Depth Estimation of Thermal Image with Lightweights Model

Anonymous CVPR submission

Paper ID *****

Abstract

This paper focuses on lightweight modifications to the published thermal image depth estimation model, SupDepth4Thermal[17], and evaluates their effects on efficiency and accuracy. All models are trained validated and tested with the Multi-Spectral Stereo (MS^2) thermal image dataset, with training parameters referenced from the original paper[17]. First, the performance of the Lite-Mono[23] model in monocular depth estimation (MDE) is examined. The results show that Lite-Mono significantly reduces the mean inference time by 75% , from 40.58 ms to 10.28 ms, while the AbsRel error increases from 0.125 to 0.527. Next, in stereo depth estimation (SDE), the original Neural Window Conditional Random Field (NeWCRF) decoder in SupDepth4Thermal is replaced with depthwise separable convolutions. This leads to a substantial reduction in the inference time and Flops. However, the error rate deteriorated sharply. Finally, a new model is proposed by integrating the Lite-Mono core encoder, Parameterized Cost Volume (PCV) [22] construction, and the NeWCRF decoder. This configuration reduces the number of trainable parameters to roughly one-tenth of the original, lowers mean inference time by about 6%, and keeps the overall accuracy reasonably close to the original model, with moderate degradations (e.g., D1-all increases by about 6% on average). My source code is available at my GitHub.

1. Introduction

Depth estimation using thermal images has great potential because Long-Wave Infrared (LWIR) signals remain reliable in undesirable conditions such as low light and rain [12]. However, prior lightweight models and advanced depth estimation algorithms are mostly evaluated on RGB[20, 23], while systematic, reproducible studies on thermal remain scarce. Moreover, thermal images are single-channel and often texture-poor, making the task more challenging. In this paper, I propose three modifications to the existing thermal depth estimation model, SupDepth4Thermal[17], and investigate their efficiency and performance.

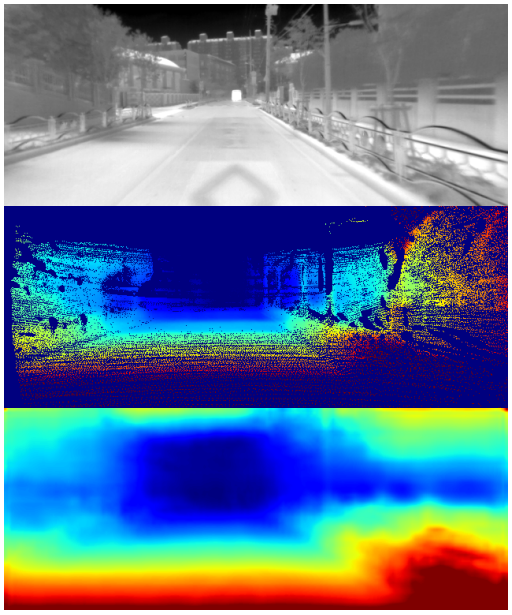


Figure 1. **Testing result of the model integrated with the Lite-Mono encoder and PCV** The original thermal image is from MS^2 daytime testset. From top to bottom: Thermal image, Ground-Truth(GT) disparity map and stereo disparity estimation

I choose to make the model lightweight due to limited computing resources. I can only run my algorithms and store all data on RCAC Scholar, where storage space is limited and GPU access is not always available (4-hour allocations with uncertain queue times). Therefore, a lightweight model is more suitable to reduce memory consumption and computing time.

In this paper, SupDepth4Thermal serves as the foundation for all implementations. The three modifications focus on these ideas:

- How does a Lite-Mono–style light encoder perform for thermal MDE?
- Can depthwise separable convolutions network replace SupDepth4Thermal's heavy decoder, NeWCRF blocks, in thermal SDE?

- Will integrating a Lite-Mono encoder together with a PCV computation method yield a better accuracy–efficiency trade-off in thermal SDE?

In all three experiments, the MS^2 dataset is used for training, validation, and testing. The dataset splits follow the settings of Shin *et al*., and all other training hyperparameters are kept the same as in their work. The scripts for training, testing, timing, and FLOPs evaluation are also the original implementations of Shin *et al*. The results of the three modifications are as follows:

- Lite-Mono for MDE: greatly shortens mean inference time to 25%, but accuracy drops at the same time. The average AbsRel increases from 0.125 to 0.527, SqRel from 1.130 to 6.739, and RMSE from 5.191 to 13.134.
- Depthwise separable decoder for SDE: replaces most of NeWCRF's heavy computation. The mean inference time remains only 59% of the original. However, this comes at the cost of a noticeable degradation in accuracy:: EPE-all increases 26%, D1-all increases 44%.
- Lite-Mono encoder plus PCV strategy for SDE: reduces parameters markedly by 10 times, while accuracy degrades: EPE-all increases 15%, D1-all increases 6.6%.

2. Related Work

2.1. MS^2 Dataset

MS^2 is a large outdoor dataset that includes about 195k pairs of synchronized and rectified multi-spectral stereo sensor data. It is multi-spectral, comprising RGB, near-infrared (NIR), long-wave infrared (LWIR), LiDAR, GNSS, and IMU. Those sufficient stereo data pairs support the construction of both MDE and SDE models. This was not possible for other benchmark datasets at the time because they were either indoor or had limited depth of field, making them less suitable for long-distance depth applications.

In addition, the dataset spans day, night, and rain conditions and also includes different scenes such as campus, residential, road, and suburban, providing robustness studies and realistic deployment evaluation.

MS^2 has been actively used in recent research. For example, Raviglia *et al*. leverages MS^2 to study RGB to IR data augmentation, synthesizing realistic infrared appearances from RGB images[15]. Zou *et al*. develop MonoTherDepth, a monocular thermal depth model trained and evaluated on MS^2[25]. These works illustrate MS^2's utility for both cross-spectral learning and thermal-specific depth estimation benchmarks.

2.2. SupDepth4Thermal

SupDepth4Thermal is a single model that can run both MDE and SDE on thermal (LWIR) images. Its backbone, Swin transformer, extracts a four-level feature pyramid (at 1/4, 1/8, 1/16, 1/32 of the input size). A pyramid pooling head injects global context, and the decoder consumes a 3D correlation cost volume when a right image is available. When only a left image is provided, which is under MDE operation, the same decoder operates with a zero-filled cost volume, so one architecture naturally "bridges" the mono and stereo tasks.

To be more specific, the functions of each module are as follows:

- Swin Transformer[11]: This transformer uses windowed self-attention plus shifted windows to link information across neighboring windows, yielding a hierarchical representation with good locality, efficiency, and long-range modeling. It is a generous vision backbone that can work on image classification, object detection and semantic segementation[1].
- PPM[24]: PPM performs multi-scale global pooling on the highest-level feature map and mixes those pooled descriptors back into the stream. This supplies scene-level context (layout, object extent) that complements weak local textures and stabilizes the decoder's predictions.
- Cost-Volume Construction: Shin *et al*. apply correlation cost volume at each scale. For each disparity d, it measures the inner-product similarity between the left feature at (x, y) and the right reference at $(x - d, y)$ as the cost volume C:

$$C^{scale}(d, x, y) = (1/N_c) * \langle f_L(x, y), f_R(x - d, y) \rangle$$

- NewCRF[21]:NewCRF fuses multi-scale features and the cost volume cues under a CRF-like regularization learned by a neural network. The neural window operations aggregate evidence within local windows, and the connected CRF encourages pixels with similar appearance to have compatible depths, while preserving sharp boundaries

Shin *et al*. evaluate the error rate and accuracy of SupDepth4Thermal with other MDE and SDE networks. The results show that it has competitive performance.

2.3. Lite-Mono

Lite-Mono is a lightweight model originally designed for RGB MDE that aims to keep accuracy competitive while cutting parameters, FLOPs, and latency. Zhang *et al*. put most effort on the design of the encoder which contains two major parts, Continuous Dilated Convolution (CDC) block and Local-Global Feature Interaction (LGFI) block. CDC block expands the effective receptive field via dilated depthwise 3×3 filters (for local structure) while keeping computation low, and LGFI block adds lightweight channel-wise attention to propagate scene-level cues that stabilize ambiguous regions.

In spite of the small model size of Lite-Mono, it still has good edge fidelity. Dynamo-Depth [18]is interested

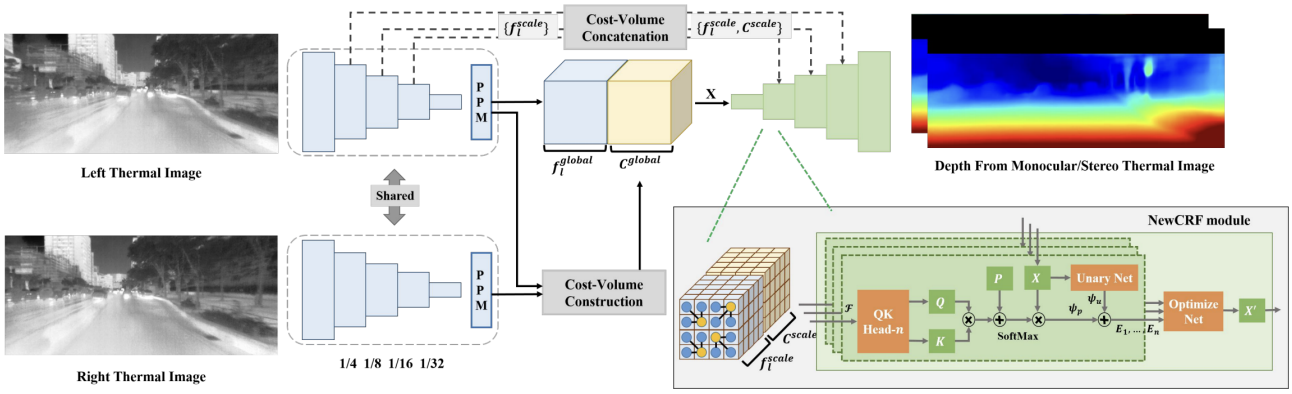


Figure 2. **Pipeline of SupDepth4Thermal** This network can do both MDE and SDE from given single or stereo thermal image. The network combined with three major parts, encoder, cost volume and decoder. In the beginning, Swin-Transformer downsamples the image and extracts the feature maps and Pyramid Pooling Module(PPM) aggregates the global context. Next, the 3D cost-volume is computed. If there is right thermal image, its cost volume will be combined into left one; otherwise, if only left image available, the network will view the right volume as zero-filled. Afterward, the volume and feature maps will be send into NewCRF based decoder, and carry out the final depth estimation

Table 1. **Comparison between MS^2 and other benchmark datasets.** Before the later published FIReStereo dataset, MS^2 addressed the lack of large-scale outdoor infrared imagery.

Dataset	Year	Environment	Platform	Total # of Data Pairs
MS^2 [17]	2022	Outdoor	Vehicle	195k
CATS [19]	2017	In\Outdoor	Handheld	1.4k
KAIST [2]	2018	Outdoor	Vehicle	Unknown
MultiSpectralMotion [4]	2021	In\Outdoor	Handheld	121k\27.3k
ViViD++ [8]	2022	Outdoor	Vehicle	5.6k
OdomBeyondVision [9]	2022	Indoor	Handheld\UGV\UAV	71k\117k\21k
FIReStereo [5]	2025	Outdoor	UAV	102k

in Lite-Mono precisely for this accuracy–efficiency balance: a compact encoder leaves compute headroom for temporal modules (e.g., motion-aware warping, recurrent refinement), while CDC delivers sharp per-frame edges and LGFI contributes global semantics that make temporal fusion more stable. In practice, Dynamo-Depth uses Lite-Mono as the backbone encoder, then stacks a temporal head on top to aggregate features across adjacent frames, improving dynamic-scene depth without significantly increasing latency or parameters.

Zhang *et al*. evaluate different size of Lite-Mono networks with other representative networks on KITTI[6] dataset. The results indicate that Lite-Monos not only have higher accuracy and lower error, but also have relatively small size.

2.4. Depthwise Separable Convolutions

The popularity of depthwise separable convolution can be attributed to its integration into models like Xception[3] and MobileNet[7].Depthwise convolution performs spatial filtering independently on each input channel. Instead of applying a single convolutional kernel across all channels, it assigns a separate spatial kernel to each channel, efficiently enlarging the receptive field and capturing local structures. This design greatly reduces the number of parameters and computational cost compared to standard convolutions, while preserving strong representational power.

The reason I use depthwise separable convolution is because I want to replace the NewCRF module with a Lite-Mono-style decoder. The similarity between Lite-Mono and depthwise separable convolution is that both follow a "local spatial filtering + channel mixing" design. They all emphasize the cheapest way to build effective receptive fields and feature representations.

One of the classic applications that implements depthwise separable convolution is FastDepth[20]. FastDepth adopts a lightweight architecture where depthwise separable convolutions are used extensively. The application of

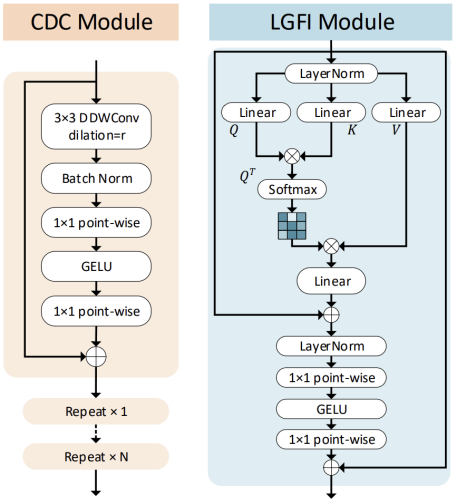


Figure 3. **Pipeline of CDC block and LGFI block.** CDC module utilizes dilated convolutions to extract multi-scale local features. LGFI module take advantage of the self-attention mechanism to encode long range global information into the features.

this network reduce the FLOPs and parameters while maintaining reasonable accuracy. This design enables real-time monocular depth estimation on embedded or mobile devices, with a very small model size and fast inference speed compared to conventional CNN-based depth networks.

2.5. PCV

Traditional stereo builds a 3D/4D cost volume ($H \times W \times D \times C$) which requires large number of memory and computation. Although iterative method like recurrent update[10] could reduce memory usage, it still takes several iterations to converge which cost time. Instead of representing each pixel's disparity evidence with a dense D-slice volume, PCV method represents its disparity distribution using only K Gaussians components. That is, each pixel then keeps just a few parameters, effectively compressing the entire disparity curve. The concept of PCV is like:

$$p(d \mid x, y) = \sum_{i=1}^{K} \alpha_i(x, y)\, \mathcal{N}\left(d;\, \mu_i(x, y),\, \sigma_i^2(x, y)\right)$$

α_i is weight, μ_i is mean and σ_i is variance. It reveals that PCV reduces the cost volume parameters from traditional ($H \times W \times D \times C$) to ($H \times W \times 3K$). In Zeng *et al*. experiments, PCV indeed achieves the intended effect. On the SceneFlow[13], KITTI[6], Middlebury[16], and Booster[14] datasets, it reduced the runtime by about 4 to 15 times compared with RAFT-Stereo[10], without sacrificing accuracy.

3. Method

3.1. Set Up

All experiments are conducted by using RCAC Scholar serve. I download the source code of SupDepth4Thermal, Lite-Mono and PCV from the following GitHub repositories:

- SupDepth4Thermal: https://github.com/UkcheolShin/SupDepth4Thermal
- Lite-Mono: https://github.com/noahzn/Lite-Mono
- PCV: https://github.com/jiaxiZeng/Parameterized-Cost-Volume-for-Stereo-Matching

For the MS^2 dataset, it is required to visit the official website https://sites.google.com/view/multi-spectral-stereo-dataset and fill a survey form. Afterward, I receive an email with the instructions and download links.

I only download the "sync_data" and the "proj_depth" folders because this paper only requires those data. I store them under RCAC Scholar personal "/scratch" directory, considering their over 200G storage requirement, and also delete the source zipped files right after the unaip in order to remain sufficient space.

The Conda environment setting could be referenced to "environment.yaml" in my GitHub repository. All experiments are developed and tested in the following package:

- OS: Rocky Linux 9.6 (Blue Onyx)
- CUDA: 12.1.105
- PyTorch: 2.3.0+cu121
- Python: 3.10.19

To mitigate unexpected interruption on RCAC Scholar, I adopt the checkpoints mechanism in training script. Checkpoints are saved when any of the following three conditions is met: every 500 steps, at the end of each epoch and upon achieving a new lowest validation loss.

3.2. Optimizer and Data Augmentation

All models are trained on RCAC Scholar using NVIDIA V100 (16 GB), A40 (48 GB), or A30 (24 GB) GPUs. MDE model is trained for 30 epochs; SDE models for 60 epochs. All other training hyperparameters and setting are aligned with those of Shin *et al*.

3.3. Source Code Modification

In general, I create dedicated yaml files and trainer scripts for each modification experiment. Moreover, I edit every "__init__.py" file to ensure the targeted files could be imported correctly.

- Lite-Mono for MDE: In the codebase, this model is registered as "litemono". From the source repository, I reuse only "depth_encoder", "depth_decoder", and "layers". I

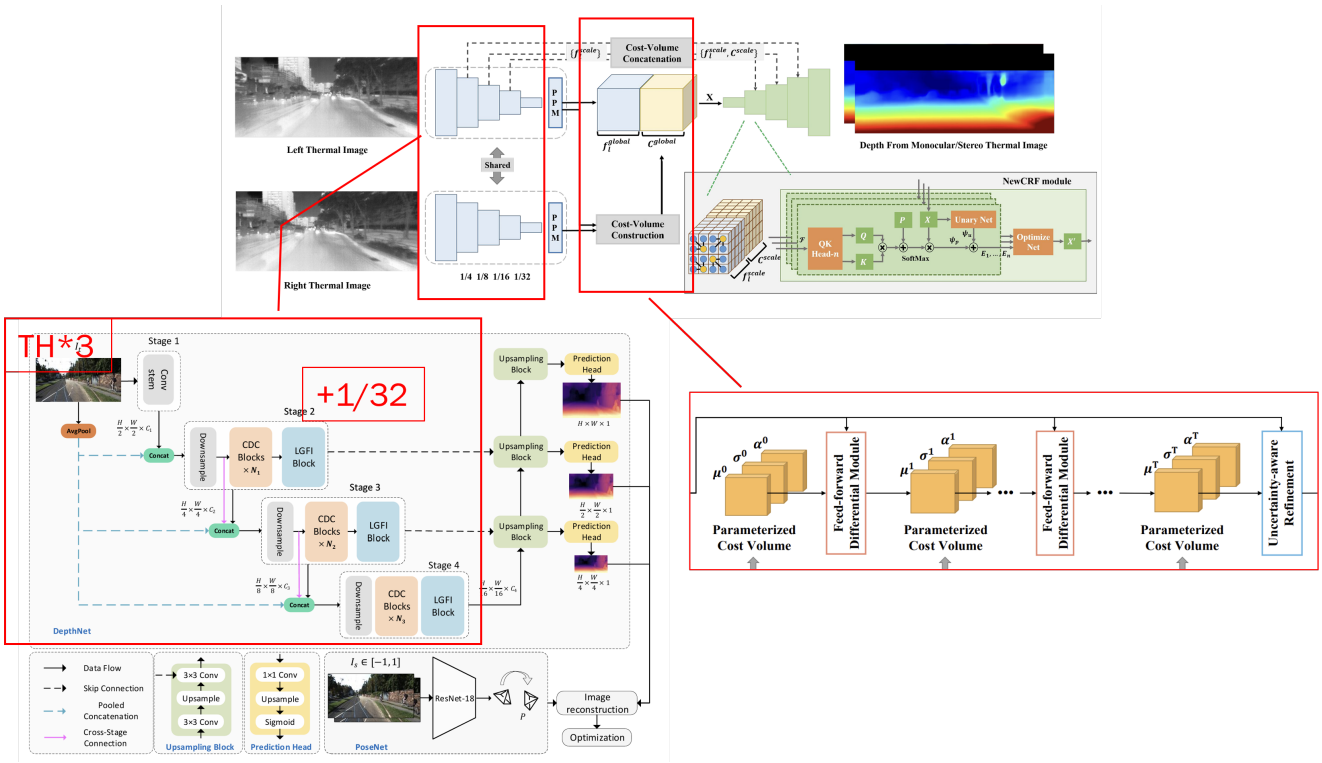


Figure 4. **Pipeline of Lite-Mono encoder plus PCV strategy for SDE** The original swin transformer is replaced with Lite-Mono encoder, and the correlation cost volume method also be replaced with PCV. Since Lite-Mono take 3-channels input, thermal image data is replicated into 3 times. An additional 1/32 scale is prepared for the NewCRF module.

exclude ”resnet_encoder”, as Lite-Mono does not rely on a ResNet backbone. The ”pose_decoder” is also omitted: the base model uses GT-depth supervision, so no self-supervised photometric pose is needed.

Since Lite-Mono expects RGB, the 1-channel thermal input is channel-replicated ($\times 3$) to conform to the 3-channel interface.

- Depthwise separable decoder for SDE: In the codebase, this model is registered as ”LiteMono_crf”, although it primarily retains only the Lite-Mono–style depthwise convolution design. The overall framework of this network is adapted from ”ms_crf” directory which created by Shin *et al*. I duplicate the directory and refactore its contents. The primary distinction is dedicated head, ”lite-mono_head”, which plugs in a depthwise design. Furthermore, I drop the original NewCRF decoder pattern

  Another crucial alteration is in the ”multiscale_loss” function of its trainer,”LiteMono_crf”. I revise the disparity to depend on itself, the weight GT and weight prediction.
- Lite-Mono encoder plus PCV strategy for SDE: In the codebase, this model is registered as ”LiteMonoEn”. I reuse not only ”depth_encoder”, ”depth_decoder”, and ”layers” from Lite-Mono repositry, but also ”corr”,”update” and ”utils” from pcv/core repository.

  This model is also derived from ”ms_crf”. I specifically create a file ”LiteMonoEn” to bridge the Lite-Mono encoder, PCV, and the original NeWCRF module. In addition, I prepare a new file ”pcv_module” for pcv to interface the features after PPM.

  A major modification in this model is to align the original three-scale LiteMono architecture with the four-scale NeWCRF. Therefore, I add an extra 1/32 scale inside Lite-Mono encoder. Beyond that, considering PCV default 1/4 input scale, it extracts features at 1/4 and then feeds the output into NeWCRF initial 1/32 stage.

4. Result

4.1. Lite-Mono for MDE:

From Tab. 2, compared with original SupDepth4Thermal, Lite-Mono network variant reduces the parameter count by roughly 88 times, decreases FLOPs by about 24 times, and achieves nearly 4 times faster inference.

However, from Tab. 3, SupDepth4Thermal achieves an average AbsRel of about 0.125 and $\delta < 1.25$ of about 0.88. In contrast, Lite-Mono yields an average AbsRel of about 0.527—roughly four times larger—and $\delta < 1.25$ of only around 0.30, indicating a significant degradation in depth

Table 2. **Comparison between SupDepth4Thermal Mono network and Lite-Mono network.** "#Parameters" counts trainable weights; "#FLOPs" denotes floating-point operations per forward pass; "Mean inference time" is averaged over the evaluation set. Lower is better.

Model	#Parameter of Shape	#Flops	Mean inference time
SupDepth	0.27G	0.158T	40.580
Lite-Mono	3.075M	6.713G	10.282

accuracy.

In summary, the LiteMono-based model is highly efficient in terms of model size, computation, and runtime, but in the current configuration this efficiency is obtained at the cost of a substantial loss in depth accuracy.

4.2. Depthwise separable decoder for SDE:

The major difference between the original SDE model and depthwise seperable network lies in the replacement of the NeWCRF decoder.. The results from Tab. 5 indicate that NewCRF is a heavy module, with immense parameters and subsantial computing time. However, from Tab. 4, it reveals that the new depthwise separable convolution model fails to maintain accuracy. In other words, the gain in speed comes at the cost of degraded performance in this modification.

4.3. Lite-Mono encoder plus PCV strategy for SDE:

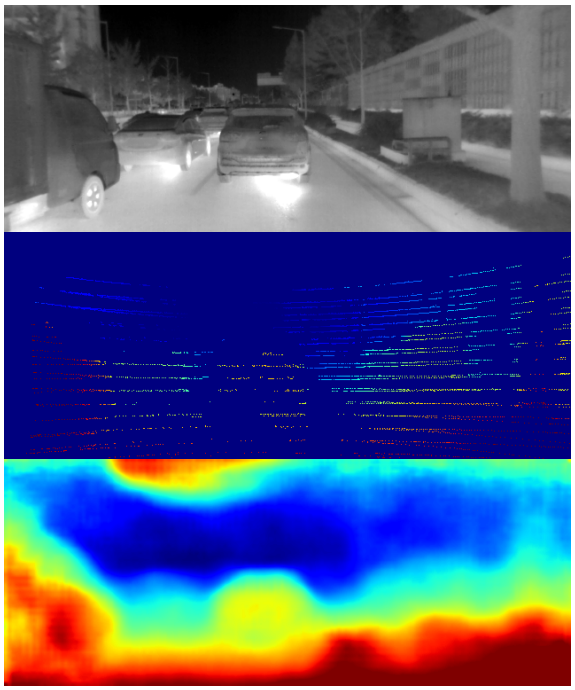
In contrast to the previous modification, in this section, the encoder and cost-volume computation algorithm is replaced from original SDE model. The Tab. 5 results show that this new model does become lightweight. However, the inference time just slightly improve. Although the combination of Lite-Mono encoder and PCV make the model lighter, the error rate of new model still increase: EPE-all increases 15%, D1-all increases 6.6%.

4.4. Comparison between new SDE models

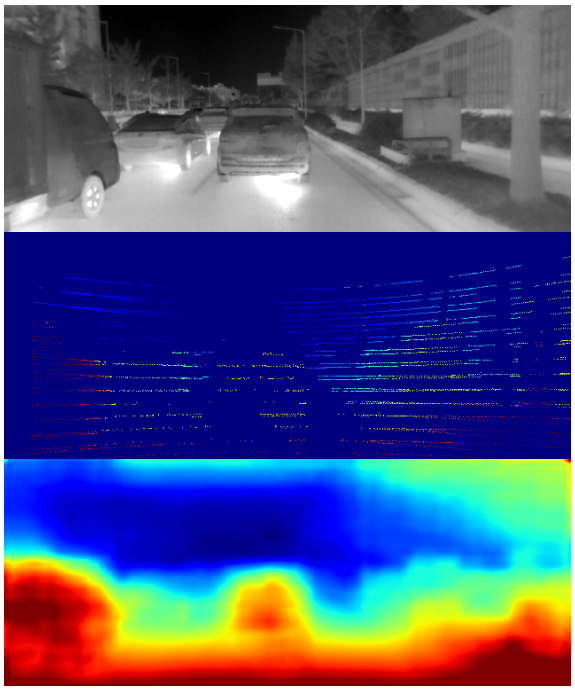
According to Tab. 4, the Lite + PCV model has better accuracy than Depthwise model. This outcome could be examined in Fig. 5.The prediction from depthwise model is not only inaccurate for the car in the center, whose estimated depth does not approach the orange GT region, but also unstable in the surrounding areas. The truck on the left is not highlighted with a consistent red region, spurious red noise appears in the sky at the top, and the electrical box on the right side in the grass is not detected at all.

4.5. Discussion

There are still many potential modifications to explore. For example, one could fully replace SupDepth4Thermal by combining LiteMono, PCV, and depthwise separable convolutions into a single unified architecture.



(a) Test result from Depthwise model.



(b) Test result from Lite + PCV model.

Figure 5. **Comparison between new SDE models** Both figures are generated from the rain scenario of the testing dataset and correspond to sample 01930. In each sub-figure, from top to bottom: Thermal image, GT disparity map and stereo disparity estimation

Table 3. **Comparison of MDE models performance.** AbsRel, SqRel, RMSE, RMSElog (lower is better) and accuracy thresholds $\delta < 1.25$,$\delta < 1.25^2$,$\delta < 1.25^3$ (higher is better).

		Error				Accuracy		
Model	Test set	AbsRel	SqRel	RMSE	RMSElog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
SupDepth4Theraml (Mono)	Day	0.115	0.983	4.895	0.201	0.882	0.952	0.977
	Night	0.107	0.850	4.658	0.185	0.894	0.961	0.981
	Rain	0.152	1.567	6.020	0.247	0.822	0.928	0.964
	Avg	0.125	1.130	5.191	0.211	0.866	0.947	0.974
LiteMono (DepthNet encoder)	Day	0.539	6.794	13.571	0.644	0.286	0.519	0.696
	Night	0.486	5.611	12.377	0.592	0.306	0.556	0.740
	Rain	0.558	7.811	13.455	0.638	0.308	0.535	0.708
	Avg	0.527	6.739	13.134	0.625	0.900	0.537	0.715

Table 4. **SDE performance comparison.**EPE-all, D1-all, and bad-pixel ratios >1px, >2px, >3px are reported on the day, night, and rain test sets (lower is better)

		Lower is better				
Model	Test set	EPE-all (px)	D1-all (%)	>1px (%)	>2px (%)	>3px (%)
SupDepth4Thermal	Day	0.957	5.7	22.7	9.1	5.7
	Night	0.853	4.8	21.3	8.2	4.8
	Rain	1.159	7.7	29.1	12.4	7.7
	Avg	0.990	6.1	24.4	9.9	6.1
Depwise	Day	1.040	6.0	33.1	12.9	6.0
	Night	1.310	9.7	41.3	18.6	9.7
	Rain	1.404	10.7	46.9	21.4	10.8
	Avg	1.251	8.8	40.4	17.6	8.8
Lite + PCV	Day	1.021	5.3	34.1	12.1	5.3
	Night	1.105	5.5	40.4	13.9	5.5
	Rain	1.293	8.8	44.5	18.3	8.8
	Avg	1.140	6.5	39.7	14.8	6.5

Table 5. **Comparison between SupDepth4Thermal SDE network, Depwise seperable decoder network (Depthwise) and Lite-Mono plus PCV network (Lite+PCV).** "#Parameters" counts trainable weights; "#FLOPs" denotes floating-point operations per forward pass; "Mean inference time" is averaged over the evaluation set. Lower is better.

Model	#Parameter of Shape	#Flops	Mean inference time
SupDepth	0.284G	0.322T	96.537
Depthwise	61.46M	47.457G	56.929
Lite+PCV	19.381M	80.764G	90.540

In addition, there is still a large hyperparameter space to investigate. For LiteMono, I currently use the tiny configuration, but the small, normal, and 8M variants size could also be tested. Training hyperparameters such as learning rate, batch size, and regularization terms can likewise be tuned more systematically.

Another interesting direction is to understand why the LiteMono + PCV model achieves a lightweight model in terms of parameters, yet does not yield a proportionally large reduction in inference time.

Finally, the trained models can be further evaluated on other benchmark datasets to examine their generalization ability and robustness across different domains and acquisition conditions.

5. Conclusion

In this work, I systematically explore three lightweight variants of SupDepth4Thermal for both monocular and stereo depth estimation. For MDE, replacing the original backbone with LiteMono significantly reduces model complex-

ity, but the error metrics in Tab. 3 show a clear degradation compared to the SupDepth4Thermal monocular baseline, indicating that the current integration is not yet competitive in terms of accuracy.

For SDE, the depthwise separable decoder and the Lite-Mono encoder plus PCV strategy both succeed in making the stereo network substantially more lightweight in terms of parameter count and FLOPs, as summarized in Tab. 5. However, this comes with an increase in disparity error across different settings, meaning that efficiency gains are achieved at the cost of prediction quality.

Overall, these results highlight a classic accuracy–efficiency trade-off. The overly simplified architecture leads to nontrivial increases in error. To close this gap, exploring alternative encoder-decoder designs and improving the cost volume computation algorithm are vital directions for future work.

References

[1] Hu Cao, Yueyue Wang, Joy Chen, Dongsheng Jiang, Xiaopeng Zhang, Qi Tian, and Manning Wang. Swin-unet: Unet-like pure transformer for medical image segmentation. In *European conference on computer vision*, pages 205–218. Springer, 2022. 2

[2] Yukyung Choi, Namil Kim, Soonmin Hwang, Kibaek Park, Jae Shin Yoon, Kyounghwan An, and In So Kweon. Kaist multi-spectral day/night data set for autonomous and assisted driving. *IEEE Transactions on Intelligent Transportation Systems*, 19(3):934–948, 2018. 3

[3] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 3

[4] Weichen Dai, Yu Zhang, Shenzhou Chen, Donglei Sun, and Da Kong. A multi-spectral dataset for evaluating motion estimation systems. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5560–5566. IEEE, 2021. 3

[5] Devansh Dhrafani, Yifei Liu, Andrew Jong, Ukcheol Shin, Yao He, Tyler Harp, Yaoyu Hu, Jean Oh, and Sebastian Scherer. Firestereo: Forest infrared stereo dataset for uas depth perception in visually degraded environments. *IEEE Robotics and Automation Letters*, 2025. 3

[6] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013. 3, 4

[7] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3

[8] Alex Junho Lee, Younggun Cho, Young-sik Shin, Ayoung Kim, and Hyun Myung. Vivid++: Vision for visibility dataset. *IEEE Robotics and Automation Letters*, 7(3):6282–6289, 2022. 3

[9] Peize Li, Kaiwen Cai, Muhamad Risqi U Saputra, Zhuangzhuang Dai, and Chris Xiaoxuan Lu. Odombeyondvision: An indoor multi-modal multi-platform odometry dataset beyond the visible spectrum. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3845–3850. IEEE, 2022. 3

[10] Lahav Lipson, Zachary Teed, and Jia Deng. Raft-stereo: Multilevel recurrent field transforms for stereo matching. In *2021 International Conference on 3D Vision (3DV)*, pages 218–227. IEEE, 2021. 4

[11] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2

[12] Yawen Lu and Guoyu Lu. An alternative of lidar in nighttime: Unsupervised depth estimation based on single thermal image. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3833–3843, 2021. 1

[13] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016. 4

[14] Pierluigi Zama Ramirez, Fabio Tosi, Matteo Poggi, Samuele Salti, Stefano Mattoccia, and Luigi Di Stefano. Open challenges in deep stereo: the booster dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21168–21178, 2022. 4

[15] Leonardo Ravaglia, Roberto Longo, Kaili Wang, David Van Hamme, Julie Moeyersoms, Ben Stoffelen, and Tom De Schepper. Rgb-to-infrared translation using ensemble learning applied to driving scenarios. *Journal of Imaging*, 11(7):206, 2025. 2

[16] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pages 31–42. Springer, 2014. 4

[17] Ukcheol Shin, Jinsun Park, and In So Kweon. Deep depth estimation from thermal image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1043–1053, 2023. 1, 3

[18] Yihong Sun and Bharath Hariharan. Dynamo-depth: Fixing unsupervised depth estimation for dynamical scenes. *Advances in Neural Information Processing Systems*, 36: 54987–55005, 2023. 2

[19] Wayne Treible, Philip Saponaro, Scott Sorensen, Abhishek Kolagunda, Michael O'Neal, Brian Phelan, Kelly Sherbondy, and Chandra Kambhamettu. Cats: A color and thermal stereo benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2961–2969, 2017. 3

[20] Diana Wofk, Fangchang Ma, Tien-Ju Yang, Sertac Karaman, and Vivienne Sze. Fastdepth: Fast monocular depth esti-

mation on embedded systems. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6101–6108. IEEE, 2019. 1, 3

[21] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3916–3925, 2022. 2

[22] Jiaxi Zeng, Chengtang Yao, Lidong Yu, Yuwei Wu, and Yunde Jia. Parameterized cost volume for stereo matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18347–18357, 2023. 1

[23] Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18537–18546, 2023. 1

[24] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 2

[25] Xingxing Zuo, Nikhil Ranganathan, Connor Lee, Georgia Gkioxari, and Soon-Jo Chung. Monother-depth: Enhancing thermal depth estimation via confidence-aware distillation. *IEEE Robotics and Automation Letters*, 2025. 2