Mobile AI 2025 Challenge Factsheet

⟨ sRGB Image Enhancement Challenge ⟩

Runhua Deng, Xuanyu Chen, Shuhui Xie, Guojie Xiao,
Zhifeng Wang, Long Peng, Aiwen Jiang

March 28, 2025

1 Team details

- Team Name: MotongAI
- Team Leader: Runhua Deng
- Jiangxi Normal University, China, 18138655705, and 2312666252@qq.com
- Rest of the team members: Xuanyu Chen, Shuhui Xie, Guojie Xiao, Zhifeng Wang, Long Peng, Aiwen Jiang

---

- User Name: Alex_Fall

- Affiliation: None
- Team website URL: https://motong-ai-studio.top/index.html
- Best scoring entries of the team during development / validation phases: None
- Link to codes / executables: None

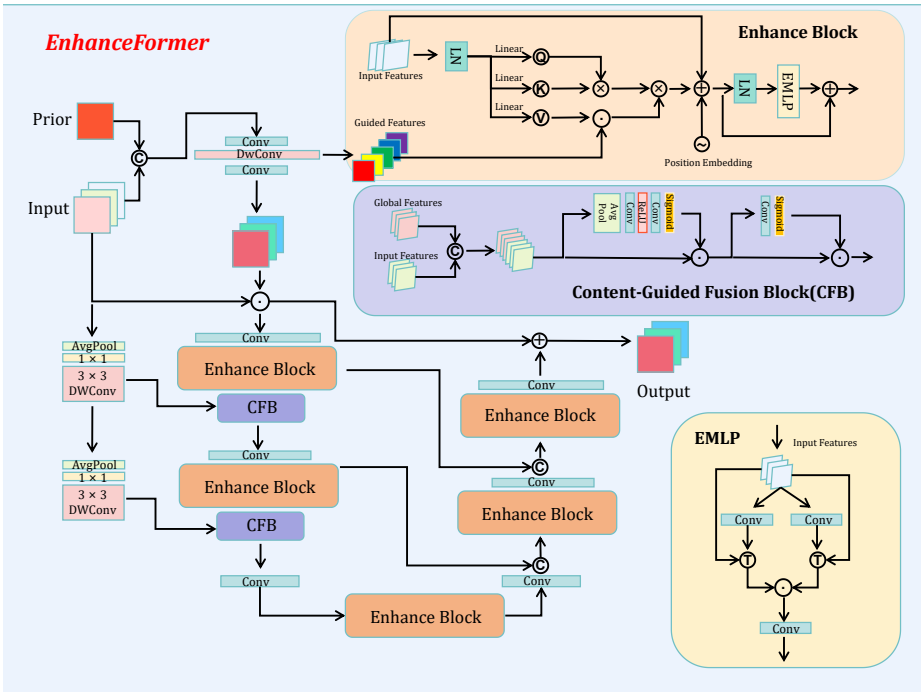

Figure 1: Overview of our Model.

2 Detailed Method Description

- In recent years Retinex theory has been popularized in Low-Level field, which is a theory that mimics the neural processing of information in the human retina.Retinex theory decomposes the input image into illuminance component and reflectance component, and then denoises and enhances them respectively to obtain a clear image. Our team's model uses a deep learning approach based on Retinex theory, where the illuminance component is first estimated for the image for enhancement, and according to Retinex theory, the enhanced image is exposed to noise. To remove the noise, we use Transformer based Unet structure for multi-scale noise removal. Our team's approach is called Enhanceformer.

- The main body of our Enhanceformer is a Unet structure, mainly characterized by the design of the Enhance Block, CFB and EMLP. First, we obtain Prior by averaging each pixel point of Input over the channel, and then splicing Input with Prior is input to a three-layer convolution, the middle layer of convolution using deep-wise convolution. this estimates the Map and bootstrap features for enhancement, which are multiplied with the Input elements to obtain the enhanced image. To remove the noise,

we feed the enhanced image to the Transformer based Unet. each layer of the Unet consists of a downsampled convolutional layer, an Enhance Block and a feature fusion module CFB. to better utilize the global information, we design a global information extraction branch consisting of an AvgPool with convolution. the CFB is used in the channel attention and spatial The CFB effectively combines the extracted global information with the input features at both channel attention and spatial attention scales.The Enhance Block consists of a LayerNorm, a self-attention mechanism and an EMLP. In order to better utilize the features of the enhanced image, we multiply the intermediate features of the enhanced image with the Value element to enhance the key information in the image.EMLP adopts the idea of partitioning and divides the features into two parts for processing, which effectively reduces the redundant processing caused by duplicated features, and the two branches undergo feature fusion after the Tanh activation function.

- We use L1 loss for supervised training.During training, we use mixed precision for accelerated training and gradient cropping to prevent gradient explosion and gradient vanishing.

- We use Adam as an optimizer with beta values of 0.9 and 0.99. We use a cosine annealing learning rate scheduler to control the variation of our learning rate. Our total iterations are set to 21562 and batchsize is set to 512. Our initial learning rate is 1e-3, which first undergoes 6612 iters to warm up to 1.5e-3, then the learning rate is reset to 1e-3 and undergoes 14950 iters to drop to 5e-6. In particular, we observe that the training data are all of the same size: 100*100. Therefore, we merge the three subsets of the training set together for training. We feed the blackberry, iphone, and sony paired datasets together to train our model. This enables our model to be robust enough to simultaneously enhance all images obtained from different devices into high quality DSLR images.

- The architecture of our model is shown in Figure 1.It should be noted that the symbol T in the EMLP structure diagram denotes the Tanh activation function.

3 Model Optimization and TFLite Conversion

- For TFLite models with fixed input sizes, we use the ai-edge-torch plugin to convert Pytorch models to TFLite models.

- Clarification on our TFLite model: Moreover, our TFLite model currently only runs successfully in CPU mode and NNAPI mode. Our local run on AI Benchmark shows that the BATCH MATMUL operator is not

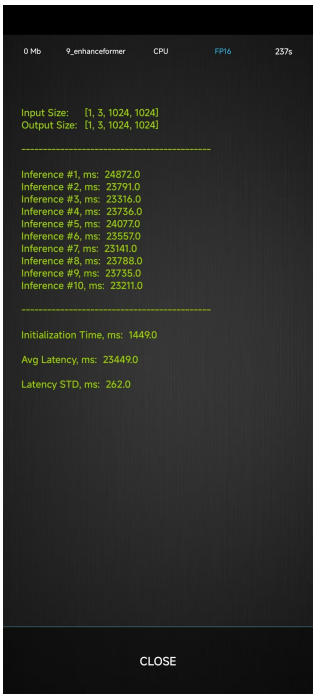

Figure 2: The results of our model run on AI Benchmark with 1*1024*1024*3 as fixed-size input and FP16+CPU as mode.

supported by GPU Delegate. Due to the ai-edge-torch plugin itself, ai-edge-torch defaults to this operator as a basic arithmetic operation, and thus a large number of BATCH MATMUL operations have been inserted everywhere in the model. We are currently unable to resolve the issue of running GPU Delegate locally.

- CPU+FP16(2): 23449ms. NNAPI+FP16(3): 19986ms.
- Model runtime on the target platform: We don't know.
- Our model, although based on Transformer, utilizes a number of structural optimizations, so we believe that our model is computationally efficient.

4 Other Technical Questions

- We use Pytorch as the model structure for implementation, training and export.

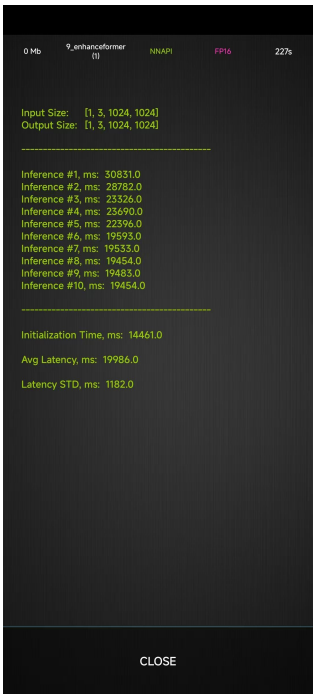

Figure 3: he results of our model run on AI Benchmark with 1*1024*1024*3 as fixed-size input and FP16+NNAPI as mode.

- We used 2 NVIDIA A100 to train a model with robustness and effectiveness.
- We did not use any external methods or pre-trained models.
- We used only Mobile AI's training data for training and did not use any external data. Similarly, the validation data is all from Mobile AI.
- We have not tried this on other datasets for the time being.

5 Other details

- We have not yet published our method as a paper.
- Mobile AI 2025's new sRGB Enhancement Challenge is a challenge that is both challenging, applied, and innovative. This challenge requires that participants' AI models be deployed to mobile, which requires participants' models to be lightweight, efficient, and fast in reasoning while maintaining

performance. This challenge provides a good reference for image enhancement techniques in industry.

- Looking forward to adding the image de-motion blur challenge.
- Nowadays, Pytorch framework is popular and many deep learning models are based on Pytorch framework, however, the conversion of Pytorch models to TFLite models is more difficult compared to Tensorflow, which may limit the enthusiasm and self-confidence of the participants. On top of that, the degree of synergy between GPU Delegate and these plugins that convert Pytorch to TFLite models is also very important. Otherwise, even if the Pytorch model is successfully converted to a TFLite model, the GPU Delegate's inability to support the operators in TFLite can be a tricky issue.

6 References

We refer to these papers and methods: [1], [2], [3]

References

[1] Yuanhao Cai, Hao Bian, Jing Lin, Haoqian Wang, Radu Timofte, and Yulun Zhang. Retinexformer: One-stage retinex-based transformer for low-light image enhancement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12504–12513, October 2023.

[2] Qingsen Yan, Yixu Feng, Cheng Zhang, Guansong Pang, Kangbiao Shi, Peng Wu, Wei Dong, Jinqiu Sun, and Yanning Zhang. Hvi: A new color space for low-light image enhancement. *arXiv preprint arXiv:2502.20272*, 2025.

[3] Shangquan Sun, Wenqi Ren, Xinwei Gao, Rui Wang, and Xiaochun Cao. Restoring images in adverse weather conditions via histogram transformer. *arXiv preprint arXiv:2407.10172*, 2024.