Signal, Image and Video Processing (2025) 19:876
https://doi.org/10.1007/s11760-025-04496-8

ORIGINAL PAPER

Multi-Source Depth Estimation: Utilizing Real, Synthetic, and Monocular Depth Data with Custom Loss Functions

Muhammad Adeel Hafeez[1] · Ganesh Sistu[2] · Michael G. Madden[1] · Ihsan Ullah[1]

Received: 26 June 2025 / Revised: 29 June 2025 / Accepted: 1 July 2025 / Published online: 15 July 2025

Abstract

Depth estimation from 2D images is an essential task in computer vision with applications in scene understanding, robotics, and autonomous systems. The performance of supervised depth models depends on network design, loss formulation, data quality, and fine-tuning strategy. In this study, we propose a progressive fine-tuning approach for metric (absolute-scale) depth estimation. Our method uses transfer learning across multiple indoor datasets: real, synthetic, and pseudo-labelled. DenseNet-169 and EfficientNet-B0 backbones are fine-tuned on MIT-G, SUN-RGBD, SceneNet, and NYU2. We apply a three-scale combined loss with weighted MAE + Edge + SSIM terms at full, 1/2, and 1/4 resolution, and add a perceptual VGG component, while we keep the global coefficients of the loss at 1 for simplicity and reproducibility. We find that EfficientNet performs better on the smaller datasets, while DenseNet benefits most from the million-image SceneNet stage and reaches REL 0.105 and RMSE 0.359 on NYU2, comparable to recent transformer baselines yet using 6× fewer parameters. The pseudo-labelled MIT-G data is used as a warm-start and shows the potential of reducing annotation cost. All headline metrics results are based on sensor ground-truth data, avoiding circular evaluation. Qualitative analysis and zero-shot tests on the unseen iBims-1 benchmark confirm that the models generalise and produce coherent, detailed depth maps across diverse indoor scenes. The proposed pipeline thus offers a balanced trade-off between accuracy and computational cost for practical indoor depth estimation.

Keywords Depth Estimation · Transfer Learning · Monocular Depth Estimation · Progressive Fine-Tuning

1 Introduction

Depth estimation is an important computer-vision task that provides information about the 3D geometry of a scene, with applications in robotics [1], autonomous driving [2], augmented reality [3], and more. Traditional approaches rely on specialised sensors such as ultrasonic sensors, radar, or LiDAR. Because these sensors increase system cost, research has shifted toward image-based depth estimation from monocular or stereo cameras.

Early image-based work relied on geometric cues (e.g. vanishing points [4]) or graphical-model formulations such as MRF/CRF [5], but these methods struggled in complex scenes. The recent advancement of deep learning, from convolutional neural networks (CNNs) through recent vision transformers (ViTs), has dramatically improved accuracy and robustness of image based depth estimation [6].

Image-based depth estimation can be divided into binocular depth estimation (BDE) and monocular depth estimation (MDE). In BDE, depth is calculated using the disparity between two 2D images through stereo matching and triangulation [7], but it performs poorly when the texture is weak [8]. MDE, on the other hand, predicts per-pixel depth from a single RGB image [9, 10] hence relies on ground-truth maps obtained with LiDAR, structured light, or photorealistic rendering engines [11]. Most state-of-the-art (SOTA) models adopt an encoder–decoder architecture in which a classifi-

✉ Ihsan Ullah
ihsan.ullah@universityofgalway.ie

Muhammad Adeel Hafeez
m.hafeez1@universityofgalway.ie

Ganesh Sistu
ganesh.sistu@valeo.com

Michael G. Madden
michael.madden@universityofgalway.ie

[1] Machine Learning Research Group, School of Computer Science, University of Galway, Galway H91 TK33, Ireland

[2] Valeo Vision Systems, Valeo, Tuam, Ireland

cation backbone extracts features and a decoder predicts the dense depth.

The performance of an MDE model depends upon several factors: (i) architecture [10]; (ii) loss function [12]; and (iii) the quantity and quality of the training data [13]. Advanced CNNs or ViTs leverage multi-scale features and attention, while carefully weighted combinations of MAE, edge loss, and SSIM [14] improve the perceptual quality of the depth estimates. Large multi-source datasets improve generalisation [15]. Recent transformer-based models (DPT [10], Depth Anything [6]) achieved impressive results but require massive compute budgets and months-long pre-training. This motivates lighter strategies that reuse conventional CNN backbones while using data from multiple sources.

In this work we present a *progressive fine-tuning* (PF) method that incrementally integrates four indoor datasets: NYU2 [16] and SUN-RGBD [17] (real RGB-D), MIT Indoor Scenes with pseudo-depth from Depth Anything [6, 18], and the large synthetic SceneNet RGB-D [19]. Starting with small real sets, the network learns fundamental indoor cues; SceneNet then injects structural diversity; finally, an NYU2 stage refines depth based on the sensor output data. We adopted the DenseDepth [9] and Efficient-Depth [14] frameworks with ImageNet-pre-trained encoders (DenseNet-169/EfficientNet-B0). Following [14] we keep the optimised weights for MAE, edge and SSIM, apply them in a three-scale manner, and add a perceptual term, yielding a combined loss that balances accuracy and efficiency without further tuning. From SceneNet, we sample 1 M frames (1/5 of the full set), retaining scene diversity while reducing computational cost.

Our experiments show that progressive finetuning yields consistent gains across all datasets: relative error (REL) on NYU2 drops from 0.142 (single-dataset baseline) to 0.105, and root-mean-square error (RMSE) from 0.471 to 0.359, competitive with heavier transformer models while running on a single GPU. Zero-shot tests on iBims-1 further demonstrate improved generalisation. Although inspired by transfer learning [9] and mixed-dataset training [13], our contribution is empirical: we offer a lightweight model that turns existing CNN encoders into strong indoor depth predictors without industrial-scale pre-training.

The rest of the paper is organised as follows. Section 2 reviews recent literature. Section 3 details the progressive fine-tuning, architecture, and loss functions. Section 4 reports ablations and SOTA comparisons, and Section 5 concludes the paper.

2 Related Work

Depth estimation has significantly advanced over the years. Early image-based techniques relied on handcrafted features and depth cues [5] but struggled with low-texture scenes [20]. Modern deep-learning approaches address these challenges, though they rely on large data volumes [6]; accuracy depends on both loss design [21] and architecture [22]. One of the earliest monocular CNN models was proposed by Eigen et al. [23]. Successive work explored deeper CNNs [9], pre-training [13], novel losses such as ordinal regression [24] and gradient consistency [25], and larger datasets [26]. GAN-based refinement further improved details in the predicted depth [27]. The introduction of vision transformers [28] enabled global reasoning; DPT [10] remains a strong baseline. Very recent transformer variants such as UniDepth [29] and BinsFormer [30] push accuracy even higher but at heavy computational cost. Diffusion models [31] provide another promising direction.

Besides these architecture updates, data strategy is also very important for MDE design. Synthetic datasets produced in simulation offer unlimited, perfectly annotated images [32, 33], whereas real-world RGB-D data reflect practical noise [9]. Combining the two can improve generalisation [34] but also introduces domain gaps [35]. Domain-adaptation methods have mixed success, yet hybrid data pipelines are increasingly popular now a days [36]. Training the model on larger and multiple datasets for depth estimation was proposed by MiDaS [13], and the large-scale Depth-Anything pre-training [6] shows that exposure to large data can yield robust predictors. Finally, the carbon and hardware demands of such large models are a growing concern [37]. Our study therefore focuses on progressive multi-dataset fine-tuning with mid-sized CNN backbones, aiming to capture these benefits while remaining computationally practical.

3 METHODOLOGY

In this section, we will discuss the details of the datasets, including their selection and generation. After that, we will discuss the model architectures and loss function. The section concludes with a description of the performance evaluation metrics.

3.1 Datasets

In this study, we have used datasets from multiple sources and applied them to perform progressive fine-tuning of the model. The datasets are summarized below in the table 1.

NYU Depth V2 [16] and SUN-RGBD [17] provide high-quality Kinect RGB-D pairs across typical indoor scenes. MIT-G is based on the MIT Indoor Scenes data [18] with pseudo-depth from Depth Anything [6], offering additional supervision for a warm start (model weights are initialised from a short training on MIT-G pseudo-labels before being further fine-tuned on sensor-labelled datasets). SceneNet

Table 1 Datasets used in progressive fine-tuning (ordered by stage)

Stage	Dataset	Pairs	Resolution	Depth m	Sensor / Source	Purpose
1	MIT-G (pseudo)	6.4 k	$\geq 640\times480$	0.5–10	RGB only + DepthAnything	warm start on pseudo labels
2	SUN-RGBD	10 k	640×480 / 730×530	0.5–10	Kinect v1,v2	small-scale real refinement
3	SceneNet RGB-D	1 M	640×480	0.5–10	synthetic renderer	broaden scene diversity
4	NYU Depth V2	120 k	640×480	0.5–10	Kinect v1	high-quality fine-tune

RGB-D [19] supplies photorealistic synthetic images spanning 13 room types, where we sample 1 M frames to reduce computational cost while preserving variety. Dataset artifacts, such as missing depths values on windows/shining surfaces and light source, are analysed in Supplement A. Supplement B presents a visual overview of the four datasets, illustrating representative RGB–depth pairs from NYU2, SUN-RGBD, MIT-G, and SceneNet.

3.2 Network Architecture

In this study, we have utilized several encoder–decoder-based models to perform depth estimation on the datasets mentioned in the previous subsection. We have progressively fine-tuned these models and, in this paper, we report the intermediate and final results. The main goal of fine-tuning these models was to capture both the overall structure and fine details in the depth maps for all datasets, leading to more accurate and visually consistent predictions.

In a previous study [14], the authors explored four encoder architectures including DenseNet121, DenseNet169, DenseNet201 and EfficientNet-B0, all pre-trained on ImageNet for classification purposes. Because DenseNet169 and EfficientNet-B0 delivered the best accuracy in that work, we adopt only these two for progressive fine-tuning here. ResNet-50 was also tested as an even lighter option but its results were not comparable.

The encoders transform the input images into rich feature vectors, which are then passed through a series of up-sampling layers in the decoder. The decoder, incorporating the skip connections, reconstructs the depth maps from these feature vectors. We initially used a simplified decoder structure across all experiments, avoiding batch normalisation or other advanced layers, following the findings of [24]. In the final model we introduce a single normalisation layer after each 3×3 convolution; this stabilises training under varying indoor lighting, as also noted in [10].

With this configuration the DenseDepth branch (DenseNet-169 encoder) contains 21.5 M parameters and 98 GFLOPs per 640×480 image, whereas the EfficientDepth branch (EfficientNet-B0 encoder) uses 23.6 M parameters and 146 GFLOPs; full speed-versus-accuracy comparisons with transformer baselines are reported in Table 3.

Figure 1 provides a schematic representation of the generic architecture employed in our study. While we experimented with various state-of-the-art models as encoders, the decoder design was kept consistent and straightforward to ensure fair comparisons across the different setups.

3.3 Loss Function

The loss function guides the network during training, and its design strongly influences depth-estimation accuracy [38]. As in our earlier work [14] we started from a *weighted loss* that mixes Mean Absolute Error (MAE), Edge loss, and SSIM with fixed coefficients {0.6, 0.2, 1.0}; full formulas and intuition for these three pixel-level terms are reproduced in Supplement B. Here we focus on the two extensions, *perceptual loss* and the *multi-scale strategy*, that are different in the present study.

3.3.1 Perceptual Loss

Perceptual loss, also known as feature-based loss, is an important component in deep learning tasks that require high-level feature matching between predicted and ground truth images. In the depth estimation, perceptual loss helps the model focus not only on getting the absolute depth values correct but also on preserving the overall structure and important details of the scene [39]. To calculate the perceptual loss, we used the first three layers of a pre-trained VGG16 on ImageNet to extract features.

The VGG16 model (or the feature-extraction model) is applied to both the true and predicted depth maps, producing feature maps which were compared to calculate the mean squared loss. By minimizing this loss, the model is encouraged to produce depth predictions that align well with the high-level visual features of the true depth map, resulting in more accurate and visually better outputs. The formula to calculate the perceptual loss is:

$$\mathcal{L}_{\text{perceptual}} = \sum_{i=1}^{N} \text{mean}\left(\left(\phi_i(Y_t) - \phi_i(Y_p)\right)^2\right) \tag{1}$$

where ϕ_i represents the feature map extracted from the i-th layer of the VGG16 model and N is total number of lay-

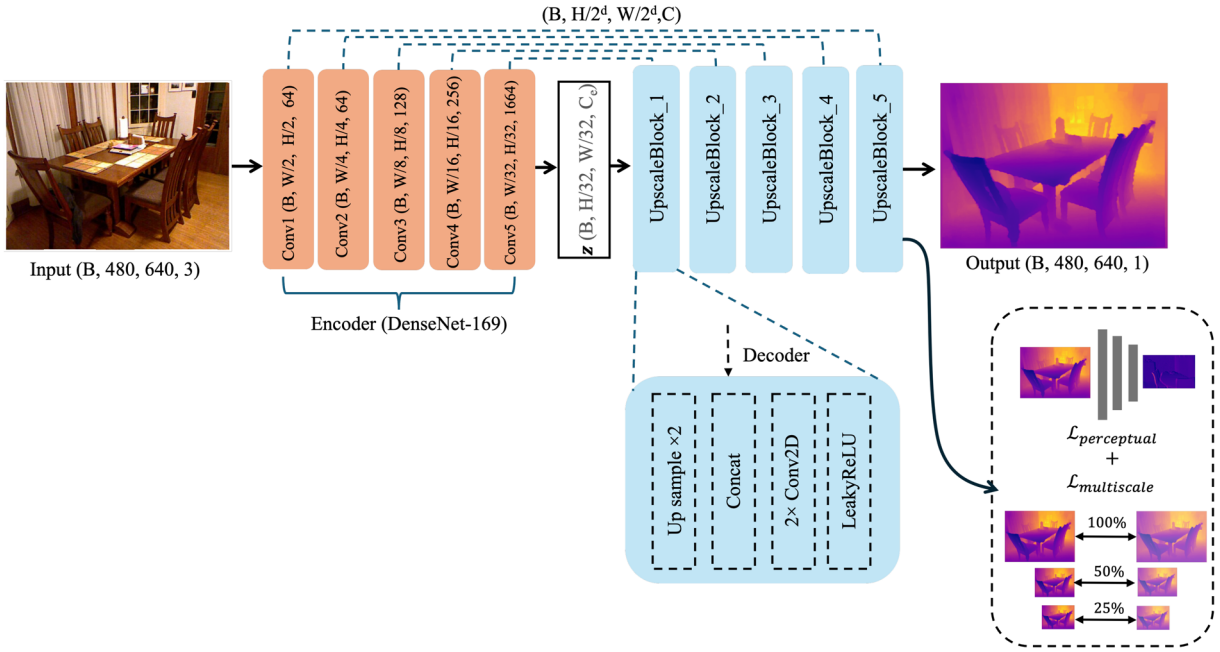

Fig. 1 Encoder–decoder architecture (illustrated with DenseNet-169). Orange blocks are the encoder stages with tensor shapes $(B, H/2^k, W/2^k, C_k)$; dashed arrows show skip connections merged in the corresponding blue *UpscaleBlocks*. Replacing the encoder stack with ResNet-50 or EfficientNet-B0 yields the other model variants evaluated in this work

ers used to extract features. In this experiment, the value of N is three as we use first three layers of VGG for feature extraction.

3.3.2 Combined Loss

To make an overall loss, we combined the individual losses and proposed a combined loss. The combined loss we proposed is given below:

$$\mathcal{L}_{\text{combined}} = \mathcal{L}_{perceptual} + \mathcal{L}_{depth} \tag{2}$$

where $\mathcal{L}_{depth}$ is a weighted combination of MAE, SSIM and Edge loss as proposed in [14].

$$\mathcal{L}_{depth} = 0.6\mathcal{L}_{MAE} + 0.2\mathcal{L}_{edge} + \mathcal{L}_{SSIM} \tag{3}$$

To further enhance the efficiency, we used a multi-scale combined loss function for $\mathcal{L}_{depth}$ that calculates losses at multiple resolutions (scales).

This approach helps the model learn consistent depth predictions across various scales, ensuring that larger structures and fine-grained details are both accurately represented. By calculating loss at different scales, the model is better equipped to handle the inherent challenges in depth estimation, such as varying object sizes, complex textures, and depth discontinuities. The multi-scale approach also encourages the model to focus on different aspects of the scene at each scale, contributing to a more robust and generalizable depth prediction. We calculated this loss for three different resolutions:

$$\mathcal{L}_{\text{combined}} = \mathcal{L}_{perceptual} + \mathcal{L}_{depth\{1,0.5,0.25\}} \tag{4}$$

Here we kept the coefficients of these loss components to 1 to balance feature-level and pixel-level terms without adding a new hyper-parameter search and tuning these coefficients is left for future work.

3.4 Performance Evaluation

To evaluate the output of our models we have used both quantitative and qualitative approaches. In the qualitative approach, we plotted the depth maps in 'plasma' colour scheme where the darker pixels show the objects near to the camera and brighter pixels show the objects far from the camera. Quantitatively we report the six standard metrics widely used in recent work [6, 30]: threshold accuracy ($\delta_1, \delta_2, \delta_3$), relative error (REL), root-mean-squared error (RMSE), and $\log_{10}$ error. All scores are computed after Eigen cropping

4 Experiments and Results

In this section, we will describe the implementation details and results for all the experiments.

4.1 Implementation

We implemented our proposed models using TensorFlow and trained them on two different machines. For most of the training, including fine-tuning on the MIT-G, SUN-RGBD, and NYU2 datasets, we used an NVIDIA GeForce 2080 Ti with 4,352 CUDA cores and 11 GB of GDDR6 memory. To fine-tune our model on the SceneNet dataset, we utilized an NVIDIA RTX 6000 Ada Generation GPU with approximately 48 GB (49,140 MiB) of memory. The training time varied between the machines, model and datasets (for example, DenseNet169 training on NYU2 for 50 epochs took 30 hours on GeForece 2080 Ti and DenseNet169 training on SceneNet 1 million images took 100 hours for 50 epochs on NVIDIA RTX 6000). Building on the previous setup, we trained an encoder-decoder-based model. In all experiments, we set the initial learning rate to 0.0001, following [9, 13]. Empirically, higher rates caused unstable gradients, while lower rates slowed convergence without accuracy gain. We used the Adam optimiser ($\beta_1 = 0.9, \beta_2 = 0.99$) for its adaptive moment estimates, which stabilise encoder–decoder training. The batch size was 4 on the 2080Ti and 8 on the RTX6000, while data augmentation comprised random horizontal flip, brightness/contrast jitter ($\pm 15\%$).

4.2 Experiment 1: Encoder Architecture

In this experiment, we evaluated the effect of three commonly used encoder backbones: ResNet50, DenseNet169, and EfficientNetB0, on the task of depth estimation using three indoor datasets (NYU2, SUN RGBD, and MIT-G). While ResNet50 provided better computational efficiency, both DenseNet169 and EfficientNetB0 showed substantial accuracy improvements. The decoder architecture was kept the same across all experiments, using upsampling and skip connections to reconstruct the depth map. To ensure consistency across datasets, we used Eigen cropping during evaluation. A detailed analysis of encoder configurations, performance metrics, and dataset-specific results is included in the Supplementary Material (Section D).

4.3 Experiment 2: Loss Function

In this experiment, we compared several loss functions using DenseNet169 and EfficientNetB0 encoders. Building on the baseline weighted loss, we evaluated the addition of perceptual and multi-scale components. While perceptual loss did not yield notable numerical improvements, introducing a multi-scale formulation showed marginal gains. These improvements became more evident when models were trained using a progressive fine-tuning strategy. A detailed breakdown of loss configurations, experimental settings, and quantitative results is presented in the Supplementary Material (Section E).

4.4 Experiment 3: Multiple Datasets

In our final experiment, we applied progressive fine-tuning to our models across multiple datasets using the multiscale-combined loss function. To start, we initialized the encoder with weights pre-trained on ImageNet, while the decoder weights were initialized randomly. We progressively fine-tuned our model on each dataset in sequence: we began with the pseudo-labeled MIT-G dataset and the smaller real SUN-RGBD data, then scaled up to the large synthetic SceneNet dataset, and finally refined the model on the high-quality NYU2 dataset. This order is important as it allows the model to adapt to simpler or smaller indoor data first, gain broader coverage from a large synthetic dataset next, and ultimately achieve finer accuracy by incorporating Kinect-based ground truth. Algorithm 1 outlines our progressive fine-tuning strategy, where we sequentially train the network on each dataset.

For evaluation, we tested the model on the test set provided by the NYU2 dataset. The step-by-step fine-tuning approach allowed the model to progressively learn and adapt to the specific characteristics of each dataset, ultimately enhancing its overall performance. Table 2 shows the results of the progressive fine-tuning where we trained two different models and fine-tuned them with a series of datasets. The final results reveal that EfficientNet initially outperformed DenseNet169 when fine-tuned on smaller datasets like MIT-G and SUN-RGBD, achieving slightly better accuracy across several metrics, including higher values for the δ thresholds and lower error rates in certain cases. However, when the models were progressively fine-tuned using the larger SceneNet dataset, DenseNet169 demonstrated a significant improvement, surpassing EfficientNet in overall performance. Specifically, DenseNet169 achieved a better balance of accuracy and error reduction, with a notable improvement in REL (0.105) and RMSE (0.359) compared to EfficientNet's performance on the larger dataset (REL 0.116, RMSE 0.390). This suggests that while EfficientNet adapted well to smaller datasets, DenseNet's architecture leveraged the larger SceneNet dataset more effectively. As a result, DenseNet achieved better depth-estimation performance across various metrics after fine-tuning. To assess whether catastrophic forgetting occurred, we re-evaluated the model on the previously used datasets after each new fine-tuning step. Specifically, once the model was fine-tuned on the fully synthetic SceneNet, we checked its performance on MIT-G and SUN-RGBD. We observed a slight drop in accuracy at that stage, but after the final fine-tuning on NYU2, the performance on earlier datasets improved again. Overall, these changes were small, indicating that catastrophic forgetting was not observed in our multi-dataset fine-tuning

Table 2 Performance of Progressive Fine-Tuning

Model	Fine-tuning Sequence	$\delta_1 \uparrow$	$\delta_2 \uparrow$	$\delta_3 \uparrow$	REL↓	RMSE↓	$\text{Log}_{10} \downarrow$
DenseNet169	MIT-G, NYU2	0.853	0.954	0.995	0.142	0.4712	0.062
	MIT-G, SUN-RGBD, NYU2	0.861	0.961	0.995	0.129	0.4215	0.053
	MIT-G, SUN-RGBD, SceneNet, NYU2	**0.897**	0.972	0.994	**0.105**	**0.358**	**0.045**
EfficientNet	MIT-G, NYU2	0.8368	0.9680	0.991	0.131	0.4172	0.055
	MIT-G, SUN-RGBD, NYU2	0.882	**0.979**	**0.996**	0.172	0.4015	0.051
	MIT-G, SUN-RGBD, SceneNet, NYU2	0.880	0.9528	0.995	0.114	0.3893	0.049

approach. It is worth noting that MIT-G serves only as an inexpensive warm-start, and final-reported metrics were computed on sensor-based test sets, so no circular evaluation arises from the use of Depth-Anything pseudo-labels.

Algorithm 1 Progressive Fine-Tuning Strategy

Require: Pre-trained encoder E (e.g., DenseNet169/EfficientNetB0) with ImageNet weights;
Datasets $\{D_1, D_2, \ldots, D_k\}$ (where $k = 4$ in this work);
Learning rate $\eta = 1 \times 10^{-4}$;
Combined loss function $\mathcal{L}_{\text{combined}}$.
Ensure: Fine-tuned depth estimation model M.
1: Initialize decoder weights *randomly*.
2: $M \leftarrow \text{Combine}(E, \text{decoder})$
Attach the newly created decoder on top of E.
3: **for** $i \leftarrow 1$ to k **do**
4: **Train**$(M,\ D_i,\ \mathcal{L}_{\text{combined}},\ \eta)$
Continue training M on D_i; both encoder and decoder update via Adam.
5: **Evaluate**$(M,\ \{D_j \mid j < i\})$
Check earlier datasets to monitor performance.
6: **end for**
7: **return** M

Sample predictions from the final models (DenseNet169 and EfficientNetB0) on all four datasets are provided in the Supplementary Material (Section F). These results highlight the models' ability to handle common visual challenges, such as missing depth values, reflective surfaces, and scene complexity. The qualitative evaluation complements the quantitative findings and supports the observed strengths of each encoder. Overall, both models demonstrated strong performance, but DenseNet's ability to preserve small depth changes across different indoor environments stood out. Compared to a baseline [9] using a single dataset (RMSE 0.465), our progressive fine-tuning method reduces the RMSE to 0.358, a 23% improvement. Although our final results do not surpass all SOTA benchmarks, they were achieved using a more limited training set and computational budget. Moreover, our approach generalises well to unseen indoor environments, as shown by our zero-shot tests on iBims-1 in Section 4.5. Since we predict absolute depth rather than relative scales, even these improvements can make a practical difference in applications such as robotics navigation or AR scene understanding.

Table 3 shows a comparison with different SOTA where Depth-Anything-L achieves the highest accuracy on all four metrics, but does so with 335M parameters, 624 GFLOPs, and a 375ms latency. On the other hand, our DenseDepth-PF model reaches a competitive $\delta_1 = 0.897$ and REL 0.105 while using 16× fewer parameters and running in 36.7ms. This demonstrates that the proposed progressive fine-tuning can deliver strong indoor depth performance without the computational overhead of recent transformer-based models. Our EfficientDepth-PF offers a similar trade-off, whereas older CNN baselines lag in both accuracy and efficiency. Further analysis of training stability, overfitting prevention strategies, and loss convergence behavior is provided in Supplementary Section E.1.

4.5 Zero-shot testing

To test the generalization of our final model, which was trained on multiple indoor datasets, we used the iBims-1 dataset [40] for zero-shot testing which was not seen during training. This dataset contains 100 RGB-D image pairs of various indoor scenes, making it a suitable choice for zero-shot testing. Due to differences in sensors, resolution, and depth ranges in the iBims-1 dataset, we selected the scale-invariant log error as a performance metric, as proposed in [41] and used in various studies [29, 42]. The SILog error was calculated using the formula $100\sqrt{\text{Var}(\varepsilon_{\log})}$, with an average value of 50.34 when no cropping was applied and missing depth values (windows or reflective surfaces) were left unchanged. The score was improved to 48.26 when the missing depth values were replaced with nearby pixel values. Sample predictions and visual comparisons are provided in the Supplementary Material (Section F). The results illustrate that both DenseNet169 and EfficientNetB0 retained strong performance in unfamiliar environments, though minor over-smoothing was observed around complex structural edges.

Table 3 Comparison with state-of-the-art monocular depth models on NYU2 (Eigen test split). Inference latency is measured on an Apple M2 Pro MacBook Pro (10-core CPU, 16-core GPU). Best value in each column is shown in bold

Model	**Parameters (M)**	**GFLOPs**	$\delta_1 \uparrow$	REL $\downarrow$	RMSE $\downarrow$	$\log_{10} \downarrow$	Time (ms)
DenseDepth-PF (ours)	**21.5**	**98**	0.897	0.105	0.359	0.045	**36.7**
EfficientDepth-PF (ours)	23.6	146	0.880	0.114	0.389	0.049	47.9
DPT-Hybrid [10]	123	220	0.904	0.110	0.357	0.045	98.4
Depth Anything-L [6]	335	624	**0.984**	**0.056**	**0.206**	**0.024**	374.7
UniDepth [29]	648	1113	0.972	0.062	0.232	–	696
Alhashim. et, al. [9]	42.6	138	0.846	0.123	0.465	0.053	51
Paul. et, al. [38]	48	147	0.845	0.123	0.524	0.053	57

5 Conclusions

The progressive fine-tuning approach that we have introduced in this paper demonstrates that the choice of encoder and dataset size play an important role in achieving accurate depth estimation. While EfficientNet outperforms DenseNet169 on smaller datasets such as MIT-G and SUN-RGBD, DenseNet's performance is significantly improved when trained on the larger SceneNet dataset. The use of a multiscale loss function enables both models to capture finer depth transitions, with DenseNet169 producing more reliable results on complex indoor scenes. Overall, our results suggest that DenseNet169 is better suited for large-scale datasets, making it a strong candidate for depth estimation tasks in real-world applications. Zero-shot results on iBims-1 further show that the final network generalises beyond the datasets seen during fine-tuning and predicts metric depth with consistent structure. Future work will explore tuning the global loss coefficients, adding lightweight attention modules, and extending progressive fine-tuning to outdoor scenes and mixed-sensor benchmarks.

Acknowledgements This publication has emanated from research conducted with the financial support of Taighde Éireann - Research Ireland under Grant No. 18/CRT/6223 and SFI/12/RC/2289_P2 the Insight Research Ireland Centre for Data Analytics. It is in partnership with VALEO.

Author Contributions M.A.H. designed the experiments, conducted data analysis, and drafted the initial manuscript. G.S. provided industry insights and supported the practical application perspective. M.G.M. and I.U. supervised the research, provided guidance on methodology, reviewed results, and critically revised the manuscript. All authors reviewed and approved the final manuscript.

Funding Open Access funding provided by the IReL Consortium. This publication has emanated from research conducted with the financial support of Taighde Éireann – Research Ireland under Grant No. 18/CRT/6223 and SFI/12/RC/2289_P2 the Insight Research Ireland Centre for Data Analytics. It is in partnership with VALEO.

Data Availability No datasets were generated or analysed during the current study.

Competing interests The authors declare no competing interests.

References

1. Dong, X., Garratt, M.A., Anavatti, S.G., Abbass, H.A.: Towards real-time monocular depth estimation for robotics: A survey. IEEE Trans. Intell. Transp. Syst. **23**(10), 16940–16961 (2022)
2. Xue, F., Zhuo, G., Huang, Z., Fu, W., Wu, Z., Ang, M.H.: Toward hierarchical self-supervised monocular absolute depth estimation for autonomous driving applications. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2330–2337 (2020)
3. Diaz, C., Walker, M., Szafir, D.A., Szafir, D.: Designing for depth perceptions in augmented reality. In: 2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 111–122 (2017)
4. Tsai, Y.-M., Chang, Y.-L., Chen, L.-G.: Block-based vanishing line and vanishing point detection for 3d scene reconstruction. In: 2006 International Symposium on Intelligent Signal Processing and Communications, pp. 586–589 (2005)
5. Liu, B., Gould, S., Koller, D.: Single image depth estimation from predicted semantic labels. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1253–1260 (2010)
6. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10371–10381 (2024)
7. Žbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. J. Mach. Learn. Res. **17**(65), 1–32 (2016)

8. Liu, F., Zhou, S., Wang, Y., Hou, G., Sun, Z., Tan, T.: Binocular light-field: Imaging theory and occlusion-robust depth perception application. IEEE Trans. Image Process. **29**, 1628–1640 (2019)
9. Alhashim, I., Wonka, P.: High quality monocular depth estimation via transfer learning. arXiv preprint arXiv:1812.11941 (2018)
10. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12179–12188 (2021)
11. Richter, S.R., Hayder, Z., Koltun, V.: Playing for benchmarks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2213–2222 (2017)
12. Lu, Z., Chen, Y.: Joint self-supervised depth and optical flow estimation towards dynamic objects. Neural Process. Lett. **55**(8), 10235–10249 (2023)
13. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE Trans. Pattern Anal. Mach. Intell. **44**(3), 1623–1637 (2020)
14. Hafeez, M.A., Madden, M.G., Sistu, G., Ullah, I.: Depth estimation using weighted-loss and transfer learning. Proceedings Copyright **780**, 787
15. Bailly, A., Blanc, C., Francis, É., Guillotin, T., Jamal, F., Wakim, B., Roy, P.: Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models. Comput. Methods Programs Biomed. **213**, 106504 (2022)
16. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12, pp. 746–760 (2012). Springer
17. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 567–576 (2015)
18. Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 413–420 (2009)
19. McCormac, J., Handa, A., Leutenegger, S., Davison, A.J.: Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2678–2687 (2017)
20. Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo-stereo matching with slanted support windows. In: Bmvc, vol. 11, pp. 1–11 (2011)
21. Carvalho, M., Le Saux, B., Trouvé-Peloux, P., Almansa, A., Champagnat, F.: On regression losses for deep depth estimation. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 2915–2919 (2018)
22. Ming, Y., Meng, X., Fan, C., Yu, H.: Deep learning for monocular depth estimation: A review. Neurocomputing **438**, 14–33 (2021)
23. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2650–2658 (2015)
24. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2002–2011 (2018)
25. Shim, D., Kim, H.J.: Learning a geometric representation for data-efficient depth estimation via gradient field and contrastive loss. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 13634–13640 (2021)
26. Cabon, Y., Murray, N., Humenberger, M.: Virtual kitti 2. arXiv preprint arXiv:2001.10773 (2020)
27. Hendra, A., Kanazawa, Y.: Tp-gan: Simple adversarial network with additional player for dense depth image estimation. IEEE Access **11**, 44176–44191 (2023)
28. Dosovitskiy, A.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
29. Piccinelli, L., Yang, Y.-H., Sakaridis, C., Segu, M., Li, S., Van Gool, L., Yu, F.: Unidepth: Universal monocular metric depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10106–10116 (2024)
30. Li, Z., Wang, X., Liu, X., Jiang, J.: Binsformer: Revisiting adaptive bins for monocular depth estimation. IEEE Transactions on Image Processing (2024)
31. Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R.C., Schindler, K.: Repurposing diffusion-based image generators for monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9492–9502 (2024)
32. Schreiber, A.M., Hong, M., Rozenblit, J.W.: Monocular depth estimation using synthetic data for an augmented reality training system in laparoscopic surgery. In: 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2121–2126 (2021)
33. Ullah, I., Abinesh, S., Smyth, D.L., Karimi, N.B., Drury, B., Glavin, F.G., Madden, M.G.: A virtual testbed for critical incident investigation with autonomous remote aerial vehicle surveying, artificial intelligence, and decision support. In: ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings 18, pp. 216–221 (2019). Springer
34. Tonioni, A., Rahnama, O., Joy, T., Stefano, L.D., Ajanthan, T., Torr, P.H.: Learning to adapt for stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9661–9670 (2019)
35. Gupta, S., Ullah, I., Madden, M.: Coyote: A dataset of challenging scenarios in visual perception for autonomous vehicles. In: AISafety@ IJCAI (2021)
36. Bhanushali, J., Muniyandi, M., Chakravarthula, P.: Cross-domain synthetic-to-real in-the-wild depth and normal estimation for 3d scene understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1290–1300 (2024)
37. Hao, K.: Training a single ai model can emit as much carbon as five cars in their lifetimes. MIT technology Review **75**, 103 (2019)
38. Paul, S., Jhamb, B., Mishra, D., Kumar, M.S.: Edge loss functions for deep-learning depth-map. Machine Learning with Applications **7**, 100218 (2022)
39. Liu, X., Gao, H., Ma, X.: Perceptual losses for self-supervised depth estimation. In: Journal of Physics: Conference Series, vol. 1952, p. 022040 (2021). IOP Publishing
40. Koch, T., Liebel, L., Fraundorfer, F., Korner, M.: Evaluation of cnn-based single-image depth estimation methods. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, pp. 0–0 (2018)
41. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. Advances in neural information processing systems **27** (2014)
42. Sagar, A.: Monocular depth estimation using multi scale neural network and feature fusion. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 656–662 (2022)