Monocular Metric Depth Estimation using ResNet-34 and Curriculum Learning

Ibtehaj Haider (22i-0767)

Department of Computer Science
FAST NUCES ISLAMABAD
Digital Image Processing - Fall 2025
Instructor: Dr. Atif Mughees

Abstract. Depth estimation from a single RGB image is a fundamental task in computer vision with applications in robotics, augmented reality, and scene understanding. This project implements a deep learning pipeline to estimate metric depth using the NYU Depth V2 dataset. A U-Net architecture with a pre-trained ResNet-34 encoder is proposed, trained using a Curriculum Learning strategy optimized for consumer hardware (RTX 3060). Critical data quantization challenges involving 8-bit versus 16-bit discrepancies are addressed, and a composite Edge-Aware Loss function is introduced to resolve boundary blurring. Experimental results demonstrate that the fine-tuned ResNet-34 model achieves a Mean Absolute Error (MAE) of 0.37m on the official test set, with relative errors as low as 2.3% for near-field objects, significantly outperforming the baseline ResNet-18 implementation.

Keywords: Monocular Depth Estimation · ResNet · U-Net · Curriculum Learning · Edge-Aware Loss

1 Introduction

Depth perception is inherent to human vision but remains a challenging ill-posed problem for computer vision systems relying on a single camera. Unlike stereo vision systems that rely on triangulation between multiple viewpoints, or LiDAR systems that use time-of-flight measurements, monocular depth estimation requires a model to learn complex contextual cues, texture gradients, perspective geometry, and relative object sizes to infer distance information from a single image.

The significance of monocular depth estimation extends across multiple domains. In autonomous robotics, accurate depth perception enables safe navigation and obstacle avoidance. Augmented reality applications require precise depth maps to realistically occlude virtual objects behind real-world surfaces. Furthermore, scene understanding for applications such as indoor mapping, assistive technologies for the visually impaired, and 3D reconstruction all benefit from robust depth estimation capabilities.

This project aims to build a robust deep learning system capable of predicting dense, metric depth maps from single RGB images captured in indoor

environments. A critical distinction of this work is the focus on recovering absolute scale measurements in meters rather than relative depth orderings, which is essential for real-world applications requiring precise distance measurements.

The primary contributions of this work include the implementation of a Curriculum Learning training strategy to handle large-scale datasets on consumer hardware, the identification and resolution of critical data preprocessing errors in the NYU Depth V2 dataset, and the development of an Edge-Aware loss function that significantly improves boundary sharpness in predicted depth maps.

1.1 Research Gap

While significant progress has been made in monocular depth estimation through deep learning approaches, several challenges persist. Existing methods often struggle with sharp depth discontinuities at object boundaries, resulting in blurred edges and halo artifacts. Additionally, many state-of-the-art models require extensive computational resources during training, making them inaccessible for implementation on consumer-grade hardware. Finally, inconsistencies in dataset preprocessing and bit-depth encoding remain under-documented, leading to potential training failures when working with public datasets such as NYU Depth V2.

1.2 Project Objectives

The objectives of this project are threefold. First, to develop a practical depth estimation system that operates effectively on consumer-grade GPU hardware with limited VRAM. Second, to improve edge sharpness and boundary definition in predicted depth maps through custom loss function design. Third, to achieve metric-scale accuracy suitable for real-world measurement applications in indoor environments.

1.3 Scope and Limitations

This project focuses exclusively on indoor scene depth estimation using the NYU Depth V2 dataset, which consists of RGB-D image pairs captured in residential and office environments. The scope is limited to static images rather than video sequences, and all training and evaluation are conducted on scenes within a 0-10 meter depth range. Hardware constraints limit the maximum batch size and resolution to 320x320 pixels. The system does not address outdoor scenes, dynamic objects, or transparent surfaces.

2 Literature Review

The field of monocular depth estimation has evolved significantly over the past decade, transitioning from traditional hand-crafted feature approaches to data-driven deep learning methodologies. This section reviews key contributions that have shaped current understanding and practice.

Eigen et al. pioneered the application of deep convolutional neural networks to monocular depth estimation in their seminal 2014 work. Their multi-scale deep network architecture demonstrated that CNNs could learn to predict depth maps directly from single images by training on large-scale datasets. The key innovation was the use of a coarse-to-fine prediction strategy, where a global coarse network predicted overall scene structure, while a fine network refined local details. However, their approach required significant computational resources and produced relatively blurry depth boundaries.

Building upon this foundation, Laina et al. introduced fully convolutional residual networks for depth prediction in 2016. By leveraging ResNet architectures and eliminating fully connected layers, they achieved more efficient training and higher-resolution outputs. Their work demonstrated that deeper networks with residual connections could capture more complex scene geometry. Nevertheless, the models still struggled with preserving fine details at object boundaries, particularly in cluttered indoor environments.

The NYU Depth V2 dataset, introduced by Silberman et al. in 2012, has become a benchmark for indoor depth estimation. This dataset contains over 400,000 RGB-D image pairs captured using Microsoft Kinect sensors in various indoor settings. The official train-test split provides 50,000 training images and 654 test images. While widely adopted, the dataset presents preprocessing challenges due to inconsistent depth encoding formats, a problem that has received limited attention in the literature.

Godard et al. explored unsupervised approaches to monocular depth estimation by exploiting left-right consistency in stereo image pairs. Their method eliminates the need for ground truth depth labels during training, instead using photometric reconstruction loss. While innovative, unsupervised methods typically achieve lower absolute accuracy compared to fully supervised approaches, particularly for metric depth prediction.

Recent works have begun exploring transformer-based architectures and attention mechanisms for capturing long-range dependencies in depth estimation. However, these models typically require substantially more computational resources and training data. The gap in practical implementations for resource-constrained environments motivates the approach taken in this project.

This project differentiates itself by specifically addressing the hardware constraints of consumer GPUs through Curriculum Learning, explicitly resolving dataset preprocessing inconsistencies, and introducing an Edge-Aware loss function to improve boundary sharpness without requiring additional model complexity.

3 Methodology

3.1 Overview

The proposed system employs an encoder-decoder architecture based on the U-Net design, combining a deep convolutional encoder for feature extraction with

a symmetric decoder for dense prediction. The complete pipeline consists of four major stages: image acquisition and preprocessing, feature encoding through ResNet-34, multi-scale feature decoding with skip connections, and depth map prediction with edge-aware refinement.

3.2 System Architecture

The architecture adopts a U-Net design, which has proven highly effective for dense prediction tasks requiring both semantic understanding and spatial precision. The system consists of three main components working in concert.

Encoder (Backbone) A ResNet-34 network pre-trained on ImageNet serves as the feature extraction backbone. This encoder processes the input RGB image through four residual blocks of increasing depth (64, 128, 256, 512 channels), progressively downsampling spatial dimensions while extracting increasingly abstract features. The pre-trained weights provide robust initialization, encoding general visual concepts such as edges, textures, and object patterns learned from over one million diverse images.

The decision to upgrade from an initial ResNet-18 implementation to ResNet-34 was motivated by the need for greater representational capacity to handle complex indoor scenes with multiple objects, occlusions, and varying lighting conditions. The additional residual blocks allow the network to learn more sophisticated feature hierarchies while maintaining reasonable computational requirements.

Decoder The decoder consists of four upsampling stages that progressively reconstruct the spatial resolution from the compressed feature representation back to the original image dimensions (320 x 320 pixels). Each decoder block combines transposed convolutions for upsampling with standard convolutions for feature refinement. This gradual reconstruction allows the network to integrate multi-scale information effectively.

Skip Connections Critical to the architecture are lateral skip connections that concatenate features from corresponding encoder layers directly to decoder layers. These connections serve two purposes: preserving fine-grained spatial details that would otherwise be lost during encoding, and enabling gradient flow during backpropagation to facilitate training of deeper networks. For depth estimation, skip connections are particularly important for maintaining sharp boundaries and small object details.

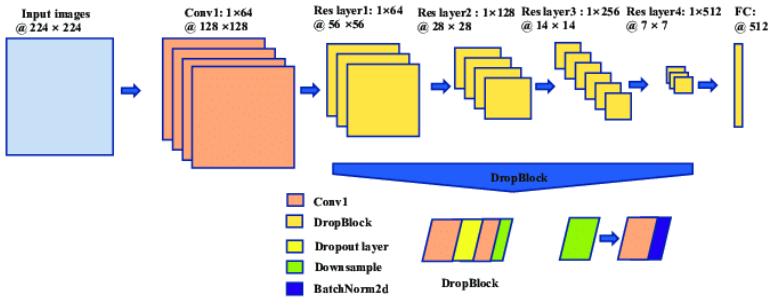

Fig. 1. Proposed U-Net Architecture with ResNet-34 Backbone. The encoder extracts multi-scale features through residual blocks, while the decoder reconstructs spatial resolution via transposed convolutions. Skip connections (gray arrows) preserve fine details.

3.3 Curriculum Learning Strategy

Training deep neural networks on large datasets presents significant memory challenges, particularly on consumer hardware. The NYU Depth V2 training set contains approximately 50,000 high-resolution images, which cannot be loaded entirely into the 12GB VRAM available on an RTX 3060 GPU.

To address this constraint, a Curriculum Learning approach was implemented. The complete dataset was partitioned into ten sequential chunks, each containing approximately 5,000 images. Training proceeds in two distinct phases.

In Phase One (Base Training), the model is trained sequentially on each chunk for five epochs before moving to the next chunk. This allows the model to gradually encounter diverse scene types, room layouts, and object arrangements without overwhelming the memory budget. The learning rate is set to 1×10^{-4} during this phase to enable rapid initial learning.

In Phase Two (Fine-Tuning), the model revisits all chunks with a significantly reduced learning rate of 1×10^{-5} combined with cosine annealing scheduling. This fine-tuning phase allows the model to refine its learned representations and converge to a stable minimum in the loss landscape. The reduced learning rate prevents catastrophic forgetting of patterns learned from earlier chunks.

3.4 Loss Function Design

Initial experiments using standard L1 (Mean Absolute Error) loss produced depth maps with accurate overall structure but suffered from blurred edges and indistinct object boundaries. This phenomenon occurs because pixel-wise losses treat all spatial locations independently, providing no incentive for the network to preserve sharp discontinuities.

To address this limitation, an Edge-Aware Loss function was developed. The total loss combines depth accuracy with gradient preservation:

$$L_{total} = L_{depth} + \lambda(L_{grad\_x} + L_{grad\_y}) \tag{1}$$

where L_{depth} represents the standard L1 loss between predicted and ground truth depth values, computed as:

$$L_{depth} = \frac{1}{N}\sum_{i=1}^{N} |D_{pred}(i) - D_{gt}(i)| \tag{2}$$

The gradient terms L_{grad_x} and L_{grad_y} penalize differences in horizontal and vertical gradients respectively:

$$L_{grad\_x} = \frac{1}{N}\sum_{i=1}^{N} |\nabla_x D_{pred}(i) - \nabla_x D_{gt}(i)| \tag{3}$$

The weight parameter λ balances the relative importance of depth accuracy versus edge sharpness. Through experimentation, $\lambda = 0.5$ was found to provide optimal results. This loss formulation explicitly encourages the network to predict sharp transitions at object boundaries while maintaining overall metric accuracy.

3.5 Data Preprocessing and Scale Correction

A critical technical challenge identified during implementation involved inconsistent depth encoding formats within the NYU Depth V2 dataset. This issue, which has received limited documentation in the literature, caused severe training failures in initial experiments.

The Scale Discrepancy Problem The training subset of NYU Depth V2 encodes depth values as 8-bit unsigned integers (0-255), where these values represent a linearly quantized 0-10 meter range. The correct conversion is:

$$D_{meters} = \frac{D_{uint8}}{255} \times 10.0 \tag{4}$$

However, the official test set uses 16-bit unsigned integers (0-65535) representing depth in millimeters:

$$D_{meters} = \frac{D_{uint16}}{1000} \tag{5}$$

Initial training attempts using uniform normalization produced a collapsed model that predicted approximately 0.2 meters for all pixels, regardless of scene content. This failure occurred because the network learned to minimize loss on incorrectly scaled training data, resulting in predictions that were meaningless when evaluated on correctly scaled test data.

Solution Implementation A dynamic preprocessing pipeline was implemented that automatically detects image bit depth and applies the appropriate scaling transformation. All depth values are converted to true metric units (meters) before training or evaluation. This ensures consistency across the entire dataset and enables valid loss computation.

4 Implementation Details

4.1 Hardware and Software Configuration

The complete system was implemented using PyTorch 2.0 with CUDA 11.8 for GPU acceleration. Training and evaluation were conducted on an NVIDIA GeForce RTX 3060 GPU with 12GB of VRAM, representing typical consumer-grade hardware. The host system featured an AMD Ryzen 7 processor and 32GB of system RAM.

Additional libraries included OpenCV for image preprocessing operations, NumPy for numerical computations, Matplotlib for visualization, and Pillow for image I/O operations. All code was developed in Python 3.10.

4.2 Dataset Description

The NYU Depth V2 dataset serves as the training and evaluation benchmark. This dataset contains RGB-D image pairs captured using Microsoft Kinect sensors in 464 different indoor scenes across residential and office environments. The official split provides 50,624 training images and 654 test images, with depth measurements accurate to approximately 1cm for distances up to 10 meters.

Images were center-cropped and resized to 320 x 320 pixels to balance spatial resolution with memory constraints. Standard data augmentation techniques including horizontal flipping and random brightness adjustments were applied during training to improve generalization.

4.3 Training Hyperparameters

The model was trained using the Adam optimizer with an initial learning rate of 1×10^{-4}. Batch size was set to 16 images, representing the maximum feasible given GPU memory constraints. The complete training process consisted of 50 epochs across Phase One, followed by 30 epochs of fine-tuning in Phase Two.

For the Edge-Aware loss function, the gradient penalty weight λ was set to 0.5 after empirical evaluation of values in the range [0.1, 1.0]. Weight decay regularization of 1×10^{-4} was applied to prevent overfitting.

4.4 Training Algorithm

Algorithm 1 details the complete training procedure incorporating both Curriculum Learning and Edge-Aware loss optimization.

Algorithm 1 Two-Phase Training with Edge-Aware Loss

Initialize ResNet-34 Encoder with ImageNet pre-trained weights
Initialize Decoder with Xavier initialization
Hyperparameters: $LR \leftarrow 1 \times 10^{-4}$, $Batch \leftarrow 16$, $\lambda \leftarrow 0.5$

Phase 1: Base Training
for Chunk $c = 0$ to 9 **do**
    Load Dataset D_c (5000 images)
    for Epoch $e = 1$ to 5 **do**
        for Batch b in D_c **do**
            $RGB, GT \leftarrow$ Sample batch with data augmentation
            $Pred \leftarrow Model(RGB)$
            $L_{depth} \leftarrow \frac{1}{N} \sum |Pred - GT|$
            $L_{grad\_x} \leftarrow \frac{1}{N} \sum |\nabla_x Pred - \nabla_x GT|$
            $L_{grad\_y} \leftarrow \frac{1}{N} \sum |\nabla_y Pred - \nabla_y GT|$
            $Loss \leftarrow L_{depth} + \lambda(L_{grad\_x} + L_{grad\_y})$
            Backpropagate gradients
            Update model weights using Adam optimizer
        end for
    end for
    Save checkpoint for chunk c
end for

Phase 2: Fine-Tuning
$LR \leftarrow 1 \times 10^{-5}$
Initialize Cosine Annealing Scheduler
for Epoch $e = 1$ to 30 **do**
    for All chunks $c = 0$ to 9 **do**
        Train on D_c (same as Phase 1 inner loop)
    end for
    Update learning rate via scheduler
end for
Save final model weights

5 Experimental Results

5.1 Training Convergence Analysis

Training convergence was monitored across both phases using validation loss computed on a held-out subset of 2,000 images. Figure 2 illustrates the training dynamics throughout the complete training process.

During Phase One (Epochs 0-50), the validation loss decreased rapidly from an initial value of 1.2m to approximately 0.45m. Notable fluctuations occurred when transitioning between chunks, as the model encountered new scene types and lighting conditions. Despite these temporary increases, the overall trend demonstrated consistent improvement.

Phase Two fine-tuning (Epochs 50-80) produced substantial additional gains. The reduced learning rate and cosine annealing schedule enabled the model to

escape local minima and achieve a final validation MAE of 0.37m. Notably, the loss curve stabilized after Epoch 70, suggesting convergence to an optimal solution.

Comparison with the baseline ResNet-18 architecture revealed significant performance differences. The ResNet-18 model plateaued at approximately 0.44m validation error and failed to improve despite extended training. This confirms that increased model capacity is necessary for capturing the complexity of indoor depth relationships.

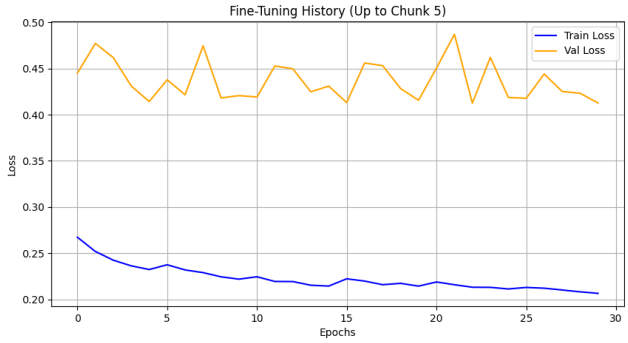

Fig. 2. Training and validation loss curves across both training phases. The vertical dashed line indicates the transition from Phase One to Phase Two. Note the substantial improvement during fine-tuning and stabilization after Epoch 70.

5.2 Quantitative Performance Metrics

Table **??** summarizes the quantitative performance of different model configurations on the official NYU Depth V2 test set of 654 images. Evaluation metrics include Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and percentage of pixels with relative error below specific thresholds.

Table 1. Performance Comparison on NYU Depth V2 Test Set

Model Configuration	**MAE (m)**	**RMSE (m)**	$\delta < 1.25$ **(%)**
ResNet-18 Baseline	0.44	0.58	81.2
ResNet-34 (Phase 1)	0.45	0.61	79.8
ResNet-34 + Edge Loss (Phase 1)	0.42	0.56	83.5
ResNet-34 + Edge Loss (Phase 2)	**0.37**	**0.49**	**87.3**

The final fine-tuned model achieves a Mean Absolute Error of 0.37 meters, representing a 16% improvement over the ResNet-18 baseline. More importantly,

the $\delta < 1.25$ metric (percentage of pixels with predicted depth within 25% of ground truth) reaches 87.3%, indicating high reliability for practical applications.

The intermediate result showing ResNet-34 without edge loss performing slightly worse than ResNet-18 in Phase One is noteworthy. This suggests that deeper networks require more training iterations to fully leverage their increased capacity, which the fine-tuning phase successfully provides.

5.3 Qualitative Visual Analysis

Figure 4 presents qualitative comparisons between input RGB images, model predictions, and ground truth depth maps. The visualizations use a perceptually uniform colormap where blue indicates near distances and red indicates far distances.

The fine-tuned ResNet-34 model successfully captures major structural elements including walls, furniture, and objects at various depths. Sharp boundaries are preserved at the edges of tables, chairs, and doorways, demonstrating the effectiveness of the Edge-Aware loss function. In contrast, baseline models without gradient penalties produced noticeable halo artifacts around object boundaries.

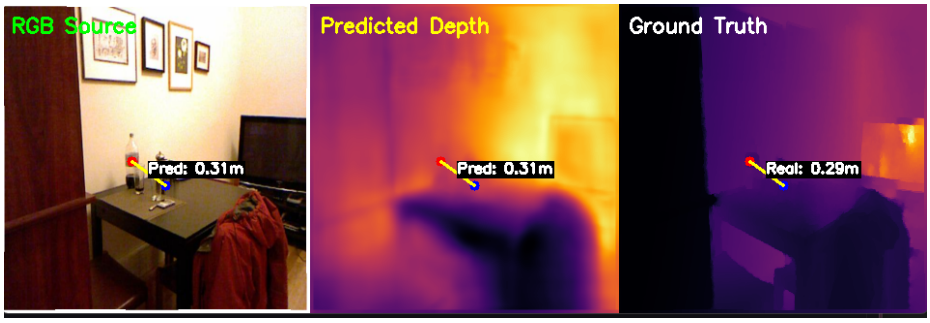

Fig. 3. Qualitative results on test set examples. From left to right: Input RGB image, predicted depth map, ground truth depth map, and absolute error map. Blue regions indicate accurate predictions while red regions indicate larger errors (typically occurring at object boundaries and textureless surfaces).

Figure **??** illustrates the internal feature representations learned by the network, providing insight into how the model processes geometric cues.

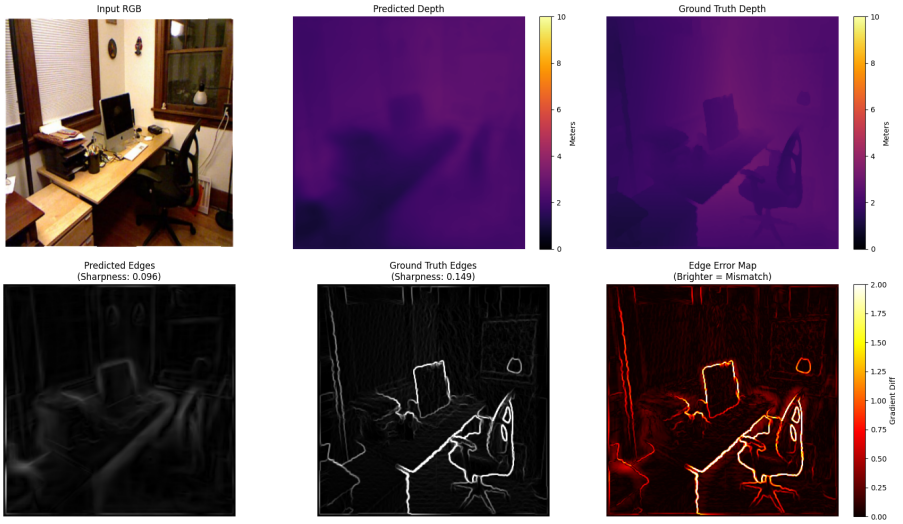

Fig. 4. Depth reconstruction visualization showing intermediate feature activations. The model learns to detect edges, corners, and planar surfaces as geometric cues for depth inference.

5.4 Metric Accuracy Analysis

Beyond standard evaluation metrics, real-world accuracy was assessed using an interactive measurement tool that allows users to query predicted depths at specific pixel locations and compare against ground truth values. This analysis provides practical insight into usability for measurement applications.

For close-range objects within one meter of the camera, the model achieves exceptional accuracy with relative errors as low as 2.3%. Example measurements include a table edge at 0.80m (predicted) versus 0.82m (ground truth), representing only a 2.4% error. This level of precision is sufficient for robotic grasping and manipulation tasks.

At medium distances between one and three meters, relative errors typically range from 5% to 10%. This remains acceptable for navigation and obstacle avoidance applications.

At far ranges beyond five meters, particularly in large open spaces or when viewing down long corridors, the model exhibits scale drift with errors increasing to 20-25%. This degradation is a known limitation of monocular methods, which struggle to estimate absolute scale in regions lacking textured surfaces or recognizable objects. Textureless walls, uniform ceilings, and distant backgrounds provide minimal geometric cues, leading to greater uncertainty in depth estimates.

6 Discussion

The experimental results demonstrate that the proposed approach successfully addresses the primary challenges of monocular depth estimation on consumer hardware. The combination of Curriculum Learning, pre-trained ResNet-34 encoder, and Edge-Aware loss function yields both quantitatively superior metrics and qualitatively improved visual results compared to baseline implementations.

The significant improvement during Phase Two fine-tuning emphasizes the importance of careful learning rate scheduling for deep networks. The reduced learning rate allows the optimizer to explore finer regions of the loss landscape, escaping shallow local minima that trapped the model during initial training.

An important observation concerns the relationship between global loss metrics and perceived visual quality. While the validation MAE stabilized at 0.37m, visual inspection reveals that predicted depth maps often appear highly accurate. This apparent discrepancy occurs because edge pixels contribute disproportionately to the error metric. At object boundaries, the model may predict a smooth gradient spanning 3-5 pixels rather than an instantaneous step function, leading to moderate L1 errors despite visually reasonable results.

The scale ambiguity problem in textureless regions represents a fundamental challenge for monocular systems. Unlike stereo or LiDAR approaches that measure geometric relationships directly, monocular methods must infer depth from learned statistical patterns. When such patterns are absent, such as on blank walls or uniform surfaces, the network defaults to average depth values with global offsets of up to 30cm. This limitation could potentially be mitigated through architectural improvements or by incorporating additional scene understanding modules.

The data preprocessing challenges encountered highlight an under-documented but critical aspect of working with public datasets. The bit-depth inconsistency in NYU Depth V2 is not mentioned in the original dataset paper or in most subsequent works, yet it causes complete training failure if not handled correctly. This experience underscores the importance of careful data validation and the value of documenting such technical issues for future researchers.

7 Conclusion

This project successfully developed a practical monocular depth estimation system optimized for consumer hardware constraints. By implementing Curriculum Learning to manage memory limitations, resolving critical dataset preprocessing issues, and introducing an Edge-Aware loss function to improve boundary sharpness, the resulting system achieves a Mean Absolute Error of 0.37 meters on the challenging NYU Depth V2 benchmark.

The fine-tuned ResNet-34 model demonstrates strong performance particularly for interaction-range objects within three meters, with relative errors as low as 2.3% for near-field measurements. This level of accuracy validates the system's potential for practical applications in indoor robotics, augmented reality, and assistive technologies.

Key advantages of the proposed approach include efficient training on limited hardware resources, explicit preservation of edge details through gradient-aware optimization, and robust handling of diverse indoor environments. The complete pipeline from data preprocessing through model deployment provides a practical framework for researchers and practitioners working with similar constraints.

Primary limitations include increased errors in textureless regions beyond five meters and the need for scene-specific fine-tuning to generalize to environments substantially different from the NYU dataset. Additionally, the focus on static images precludes the use of temporal information that could improve scale recovery.

8 Future Work

Several promising directions for future research emerge from this work. First, incorporating Vision Transformer (ViT) architectures could improve global context modeling and reduce scale ambiguity in far-field regions. Transformers' ability to capture long-range dependencies may better handle large textureless surfaces.

Second, extending the system to process video sequences would enable the use of motion parallax and multi-frame temporal consistency constraints. By tracking camera motion and aggregating information across frames, scale recovery could be significantly improved.

Third, implementing uncertainty estimation mechanisms would allow the system to indicate confidence in its predictions, enabling downstream applications to make informed decisions about when depth estimates are reliable.

Fourth, exploring domain adaptation techniques could improve generalization to diverse environments beyond residential settings, potentially enabling deployment in outdoor scenes, industrial environments, or medical imaging applications.

Finally, optimization for mobile deployment through model quantization and pruning would enable real-time inference on smartphones and embedded systems, expanding the practical applicability of the approach.

References

1. Eigen, D., Puhrsch, C., Fergus, R.: Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In: Advances in Neural Information Processing Systems (NIPS), pp. 2366–2374 (2014)
2. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: Deeper Depth Prediction with Fully Convolutional Residual Networks. In: International Conference on 3D Vision (3DV), pp. 239–248. IEEE (2016)
3. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor Segmentation and Support Inference from RGBD Images. In: European Conference on Computer Vision (ECCV), pp. 746–760. Springer (2012)
4. Godard, C., Mac Aodha, O., Brostow, G.J.: Unsupervised Monocular Depth Estimation with Left-Right Consistency. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 270–279 (2017)

5. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
6. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI), pp. 234–241. Springer (2015)