Learned Smartphone ISP on Mobile NPUs with Deep Learning, Mobile AI 2021 Challenge: Report

Andrey Ignatov Cheng-Ming Chiang Hsien-Kai Kuo Anastasia Sycheva
Radu Timofte Min-Hung Chen Man-Yu Lee Yu-Syuan Xu Yu Tseng
Shusong Xu Jin Guo Chao-Hung Chen Ming-Chun Hsyu Wen-Chia Tsai
Chao-Wei Chen Grigory Malivenko Minsu Kwon Myungje Lee Jaeyoon Yoo
Changbeom Kang Shinjo Wang Zheng Shaolong Hao Dejun Xie Fen
Feng Zhuang Yipeng Ma Jingyang Peng Tao Wang Fenglong Song
Chih-Chung Hsu Kwan-Lin Chen Mei-Hsuang Wu Vishal Chudasama
Kalpesh Prajapati Heena Patel Anjali Sarvaiya Kishor Upla Kiran Raja
Raghavendra Ramachandra Christoph Busch Etienne de Stoutz

Abstract

As the quality of mobile cameras starts to play a crucial role in modern smartphones, more and more attention is now being paid to ISP algorithms used to improve various perceptual aspects of mobile photos. In this Mobile AI challenge, the target was to develop an end-to-end deep learning-based image signal processing (ISP) pipeline that can replace classical hand-crafted ISPs and achieve nearly real-time performance on smartphone NPUs. For this, the participants were provided with a novel learned ISP dataset consisting of RAW-RGB image pairs captured with the Sony IMX586 Quad Bayer mobile sensor and a professional 102-megapixel medium format camera. The runtime of all models was evaluated on the MediaTek Dimensity 1000+ platform with a dedicated AI processing unit capable of accelerating both floating-point and quantized neural networks. The proposed solutions are fully compatible with the above NPU and are capable of processing Full HD photos under 60-100 milliseconds while achieving high fidelity results. A detailed description of all models developed in this challenge is provided in this paper.

1. Introduction

While the quality of modern smartphone cameras increases gradually, many major improvements are currently coming from advanced image processing algorithms used, *e.g.*, to perform noise suppression, accurate color reconstruction or high dynamic range processing. Though the image enhancement task can be efficiently solved with deep learning-based approaches, the biggest challenge here comes from getting the appropriate training data and, in particular, the high-quality ground truth images. The problem of end-to-end mobile photo quality enhancement was first addressed in [16, 17], where the authors proposed to enhance all aspects of low-quality smartphone photos by mapping them to superior-quality images obtained with a high-end reflex camera. The collected DPED dataset was later used in many subsequent competitions [29, 23] and works [51, 41, 8, 14, 13, 38] that have significantly improved the results on this problem. While the proposed methods were quite efficient, they worked with the data produced by smartphones' built-in ISPs, thus a significant part of information present in the original sensor data was irrecoverably lost after applying many image processing steps. To address this problem, in [31] the authors proposed to work directly with the original RAW Bayer sensor data and learn all ISP steps with a single deep neural network. The experiments conducted on the collected *Zurich RAW to RGB* dataset containing RAW-RGB image pairs captured by a mobile camera sensor and a high-end DSLR camera demonstrated that the proposed solution was able to get to the level of commercial ISP of the Huawei P20 cameraphone, while these results were later improved in [30, 7, 45, 35, 26]. In this challenge, we take one step further in solving this problem by using more advanced data and by putting additional efficiency-related constraints on the developed solutions.

When it comes to the deployment of AI-based solutions on mobile devices, one needs to take care of the particu-

*Andrey Ignatov, Cheng-Ming Chiang, Hsien-Kai Kuo and Radu Timofte are the main MAI 2021 challenge organizers *(andrey@vision.ee.ethz.ch, jimmy.chiang@mediatek.com, hsienkai.kuo@mediatek.com, radu.timofte @vision.ee.ethz.ch)*. The other authors participated in the challenge.
Appendix A contains the authors' team names and affiliations.

Mobile AI 2021 Workshop website:
https://ai-benchmark.com/workshops/mai/2021/

Figure 1. A sample set of images from the collected dataset. From left to right: the original RAW image visualized with Photoshop's built-in raw image processing engine, RGB image obtained with MediaTek's built-in ISP system, and the target Fujifilm photo.

larities of mobile NPUs and DSPs to design an efficient model. An extensive overview of smartphone AI acceleration hardware and its performance is provided in [27, 24]. According to the results reported in these papers, the latest mobile NPUs are already approaching the results of mid-range desktop GPUs released not long ago. However, there are still two major issues that prevent a straightforward deployment of neural networks on mobile devices: a restricted amount of RAM, and limited and not always efficient support for many common deep learning layers and operators. These two problems make it impossible to process high-resolution data with standard NN models, thus requiring a careful adaptation of each architecture to the restrictions of mobile AI hardware. Such optimizations can include network pruning and compression [6, 20, 37, 39, 43], 16-bit / 8-bit [6, 34, 33, 55] and low-bit [5, 50, 32, 40] quantization, device- or NPU-specific adaptations, platform-aware neural architecture search [11, 46, 54, 52], *etc*.

While many challenges and works targeted at efficient deep learning models have been proposed recently, the evaluation of the obtained solutions is generally performed on desktop CPUs and GPUs, making the developed solutions not practical due to the above-mentioned issues. To address this problem, we introduce the first *Mobile AI Workshop and Challenges*, where all deep learning solutions are developed for and evaluated on real mobile devices. In this competition, the participating teams were provided with a new ISP dataset consisting of RAW-RGB image pairs captured with the Sony IMX586 mobile sensor and a professional 102-megapixel Fujifilm camera, and were developing an end-to-end deep learning solution for the learned ISP task. Within the challenge, the participants were evaluating the runtime and tuning their models on the MediaTek Dimensity 1000+ platform featuring a dedicated AI Processing Unit (APU) that can accelerate floating-point and quantized neural networks. The final score of each submitted solution was based on the runtime and fidelity results, thus balancing between the image reconstruction quality and efficiency of the proposed model. Finally, all developed solutions are fully compatible with the TensorFlow Lite framework [47], thus can be deployed and accelerated on any mobile platform providing AI acceleration through the Android Neural Networks API (NNAPI) [1] or custom TFLite delegates [9].

This challenge is a part of the *MAI 2021 Workshop and Challenges* consisting of the following competitions:

- Learned Smartphone ISP on Mobile NPUs
- Real Image Denoising on Mobile GPUs [15]
- Quantized Image Super-Resolution on Mobile NPUs [25]
- Real-Time Video Super-Resolution on Mobile GPUs [22]
- Single-Image Depth Estimation on Mobile Devices [18]
- Quantized Camera Scene Detection on Smartphones [19]
- High Dynamic Range Image Processing on Mobile NPUs

The results obtained in the other competitions and the description of the proposed solutions can be found in the corresponding challenge report papers.

2. Challenge

To develop an efficient and practical solution for mobile-related tasks, one needs the following major components:

1. A high-quality and large-scale dataset that can be used to train and evaluate the solution on real (not synthetically generated) data;
2. An efficient way to check the runtime and debug the model locally without any constraints;
3. An ability to regularly test the runtime of the designed neural network on the target mobile platform or device.

This challenge addresses all the above issues. Real training data, tools, and runtime evaluation options provided to the challenge participants are described in the next sections.

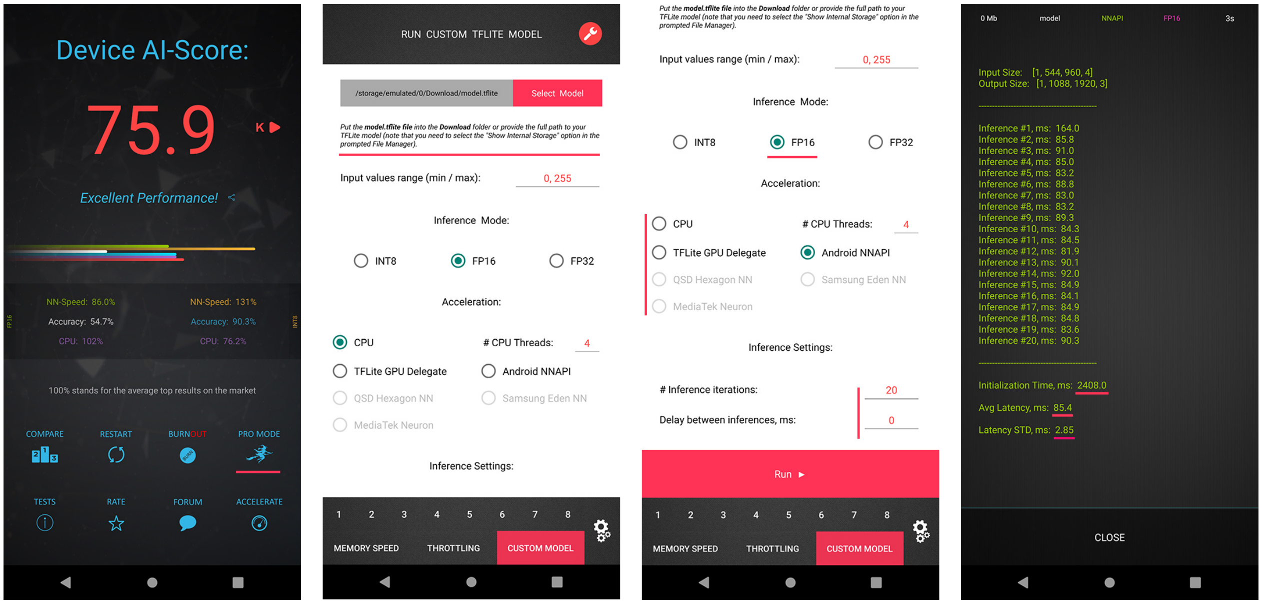

Figure 2. Loading and running custom TensorFlow Lite models with AI Benchmark application. The currently supported acceleration options include Android NNAPI, TFLite GPU, Hexagon NN, Samsung Eden and MediaTek Neuron delegates as well as CPU inference through TFLite or XNNPACK backends. The latest app version can be downloaded at https://ai-benchmark.com/download.

2.1. Dataset

To handle the problem of image translation from the original RAW photos captured with modern mobile camera sensors to superior quality images achieved by professional full-frame or medium format cameras, a large-scale real-world dataset containing RAW-RGB image pairs was collected. The dataset consists of photos taken in the wild synchronously by a 102-MP Fujifilm medium format camera and the Sony IMX586 Quad Bayer mobile sensor shooting RAW images. The photos were taken during the daytime in a wide variety of places and various illumination and weather conditions. The photos were captured in automatic mode, and the default settings were used for both cameras throughout the whole collection procedure. An example set of collected images can be seen in Fig. 1.

Since the captured RAW-RGB image pairs are not perfectly aligned, they were matched using an advanced dense correspondence algorithm [49], and then smaller patches of size 256×256 px were extracted. The participants were provided with around 24 thousand training RAW-RGB image pairs (of size 256×256×1 and 256×256×3, respectively). It should be mentioned that all alignment operations were performed on RGB Fujifilm images only, therefore RAW photos from the Sony sensor remained unmodified. A comprehensive tutorial demonstrating how to work with the data and how to train a baseline PUNET model on the provided images was additionally released to the participants: https://github.com/MediaTek-NeuroPilot/mai21-learned-smartphone-isp.

2.2. Local Runtime Evaluation

When developing AI solutions for mobile devices, it is vital to be able to test the designed models and debug all emerging issues locally on available devices. For this, the participants were provided with the *AI Benchmark* application [24, 27] that allows to load any custom TensorFlow Lite model and run it on any Android device with all supported acceleration options. This tool contains the latest versions of *Android NNAPI, TFLite GPU, Hexagon NN, Samsung Eden* and *MediaTek Neuron* delegates, therefore supporting all current mobile platforms and providing the users with the ability to execute neural networks on smartphone NPUs, APUs, DSPs, GPUs and CPUs.

To load and run a custom TensorFlow Lite model, one needs to follow the next steps:

1. Download AI Benchmark from the official website[1] or from the Google Play[2] and run its standard tests.
2. After the end of the tests, enter the *PRO Mode* and select the *Custom Model* tab there.
3. Rename the exported TFLite model to *model.tflite* and put it into the *Download* folder of the device.
4. Select mode type *(INT8, FP16, or FP32)*, the desired acceleration/inference options and run the model.

These steps are also illustrated in Fig. 2.

[1] https://ai-benchmark.com/download
[2] https://play.google.com/store/apps/details?id=org.benchmark.demo

Team	Author	Framework	Model Size, KB	PSNR↑	SSIM↑	Runtime, ms ↓	Final Score
dh_isp	xushusong001	PyTorch / TensorFlow	21	23.2	0.8467	**61**	**25.98**
AIISP	AIISP	TensorFlow	123	**23.73**	0.8487	90.8	**25.91**
Tuned U-Net	Baseline	PyTorch / TensorFlow	13313	23.30	0.8395	78	25.74
ENERZAi Research	Minsu.Kwon	TensorFlow	9	22.97	0.8392	65	25.67
isp_forever	LearnedSmartphoneISP	PyTorch	175	22.78	0.8472	77	25.24
NOAHTCV	noahtcv	TensorFlow	244	23.08	0.8237	94.5	25.19
ACVLab	jesse1029	TensorFlow	7	22.03	0.8217	76.3	24.5
CVML	vishalchudasama	TensorFlow	76	22.84	0.8379	167	23.5
ENERZAi Research *	*jaeyon*	PyTorch / TensorFlow	11	23.41	**0.8534**	231	23.39
EdS	Etienne	TensorFlow	1017	23.23	0.8481	1861	22.4

Table 1. Mobile AI 2021 smartphone ISP challenge results and final rankings. The runtime values were obtained on Full HD (1920×1088) images. Teams *dh_isp* and *AIISP* are the challenge winners. *Tuned U-Net* corresponds to a baseline U-Net model [44] tuned specifically for the target Dimensity 1000+ platform. Team *EdS* was ranked second in the PIRM 2018 Image Enhancement Challenge [29], its results on this task are provided for the reference. * The second solution from *ENERZAi Research* team did not participate in the official test phase, its scores are shown for general information only.

2.3. Runtime Evaluation on the Target Platform

In this challenge, we use the *MediaTek Dimensity 1000+* SoC as our target runtime evaluation platform. This chipset contains a powerful APU [36] capable of accelerating floating point, INT16 and INT8 models, being ranked first by AI Benchmark at the time of its release [2]. It should be mentioned that FP16/INT16 inference support is essential for this task as raw Bayer data has a high dynamic range (10- to 14-bit images depending on the camera sensor model).

Within the challenge, the participants were able to upload their TFLite models to the runtime validation server connected to a real device and get instantaneous feedback: the runtime of their solution on the Dimensity 1000+ APU or a detailed error log if the model contains some incompatible operations (ops). The models were parsed and accelerated using MediaTek Neuron delegate[3]. The same setup was also used for the final runtime evaluation. The participants were additionally provided with a detailed model optimization guideline demonstrating the restrictions and the most efficient setups for each supported TFLite op.

2.4. Challenge Phases

The challenge consisted of the following phases:

I. *Development:* the participants get access to the data and AI Benchmark app, and are able to train the models and evaluate their runtime locally;

II. *Validation:* the participants can upload their models to the remote server to check the fidelity scores on the validation dataset, to get the runtime on the target platform, and to compare their results on the validation leaderboard;

III. *Testing:* the participants submit their final results, codes, TensorFlow Lite models, and factsheets.

[3]https://github.com/MediaTek-NeuroPilot/tflite-neuron-delegate

2.5. Scoring System

All solutions were evaluated using the following metrics:

- Peak Signal-to-Noise Ratio (PSNR) measuring fidelity score,
- Structural Similarity Index Measure (SSIM), a proxy for perceptual score,
- The runtime on the target Dimensity 1000+ platform.

The score of each final submission was evaluated based on the next formula:

$$\text{Final Score} = \text{PSNR} + \alpha \cdot (0.2 - clip(\text{runtime})),$$

where:

$$\alpha = \begin{cases} 20, & \text{if runtime} \le 0.2 \\ 0.5, & \text{otherwise} \end{cases},$$

$$clip = min(max(\text{runtime}, 0.03), 5).$$

During the final challenge phase, the participants did not have access to the test dataset. Instead, they had to submit their final TensorFlow Lite models that were subsequently used by the challenge organizers to check both the runtime and the fidelity results of each submission under identical conditions. This approach solved all the issues related to model overfitting, reproducibility of the results, and consistency of the obtained runtime/accuracy values.

3. Challenge Results

From the above 190 registered participants, 9 teams entered the final phase and submitted valid results, TFLite models, codes, executables, and factsheets. Table 1 summarizes the final challenge results and reports PSNR, SSIM, and runtime numbers for each submitted solution on the final test dataset on the target evaluation platform. The proposed methods are described in Section 4, and the team members and affiliations are listed in Appendix A.

3.1. Results and Discussion

All submitted solutions demonstrated a very high efficiency: the majority of models are able to process one Full HD (1920×1088 px) image under 100 ms on the target MediaTek APU. Teams *dh_isp* and *AIISP* are the winners of this challenge, achieving the best runtime and fidelity results on this task, respectively. These solutions are following two absolutely different approaches. *dh_isp* is using an extremely shallow 3-layer *FSRCNN* [10]-inspired model with one pixel-shuffle block, and is processing the input image at the original scale. The size of this model is only 21 KB, and it is able to achieve an impressive 16 FPS on the Dimensity 1000+ SoC. In contrast, the solution proposed by *AIISP* is downsampling the input data and applying a number of convolutional layers and several sophisticated attention blocks to get high fidelity results, while also demonstrating more than 10 FPS on the target platform.

Tuned U-Net is a solid U-Net baseline with several hardware-driven adaptations designed to demonstrate the performance that can be achieved with a common APU-aware tuned deep learning architecture. It is also showing that the weight and the number of layers do not necessarily play key roles in model efficiency if the majority of processing is happening at lower scales. While its size is more than 1000 times larger compared to the model proposed by *ENERZAi Research*, it demonstrates comparable runtime results on the same hardware. The tremendous model size reduction in the latter case was achieved by using an efficient knowledge transfer approach consisting of the joint training of two (tiny and large) models sharing the same feature extraction block. Another interesting approach was proposed by *NOAHTCV* which model is processing chroma and texture information separately.

In the final ranking table, one can also find the results of team *EdS* that was ranked second in the *PIRM 2018 Image Enhancement Challenge* [29]. This model was deliberately not optimized for the target platform to demonstrate the importance of such fine-tuning. While its fidelity scores are still high, showing the second-best PSNR result, it requires almost 2 seconds to process one image on the Dimensity 1000+ platform. The reason for this is quite straightforward: it is using several ops not yet adequately supported by NNAPI (despite claimed as officially supported). By removing or replacing these ops, the runtime of this model improves by more than 10 times, while the corresponding difference on desktop CPUs / GPUs is less than 10%. This example explicitly shows that the runtime values obtained on common deep learning hardware are not representative when it comes to model deployment on mobile AI silicon: even solutions that might seem to be very efficient can struggle significantly due to the specific constraints of smartphone AI acceleration hardware and frameworks. This makes deep learning development for mobile devices so challenging, though the results obtained in this competition demonstrate that one can get a very efficient model when taking the above aspects into account.

4. Challenge Methods

This section describes solutions submitted by all teams participating in the final stage of the MAI 2021 Learned Smartphone ISP challenge.

4.1. dh_isp

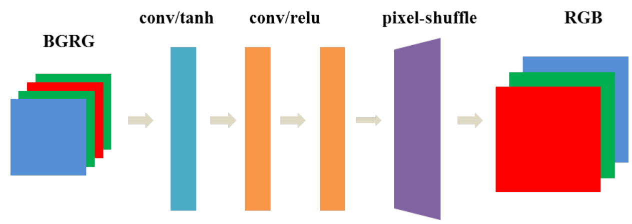

Figure 3. Smallnet architecture proposed by team dh_isp.

Team dh_isp proposed a very compact Smallnet architecture illustrated in Fig. 3 that consists of three convolutional and one pixel-shuffle layer. Each convolutional layer is using 3×3 kernels and has 16, 16, and 12 channels, respectively. *Tanh* activation function is used after the first layer, while the other ones are followed by the *ReLU* function. The authors especially emphasize the role of the *Tanh* and pixel-shuffle ops in getting high fidelity results on this task.

The model was first trained with L_1 loss only, and then fine-tuned with a combination of L_1 and perceptual-based *VGG-19* loss functions. The network parameters were optimized with the Adam algorithm using a batch size of 4 and a learning rate of $1e-4$ that was decreased within the training.

4.2. AIISP

The authors proposed a Channel Spacial Attention Network (Fig. 4) that achieved the best fidelity results in this challenge. The architecture of this model consists of the following three main parts. In the first part, two convolutional blocks with *ReLU* activations are used to perform feature extraction and downsize the input RAW data. After that, a series of processing blocks are cascaded. The middle double attention modules (DAM) with skip connections are mainly designed to enhance the spatial dependencies and to highlight the prominent objects in the feature maps. These skip connections are used not only to avoid the vanishing gradient problem but also to keep the similarities between the learned feature maps from different blocks. The last part of the network uses transposed convolution and depth-to-space modules to upscale the feature maps to their target size. Finally, a conventional convolution followed by the sigmoid activation function restores the output RGB image.

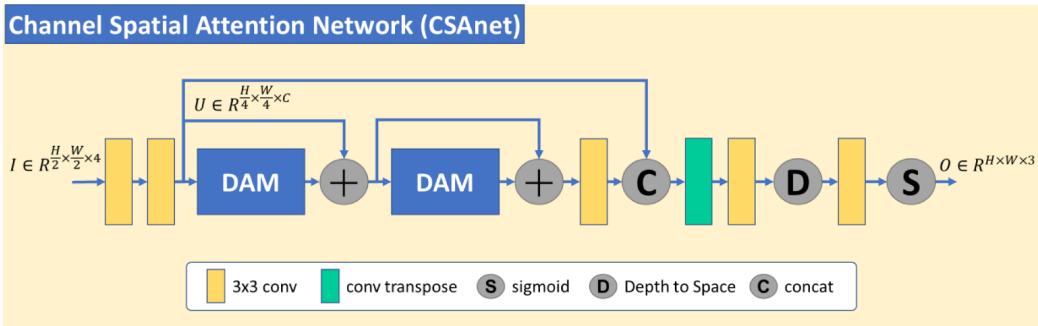

Figure 4. CSANet model from team AIISP with several attention blocks.

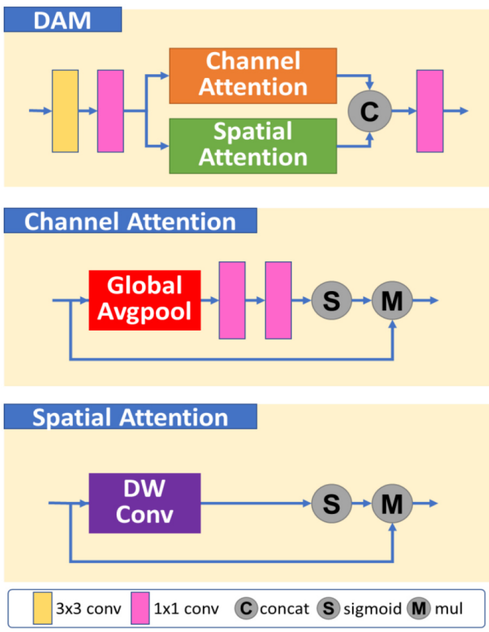

Figure 5. The structure of double attention module (DAM), spatial attention and channel attention blocks.

The sub-network structure of DAM is shown in Fig. 5. Given the feature maps obtained after applying two convolutions, DAM performs feature recalibration by using two attention mechanisms: spatial attention (SA) and channel attention (CA). The results of these concatenated attentions are then followed by a $1{\times}1$ convolutional layer to yield an adaptive feature refinement. The spatial attention module is designed to learn spatial dependencies in the feature maps. In order to have a distant vision over these maps, a depth-wise dilated convolution with a $5{\times}5$ kernel is used to extract the information. The output of this module is multiplied with the corresponding input feature map to get the final result. Channel attention block uses squeeze-and-excite operations to learn the inter-channel relationship between the feature maps. The squeeze operation is implemented by computing the mean values over individual feature maps. The excite operation is composed of two $1{\times}1$ convolution layers with different channel sizes and activations (*ReLU* and sigmoid, respectively) and re-calibrates the squeeze output. The output of the module is also obtained by elemental-wise multiplication of the input feature maps and the calibrated descriptor. A more detailed description of the CSANet architecture is provided in [12].

The model is trained with a combination of the *Charbonnier* loss function (used to approximate L_1 loss), perceptual *VGG-19* and *SSIM* losses. The weights of the model are optimized using Adam for 100K iterations with a learning rate of $5e-4$ decreased to $1e-5$ throughout the training. A batch size of 100 was used, the training data was augmented by random horizontal flipping.

4.3. Tuned U-Net Baseline

A U-Net [44] based model was developed to get an effective baseline for this challenge. This model follows the standard U-Net architecture with skip connections, and introduces several hardware-specific adaptations for the target platform such as a reduced number of feature maps, modified convolutional filter sizes, and activation functions, and additional skip connections used to maintain a reasonable accuracy. The model was trained with a combination of *MSE* and *SSIM* loss functions using Adam optimizer with a learning rate of $1e-4$.

4.4. ENERZAi Research

The solution proposed by ENERZAi Research is inspired by the *Once-for-All* approach [3] and consists of two models: one super-network and one sub-network. They both share the same Dense-Net-like module, and the difference comes from their top layers: the sub-network has one deconvolution, convolution, and sigmoid layers, while the super-network additionally contains several residual dense blocks as shown in Fig. 6. Both models are first trained jointly using a combination of the *Charbonnier* and *MS-SSIM* loss functions. The super-network is then detached after the PSNR score goes above a predefined threshold, and the sub-net is further fine-tuned alone. The model was trained using Adam optimizer with a batch size of 4 and a

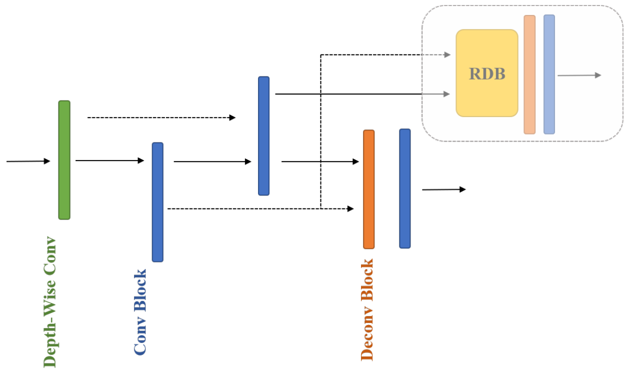

Figure 6. The model architecture proposed by ENERZAi Research team. Semitransparent residual dense block belongs to the supernetwork and is detached after training.

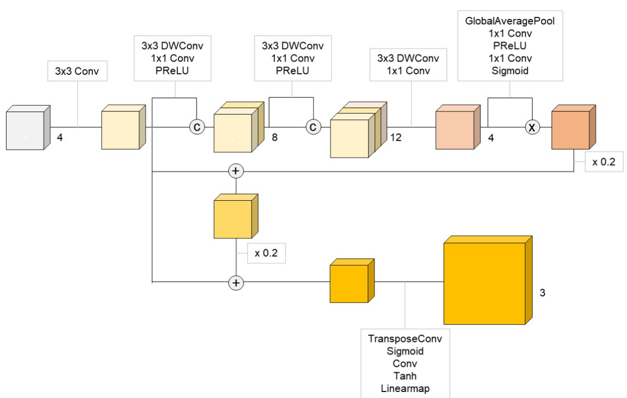

Figure 7. ESRGAN-based architecture with additional attention block proposed by ENERZAi Research.

learning rate of $1e-3$.

The second model proposed by this team (which did not officially participate in the final test phase) is demonstrated in Fig. 7. It follows the ESRGAN architecture [53] and has a shallow feature extractor and several DenseNet-based residual blocks with separable convolutions followed by a transpose convolution layer. The authors also used an additional channel attention block to boost the fidelity results at the expense of a very slight speed degradation. To choose the most appropriate activation function, the authors applied NAS technique that resulted in selecting the *PReLU* activations. The model was trained with a combination of the *MS-SSIM* and L_1 losses. It should be also mentioned that the original model was implemented and trained using PyTorch. To avoid the problems related to inefficient PyTorch-to-TensorFlow conversion, the authors developed their own scripts translating the original model architecture and weights to TensorFlow, and then converted the obtained network to TFLite.

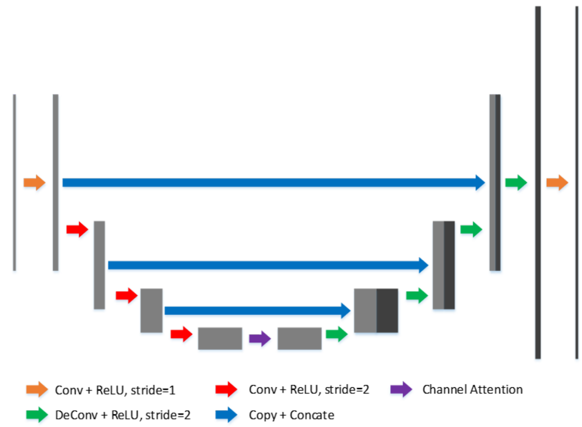

Figure 8. U-Net based network with a channel attention module from isp_forever.

4.5. isp_forever

Team isp_forever proposed another truncated U-Net based model for this task that is demonstrated in Fig. 8. The authors augmented their network with a channel attention module and trained the entire model with a combination of L_1, *SSIM*, and *VGG*-based losses using Adam optimizer with a learning rate of $1e-4$ and a batch size of 16.

4.6. NOAHTCV

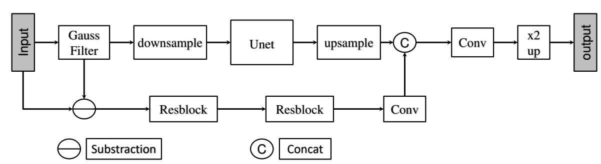

Figure 9. Network architecture proposed by team NOAHTCV and processing chroma and texture parts separately.

This team proposed to decompose the input image into two parts (Fig. 9): chroma part that contains color information, and texture part that includes high-frequency details. The network processes these two parts in separate paths: the first one has a U-Net like architecture and performs patch-level information extraction and tone-mapping, while the second one applies residual blocks for texture enhancement. The outputs from both paths are fused at the end and then upsampled to get the final result. L_1 and *SSIM* losses were used to train the network, its parameters were initialized with Xavier and optimized using Adam with a learning rate of $1e-4$.

4.7. ACVLab

ACVLab proposed a very compact CNN model with a local fusion block. This block consists of two parts: multiple stacked adaptive weight residual units, termed as RRDB (residual in residual dense block), and a local residual fusion unit (LRFU). The RRDB module can improve the information flow and gradients, while the LRFU module can effectively fuse multi-level residual information in the local fusion block. The model was trained using *VGG*-based, *SSIM*, and *Smooth* L_1 losses.

4.8. CVML

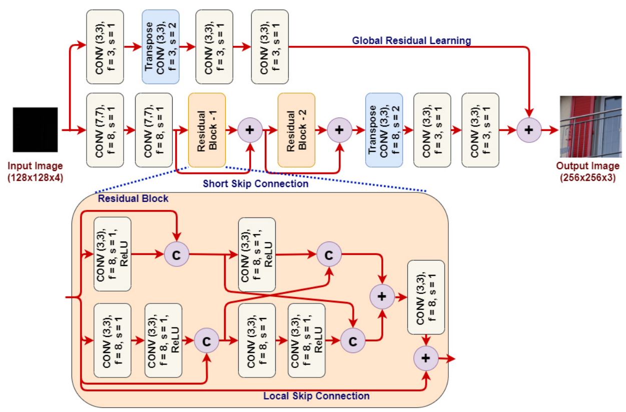

Figure 10. CNN architecture proposed by CVML team.

Figure 10 demonstrates the model proposed by team CVML. This architecture is using residual blocks to extracts a rich set of features from the input data. Transposed convolution layer is used to upsample the final feature maps to the target resolution. To stabilize the training process, a global residual learning strategy was employed that also helped to reduce the color shift effect. The model was trained to minimize the combination of L_1 and *SSIM* losses and was optimized using Adam with a learning rate of $1e-4$ for 1M iterations.

4.9. EdS

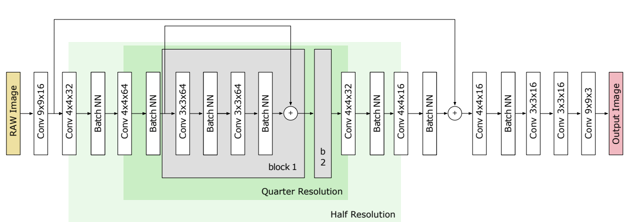

Figure 11. Residual network proposed by team EdS.

EdS proposed a ResNet-based architecture shown in Fig. 11 that was derived from [16]. The main difference consists in using two 4×4 convolutional layers with stride 2 for going into lower-dimensional space, and additional skip connections for faster training. The network was trained for 33K iterations using the same losses and setup as in [8].

5. Additional Literature

An overview of the past challenges on mobile-related tasks together with the proposed solutions can be found in the following papers:

- Learned End-to-End ISP: [26, 30]
- Perceptual Image Enhancement: [29, 23]
- Bokeh Effect Rendering: [21, 28]
- Image Super-Resolution: [29, 42, 4, 48]

Acknowledgements

We thank MediaTek Inc., AI Witchlabs and ETH Zurich (Computer Vision Lab), the organizers and sponsors of this Mobile AI 2021 challenge.

A. Teams and Affiliations

Mobile AI 2021 Team

Title:
Mobile AI 2021 Learned Smartphone ISP Challenge
Members:
Andrey Ignatov[1,3] *(andrey@vision.ee.ethz.ch)*, Cheng-Ming Chiang[2] *(jimmy.chiang@mediatek.com)*, Hsien-Kai Kuo[2] *(hsienkai.kuo@mediatek.com)*, Anastasia Sycheva[1], Radu Timofte[1,3] *(radu.timofte@vision.ee.ethz.ch)*, Min-Hung Chen[2] *(mh.chen@mediatek.com)*, Man-Yu Lee[2] *(my.lee@mediatek.com)*, Yu-Syuan Xu[2] *(yu-syuan.xu @mediatek.com)*, Yu Tseng[2] *(ulia.tseng@mediatek.com)*
Affiliations:
[1] Computer Vision Lab, ETH Zurich, Switzerland
[2] MediaTek Inc., Taiwan
[3] AI Witchlabs, Switzerland

dh_isp

Title:
Smallnet for An End-to-end ISP Pipeline
Members:
Shusong Xu (13821177832@163.com), Jin Guo
Affiliations:
Dahua Technology, China

AIISP

Title:
CSAnet: High Speed Channel Spacial Attention Network for Mobile ISP [12]
Members:
Chao-Hung Chen (ZHChen@itri.org.tw), Ming-Chun

Hsyu, Wen-Chia Tsai, Chao-Wei Chen
Affiliations:
Industrial Technology Research Institute (ITRI), Taiwan

U-Net Baseline

Title:
Platform-aware Tuned U-Net Model
Members:
Grigory Malivenko (grigory.malivenko@gmail.com)
Affiliations:
Out of competition participation

ENERZAi Research

Title:
Learning DenseNet by Shrinking Large Network for ISP
Members:
Minsu Kwon (minsu.kwon@enerzai.com), Myungje Lee, Jaeyoon Yoo, Changbeom Kang, Shinjo Wang
Affiliations:
ENERZAi, Seoul, Korea
enerzai.com

isp_forever

Title:
A Lightweight U-Net Model for Learned Smartphone ISP
Members:
Zheng Shaolong (master5158@163.com), Hao Dejun, Xie Fen, Feng Zhuang
Affiliations:
Dahua Technology, China

NOAHTCV

Title:
Decomposition Network for AutoISP
Members:
Yipeng Ma (mayipeng@huawei.com), Jingyang Peng, Tao Wang, Fenglong Song
Affiliations:
Huawei Noah's Ark Lab, China

ACVLab

Title:
Lightweight Residual Learning Network for Learned ISP
Members:
Chih-Chung Hsu (cchsu@gs.ncku.edu.tw), Kwan-Lin Chen, Kwan-Lin Chen, Mei-Hsuang Wu
Affiliations:
Institute of Data Science, National Cheng Kung University, Taiwan

CVML

Title:
Mobile compatible Convolution Neural Network for Learned Smartphone ISP
Members:
Vishal Chudasama (vishalmchudasama88@gmail.com), Kalpesh Prajapati, Heena Patel, Anjali Sarvaiya, Kishor Upla, Kiran Raja, Raghavendra Ramachandra, Christoph Busch
Affiliations:
Sardar Vallabhbhai National Institute of Technology, India

EdS

Title:
Fast learned ISP using Resnet Variant with Transposed Convolutions
Members:
Etienne de Stoutz (edestoutz@gmail.com)
Affiliations:
ETH Zurich, Switzerland

References

[1] Android Neural Networks API. https://developer.android.com/ndk/guides/neuralnetworks. 2

[2] AI Benchmark Archive. https://bit.ly/32wykta. 4

[3] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019. 6

[4] Jianrui Cai, Shuhang Gu, Radu Timofte, and Lei Zhang. Ntire 2019 challenge on real image super-resolution: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 8

[5] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13169–13178, 2020. 2

[6] Cheng-Ming Chiang, Yu Tseng, Yu-Syuan Xu, Hsien-Kai Kuo, Yi-Min Tsai, Guan-Yu Chen, Koan-Sin Tan, Wei-Ting Wang, Yu-Chieh Lin, Shou-Yao Roy Tseng, et al. Deploying image deblurring across mobile devices: A perspective of quality and latency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 502–503, 2020. 2

[7] Linhui Dai, Xiaohong Liu, Chengqi Li, and Jun Chen. Awnet: Attentive wavelet network for image isp. *arXiv preprint arXiv:2008.09228*, 2020. 1

[8] Etienne de Stoutz, Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, and Luc Van Gool. Fast perceptual image enhancement. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 1, 8

[9] TensorFlow Lite delegates. https://www.tensorflow.org/lite/performance/delegates. 2

[10] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European conference on computer vision*, pages 391–407. Springer, 2016. 5

[11] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019. 2

[12] Ming-Chun Hsyu, Chih-Wei Liu, Chao-Hung Chen, Chao-Wei Chen, and Wen-Chia Tsai. Csanet: High speed channel spatial attention network for mobile isp. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2021. 6, 8

[13] Jie Huang, Pengfei Zhu, Mingrui Geng, Jiewen Ran, Xingguang Zhou, Chen Xing, Pengfei Wan, and Xiangyang Ji. Range scaling global u-net for perceptual image enhancement on mobile devices. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 1

[14] Zheng Hui, Xiumei Wang, Lirui Deng, and Xinbo Gao. Perception-preserving convolutional networks for image enhancement on smartphones. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 1

[15] Andrey Ignatov, Kim Byeoung-su, and Radu Timofte. Fast camera image denoising on mobile gpus with deep learning, mobile ai 2021 challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2021. 2

[16] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Dslr-quality photos on mobile devices with deep convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3277–3285, 2017. 1, 8

[17] Andrey Ignatov, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Wespe: weakly supervised photo enhancer for digital cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 691–700, 2018. 1

[18] Andrey Ignatov, Grigory Malivenko, David Plowman, Samarth Shukla, and Radu Timofte. Fast and accurate single-image depth estimation on mobile devices, mobile ai 2021 challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2021. 2

[19] Andrey Ignatov, Grigory Malivenko, and Radu Timofte. Fast and accurate quantized camera scene detection on smartphones, mobile ai 2021 challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2021. 2

[20] Andrey Ignatov, Jagruti Patel, and Radu Timofte. Rendering natural camera bokeh effect with deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 418–419, 2020. 2

[21] Andrey Ignatov, Jagruti Patel, Radu Timofte, Bolun Zheng, Xin Ye, Li Huang, Xiang Tian, Saikat Dutta, Kuldeep Purohit, Praveen Kandula, et al. Aim 2019 challenge on bokeh effect synthesis: Methods and results. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3591–3598. IEEE, 2019. 8

[22] Andrey Ignatov, Andres Romero, Heewon Kim, and Radu Timofte. Real-time video super-resolution on smartphones with deep learning, mobile ai 2021 challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2021. 2

[23] Andrey Ignatov and Radu Timofte. Ntire 2019 challenge on image enhancement: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1, 8

[24] Andrey Ignatov, Radu Timofte, William Chou, Ke Wang, Max Wu, Tim Hartley, and Luc Van Gool. Ai benchmark: Running deep neural networks on android smartphones. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 2, 3

[25] Andrey Ignatov, Radu Timofte, Maurizio Denna, and Abdel Younes. Real-time quantized image super-resolution on mobile npus, mobile ai 2021 challenge: Report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2021. 2

[26] Andrey Ignatov, Radu Timofte, Sung-Jea Ko, Seung-Wook Kim, Kwang-Hyun Uhm, Seo-Won Ji, Sung-Jin Cho, Jun-Pyo Hong, Kangfu Mei, Juncheng Li, et al. Aim 2019 challenge on raw to rgb mapping: Methods and results. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3584–3590. IEEE, 2019. 1, 8

[27] Andrey Ignatov, Radu Timofte, Andrei Kulik, Seungsoo Yang, Ke Wang, Felix Baum, Max Wu, Lirong Xu, and Luc Van Gool. Ai benchmark: All about deep learning on smartphones in 2019. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3617–3635. IEEE, 2019. 2, 3

[28] Andrey Ignatov, Radu Timofte, Ming Qian, Congyu Qiao, Jiamin Lin, Zhenyu Guo, Chenghua Li, Cong Leng, Jian Cheng, Juewen Peng, et al. Aim 2020 challenge on rendering realistic bokeh. In *European Conference on Computer Vision*, pages 213–228. Springer, 2020. 8

[29] Andrey Ignatov, Radu Timofte, Thang Van Vu, Tung Minh Luu, Trung X Pham, Cao Van Nguyen, Yongwoo Kim, Jae-Seok Choi, Munchurl Kim, Jie Huang, et al. Pirm challenge on perceptual image enhancement on smartphones: Report. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 1, 4, 5, 8

[30] Andrey Ignatov, Radu Timofte, Zhilu Zhang, Ming Liu, Haolin Wang, Wangmeng Zuo, Jiawei Zhang, Ruimao

Zhang, Zhanglin Peng, Sijie Ren, et al. Aim 2020 challenge on learned image signal processing pipeline. *arXiv preprint arXiv:2011.04994*, 2020. 1, 8

[31] Andrey Ignatov, Luc Van Gool, and Radu Timofte. Replacing mobile camera isp with a single deep learning model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 536–537, 2020. 1

[32] Dmitry Ignatov and Andrey Ignatov. Controlling information capacity of binary neural network. *Pattern Recognition Letters*, 138:276–281, 2020. 2

[33] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018. 2

[34] Sambhav R Jain, Albert Gural, Michael Wu, and Chris H Dick. Trained quantization thresholds for accurate and efficient fixed-point inference of deep neural networks. *arXiv preprint arXiv:1903.08066*, 2019. 2

[35] Byung-Hoon Kim, Joonyoung Song, Jong Chul Ye, and Jae-Hyun Baek. Pynet-ca: enhanced pynet with channel attention for end-to-end mobile image signal processing. In *European Conference on Computer Vision*, pages 202–212. Springer, 2020. 1

[36] Yen-Lin Lee, Pei-Kuei Tsung, and Max Wu. Techology trend of edge ai. In *2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, pages 1–2. IEEE, 2018. 4

[37] Yawei Li, Shuhang Gu, Luc Van Gool, and Radu Timofte. Learning filter basis for convolutional neural network compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5623–5632, 2019. 2

[38] Hanwen Liu, Pablo Navarrete Michelini, and Dan Zhu. Deep networks for image-to-image translation with mux and demux layers. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 1

[39] Zechun Liu, Haoyuan Mu, Xiangyu Zhang, Zichao Guo, Xin Yang, Kwang-Ting Cheng, and Jian Sun. Metapruning: Meta learning for automatic neural network channel pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3296–3305, 2019. 2

[40] Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *Proceedings of the European conference on computer vision (ECCV)*, pages 722–737, 2018. 2

[41] Andreas Lugmayr, Martin Danelljan, and Radu Timofte. Unsupervised learning for real-world super-resolution. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3408–3416. IEEE, 2019. 1

[42] Andreas Lugmayr, Martin Danelljan, and Radu Timofte. Ntire 2020 challenge on real-world image super-resolution: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 494–495, 2020. 8

[43] Anton Obukhov, Maxim Rakhuba, Stamatios Georgoulis, Menelaos Kanakis, Dengxin Dai, and Luc Van Gool. T-basis: a compact representation for neural networks. In *International Conference on Machine Learning*, pages 7392–7404. PMLR, 2020. 2

[44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4, 6

[45] Jose Ivson S Silva, Gabriel G Carvalho, Marcel Santana Santos, Diego JC Santiago, Lucas Pontes de Albuquerque, Jorge F Puig Battle, Gabriel M da Costa, and Tsang Ing Ren. A deep learning approach to mobile camera image signal processing. In *Anais Estendidos do XXXIII Conference on Graphics, Patterns and Images*, pages 225–231. SBC, 2020. 1

[46] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019. 2

[47] TensorFlow-Lite. https://www.tensorflow.org/lite. 2

[48] Radu Timofte, Shuhang Gu, Jiqing Wu, and Luc Van Gool. Ntire 2018 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 852–863, 2018. 8

[49] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning accurate dense correspondences and when to trust them. *arXiv preprint arXiv:2101.01710*, 2021. 3

[50] Stefan Uhlich, Lukas Mauch, Fabien Cardinaux, Kazuki Yoshiyama, Javier Alonso Garcia, Stephen Tiedemann, Thomas Kemp, and Akira Nakamura. Mixed precision dnns: All you need is a good parametrization. *arXiv preprint arXiv:1905.11452*, 2019. 2

[51] Thang Vu, Cao Van Nguyen, Trung X Pham, Tung M Luu, and Chang D Yoo. Fast and efficient image quality enhancement via desubpixel convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 1

[52] Alvin Wan, Xiaoliang Dai, Peizhao Zhang, Zijian He, Yuandong Tian, Saining Xie, Bichen Wu, Matthew Yu, Tao Xu, Kan Chen, et al. Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12965–12974, 2020. 2

[53] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 7

[54] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10734–10742, 2019. 2

[55] Jiwei Yang, Xu Shen, Jun Xing, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-sheng Hua. Quantization networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7308–7316, 2019. 2