5

Image Signal Processor (ISP) Tuning using Machine Learning (ML) methods

Sepehr Bijani

NXP Semiconductors, Germany

Abstract

Image Signal Processor (ISP) is responsible for improving camera signal quality and producing high-quality images. ISP has a vast number of parameters which should be tunned based on both camera sensor and operating environment. Sensor related ISP parameters are tunned offline for each camera sensor. Parameters related to environment such as white balancing gains must be fine-tuned during the runtime. The offline phase is cumbersome and costly. At the same time, the runtime for fine-tuning should be fast and accurate. Therefore, the proposed tuning framework needs to achieve two goals at the same time: a) tune the sensor related parameters automatically in lab and b) fine-tuning ISP in the field with a runtime.

For the static parameters, a tuner finds the optimal parameters in lab condition. Tuning dynamic parameters needs a dataset with various scenes and corresponding optimal parameters. A data generation pipeline produces the dataset by running the tuner in a loop. An ML model is trained based on generated dataset as runtime for fine-tuning ISP.

Keywords: image signal processor, ISP, machine learning, ML, tuning, camera tuning.

5.1 Introduction and Background

5.1.1 Tuning problem

Improving System performance by changing bounded parameters respect to predefined Key Performance Indicator (KPI).

5.1.2 Image Signal processor (ISP)

ISP is a hardware or software component responsible for converting camera signal (Bayer Pattern Image) to perceivable image for human eye. It improves image quality by attenuating image artifacts. For achieving optimal performance in different environmental scenarios and various cameras, ISP has many parameters which should be tuned and optimized for different use cases. The tuning process is manual and costly. The effect of varying ISP parameters has a considerable impact on deep learning-based object detection systems, so having an automatic process and measuring the ISP performance will improve overall Advanced Driver Assistance Systems (ADAS) response [1].

Figure 5.1 Image Generation using ISP and Camera.

5.1.3 Mathematical Optimization Problem

Having a mathematical model for a system, the behavior of the system can be improved by defining an optimization problem as [2]:

$$\begin{array}{lll} \underset{x}{\text{minimize}} & f(x) & \\ \text{subject to} & g_i(x) \leq 0, & i = 1, \ldots, m \\ & h_j(x) = 0, & j = 1, \ldots, p \end{array}$$

Constrains

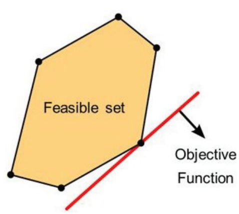


Figure 5.2 Linear Optimization Problem [3].

Where cost function f(x) models the KPI needed to improve system performance. In a tuning problem, f(x) is blocking, and high number of f(x) calls will be impractically slow. As a result, running f with a parameter set x is time consuming, and any solver for tuning problem should find optimal parameters with the smallest number of iterations. An example of optimization problem is Linear Programming (LP) where f, g, and h are linear function and define a polyhedron. Changing parameter x will move a linear function across the feasibility set as show in Figure 5.2.

5.1.4 Static and Dynamic Parameters in ISP

Different algorithms utilized in ISP which have individual parameters. Each algorithm tries to attenuate artifacts from a specific source:

a) Static Parameters: Algorithms responsible for improving artifacts originated from camera sensor have static parameters which should be tuned only once.
b) Dynamic Parameters: Algorithm responsible for improving artifacts due to light condition and environmental phenomena have dynamic parameters which should be updated during runtime.

The static parameters which are related to camera sensor characteristics can be tuned once for the specific camera. After tuning the ISP for the specific camera, the parameters can be fixed in configuration file for deployment.

Tuning dynamic parameters improves the image quality in different environmental conditions. The dynamic parameters should be updated during runtime to guaranty best image quality performance.

5.1.5 State of Art

The tuning process is done manually by experts. Each ISP algorithm is responsible for reducing specific artifact in image and the expert can measure the artifact intensity using a specific KPI. Then by changing the ISP parameters and try and error, the expert can find best parameters combination for specific camera sensor.

For tuning dynamic parameters, expert do the same process, but for a range of environmental conditions. That means, first a set of input ISP images (Bayer Pattern) are captured from sensor in different environmental condition, then expert should tune the ISP for each one of the input images. ISP has an algorithm for gathering the statistical data for Bayer pattern image. Having the statical data for all images and corresponding optimal ISP parameters, one can make a “decision tree” which changes the parameters on flight based on statistical data provided by ISP.

5.2 Automatic ISP Tuning

The automatic tuning process should be done for both Dynamic and Static parameters.

5.2.1 KPIs for Artifact Attenuation

For measuring ISP performance, various KPIs should be defined. Each KPI measures the intensity of a specific artifact in image. It should be noted that there is no one to one relation between artifacts and ISP algorithms. In many cases one KPI could be used for tuning multiple ISP algorithms. The list of KPIs is [4]:

Table 5.1 KPIs for Measuring Image Artifacts.

Artifact	ISP block	KPI
Noise	Noise Reduction Block(s)	PSNR
Loss of detail	Sharpness Correction	MTF50
Color Inaccuracy	Color Correction Matrix (CCM)	ΔE
Color Casting	White Balancing	ΔE

5.2.2 Static Parameters

The setup is done once in lab and a camera is attached to the capturing device. The captured Bayer pattern is fed to the ISP as input. The ISP generates an image. For measuring the performance of specific ISP algorithm in attenuating an artifact, corresponding KPI in Table 1 is used. ISP tuner can track KPI value for judging performance result of a set of parameters. ISP tuner changes the ISP parameters in multiple iterations and tries to optimize parameters based on KPI value. The process is iterative, and the iterative process is needed to be done only once for static ISP parameters as shown in Figure 5.3. The optimal value found by Tuner will be stored as fixed configuration for runtime.

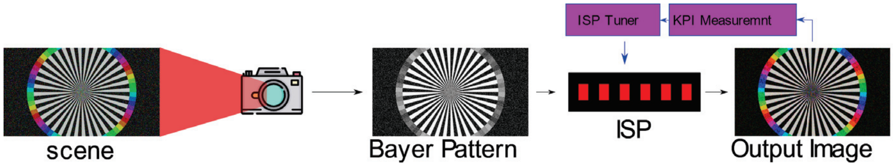


Figure 5.3 Tuning ISP Static Parameters.

5.2.3 Dynamic Parameters and Runtime

Achieving optimal performance with dynamic parameters is harder. The parameters should be re-tuned to adapt environmental effects such as light condition, temperature, etc.

There are limitations in using same iterative approach for tuning static parameters:

1. The iterative approach makes it impossible to have optimal parameters per frame or even per minute.
2. Measuring the KPI during runtime is challenging since there is no reference for the scene captured by camera.

A proposal solution is to use a machine learning model which can map image statistical data to optimal parameters. All ISPs measure image statistical data and provide it per frame. The dataset can be created by utilizing same tuning procedure mentioned for tuning statistical parameters in a loop as demonstrated in Figure 5.4.

E: All desired environmental conditions which ISP should operate in.
For e in E
{
1. Run tuner for optimizing ISP.
2. Store Optimal ISP parameters + ISP statistical data for the scene.
}

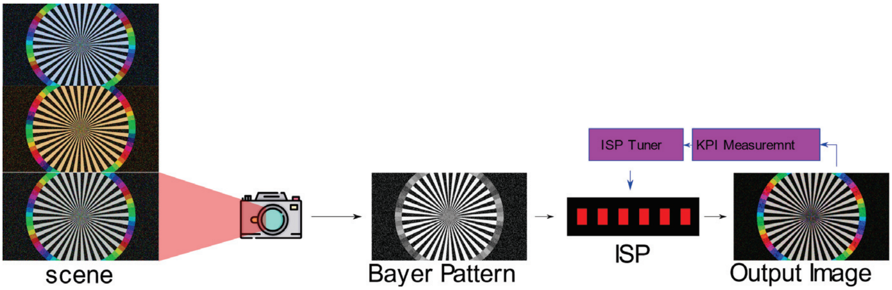


Figure 5.4 Dynamic Parameters Data Generation.

In theory, a Neural Network (NN) should be able to predict optimal values for dynamic parameters after training; however, an NN creates a high computation load during inference, so a more efficient solution is needed.

In this paper, a gradient boost model is proposed for inferencing the optimal parameters. Gradient boosting models are trainable decision trees. Unlike NN which use Directed acyclic graphs (DAGs) as underlaying data structure for training and inferencing, gradient boost (GB) utilizes trees which are simpler data structures [5]. GB models are fast to inference and has similar performance as NNs for tabular data which exactly matches the use case and dataset we have for dynamic tuning application.

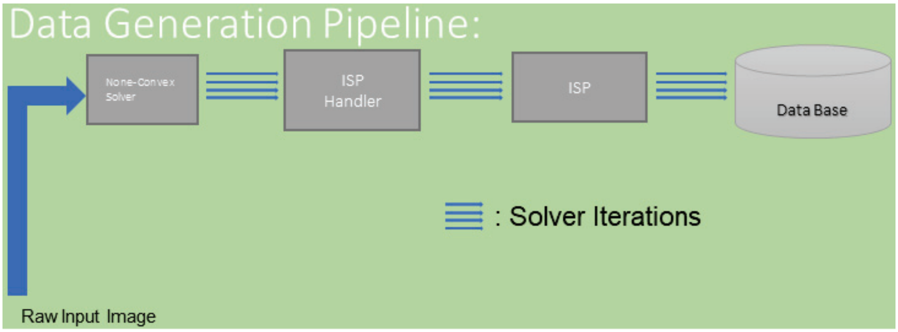


Figure 5.5 Storing Optimal Parameters and ISP Statistical Data for Training ML model.

5.2.4 Test Setup

Testing the proposed methods for tuning static and dynamic parameter are done with Solectrix SoftISP SXIVE[1]. The ISP runs on PC with dedicated graphic card. The tuner uses 16 cores CPU to speed up the process in lab for finding static parameters in ISP.

Same tuner is utilized to create a dataset of ISP statistical data and optimal parameters for ten lighting color temperature. The dataset is then used for training a GB model named XGboost as runtime. The trained model then runs on single core with minimum load on the same machine to update White Balancing parameters.

5.2.5 Results

The demo software (SW) is instantiated with init button. ISP with random parameters is run and the ISP output image is shown to the user. The user can select the boundaries of the Color Checker (CC) board inside the image then press “Tune”. The SW will crop the image to find cc board and samples

[1] sxive.com

patches from the board and shows the sampling areas to the user (Annotated CC). Then, tuning process begins. In Figure 5.6 the process for finding a better optimum point is illustrated as tuner progress. The measured ΔE for all color blocks in color checkerboard is calculated and aggregated as Mean Squared Error (MSE) of ΔE values:

$$\mathrm{MSE}(\Delta\mathrm{E}) = \sum_{k=0}^{23} \frac{(\Delta\mathrm{E}_k)^2}{24}.$$

The SW results in Figure 5.6 can be interpreted as:

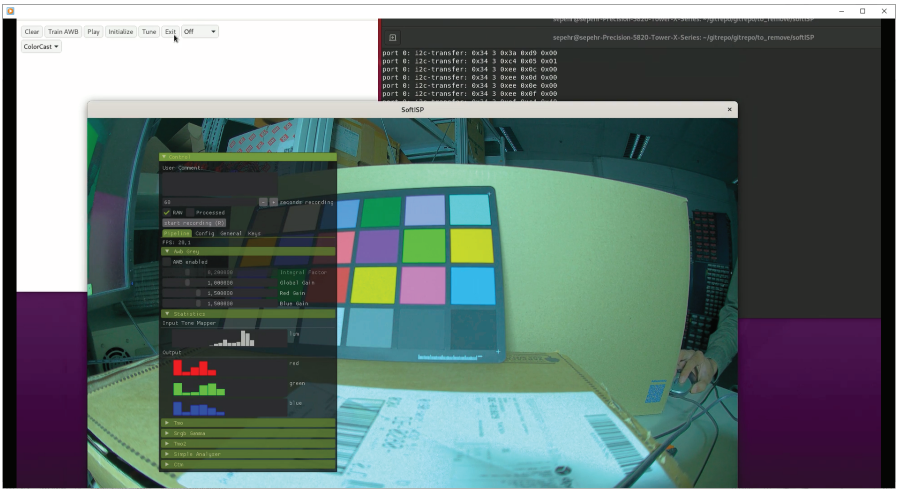
"ISP output" shows ISP output image for the current iteration, and "Best Config" shows the best result found by tuner until the current iteration. When tuner iterates over different configurations, it generates various "ISP outputs"

Iteration	MSE(ΔE)
0	1100
500	560
1000	330
1500	270
1600	226

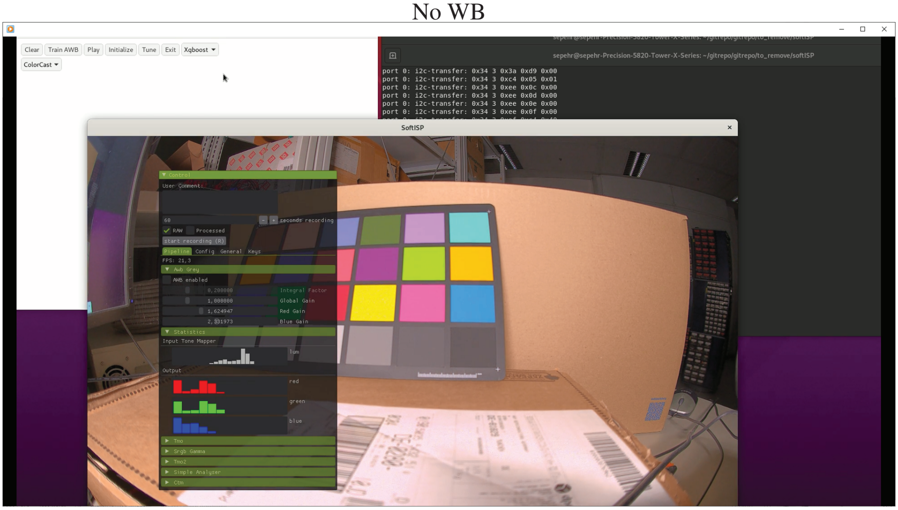
Figure 5.6 Offline Tuning Results for Color Correction Matrix Tuning.

and corresponding KPI values; Based on observed KPI value, tuner guesses a better parameter set for the next iteration. As iterations go on, tuner can find better parameter sets which results in better KPI values, so as it can be seen, the "Best Config" image is improved when tuner progresses.

White balancing (WB) is the algorithm chosen for dynamic tuning. Same tuner finds optimal parameter for various light conditions and intensities in



No WB



XGboost WB

Figure 5.7 Runtime Result of Trained XGboost WB.

lab in a loop as explained in Figure 5.4. The generated dataset maps image histogram to optimal WB configuration is used to train a XGboost model. The ISP has its own Automatic White Balancing (AWB), but we turned it off to show the effectiveness our of XGboost WB. The results are shown in Figure 5.7 without and with white balancing under blue light source.

5.3 Conclusion

Tuning ISP was conventionally a cumbersome, costly, and suboptimal task. The iterative process should have been done for combination of numerous ISPs and cameras. In pursue of a more automated solution, the proposed method tries to distinguish parameters based on static/dynamic nature. The parameters which are related to specific camera, can be tuned in lab, and the optimal parameters will be fixed as static parameters. For ISP algorithms which attenuates environmental impact on image quality, a runtime should fine-tune the ISP in the field. The runtime algorithm should be light enough to run on a restricted HW processor. For tuning both static and dynamic parameters, the paper presents a tuning framework to automate the process. A tuner finds optimal parameters for static parameters. A data generation pipeline utilizes same tuner in a loop for various environmental conditions. The generated data maps the statistical data provided by ISP to optimal parameters found by tuner. In the next steps, the trained GB model based on the generated dataset is used as a lightweighted runtime for tuning dynamic parameters in changing environmental condition.

References

[1] D. Molloy et al., “Impact of ISP Tuning on Object Detection,” Journal of Imaging, vol. 9, no. 12, pp. 260-260, Nov. 2023, doi: https://doi.org/10.3390/jimaging9120260.

[2] S. Boyd and L. Vandenberghe, “Convex Optimization,” Mar. 2004, doi: https://doi.org/10.1017/cbo9780511804441.

[3] https://en.wikipedia.org/wiki/Linear_programming.

[4] https://www.imatest.com/support/docs/23-2/colorcheck/

[5] T. Chen and C. Guestrin, “XGBoost: a Scalable Tree Boosting System,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD ’16, pp. 785-794, 2016, doi: https://doi.org/10.1145/2939672.2939785.