samples, our model demonstrated impressive performance on a subset of 5,000 samples. This highlights the model's potential even with limited data. However, we acknowledge that training on the complete dataset could further enhance the model's accuracy and robustness.

We trained five models with different backbones: DenseNet121, InceptionV3, EfficientNetV2, MobileNetV2, and ResNet50. The best-performing model, using ResNet50 as the backbone, achieved the highest SSIM accuracy of 94.62%, followed by the MobileNetV2-based model with 93.82%. These results underscore the effectiveness of leveraging advanced CNN architectures and techniques for depth feature extraction.

Additionally, we successfully quantized our model, reducing its size from 148.740 MB to 13.005 MB, making it lightweight and efficient. Both the quantized and original models have been deployed in a Streamlit web interface, demonstrating their practicality and effectiveness in real-world scenarios. This deployment highlights the model's readiness for integration into various platforms and its potential for widespread use in depth estimation tasks, particularly in applications requiring low-latency and high efficiency, such as autonomous navigation and augmented reality.

Looking ahead, future work will focus on improving the model's accuracy by incorporating additional techniques, such as advanced data augmentation, semi-supervised learning, and the integration of geometric constraints. These improvements will aim to refine the model's ability to discern subtle depth cues and nuances inherent in complex visual scenes. Furthermore, we plan to deploy our model on edge devices, leveraging quantization techniques to reduce complexity and improve efficiency. This step is crucial for real-world applications, especially in scenarios requiring low-latency and high-efficiency depth estimation, such as autonomous navigation and augmented reality.

References

[1] J. Liu and Y. Zhang, "High quality monocular depth estimation with parallel decoder," *Scientific Reports*, vol. 12, no. 1, Oct. 2022, doi: 10.1038/s41598-022-20909-x.

[2] V. Guizilini, I. Vasiljevic, D. Chen, R. Ambrus, and A. Gaidon, "Towards Zero-Shot Scale-Aware Monocular Depth Estimation," *arXiv preprint arXiv:2306.17253*, Jun. 29, 2023.

[3] M. Gui et al., "DepthFM: Fast Monocular Depth Estimation with Flow Matching," arXiv.org, 2024. https://arxiv.org/abs/2403.13788

[4] S. Shao, Z. Pei, W. Chen, R. Li, Z. Liu, and Z. Li, "URCDC-Depth: Uncertainty Rectified Cross-Distillation with CutFlip for Monocular Depth Estimation," *arXiv preprint arXiv:2302.08149*, 2023.

[5] A. Luginov and I. Makarov, "SwiftDepth: An Efficient Hybrid CNN-Transformer Model for Self-Supervised Monocular Depth Estimation on Mobile Devices," *IEEE ISMAR Adjunct*, Oct. 2023, doi: 10.1109/ismar-adjunct60411.2023.00137.

[6] S. Tang, T. Lu, X. Liu, H. Zhou, and Y. Zhang, "CATNet: Convolutional attention and transformer for monocular depth estimation," *Pattern Recognition*, vol. 145, p. 109982, Sep. 2023, doi: 10.1016/j.patcog.2023.109982.

[7] Z. Zheng, T. Huang, G. Li, and Z. Wang, "Promoting CNNs with Cross-Architecture Knowledge Distillation for Efficient Monocular Depth Estimation," *arXiv preprint arXiv:2404.16386v*, 2024.

[8] Y. Tadepalli, M. Kollati, S. Kuraparthi, and P. Kora, "EfficientNet-B0 Based Monocular Dense-Depth Map Estimation," *Traitement du Signal*, vol. 38, no. 5, pp. 1485–1493, Oct. 2021, doi: 10.18280/ts.380524.

[9] N. Zhang, F. Nex, G. Vosselman, and N. Kerle, "Lite-Mono: A Lightweight CNN and Transformer Architecture for Self-Supervised Monocular Depth Estimation," *arXiv preprint arXiv:2211.13202*, Nov. 23, 2022.

[10] S. Saxena, A. Kar, M. Norouzi, and D. J. Fleet, "Monocular Depth Estimation using Diffusion Models," *arXiv preprint arXiv:2302.14816*, 2023.

[11] M. Hu, et al., "Metric3D v2: A Versatile Monocular Geometric Foundation Model for Zero-shot Metric Depth and Surface Normal Estimation," *arXiv preprint arXiv:2404.15506*, 2024.

[12] W. Yin, et al., "UniDepth: Universal Monocular Metric Depth Estimation," *arXiv preprint arXiv:2404.15519*, 2024.

[13] K. Kim, et al., "HybridDepth: Robust Depth Fusion for Mobile AR by Leveraging Depth from Focus and Single-Image Priors," *arXiv preprint arXiv:2405.03252*, 2024.

[14] Y. Ben-Gal, et al., "Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data," *arXiv preprint arXiv:2405.01935*, 2024.

[15] M. Liu, et al., "ECoDepth: Effective Conditioning of Diffusion Models for Monocular Depth Estimation," *arXiv preprint arXiv:2405.00918*, 2024.