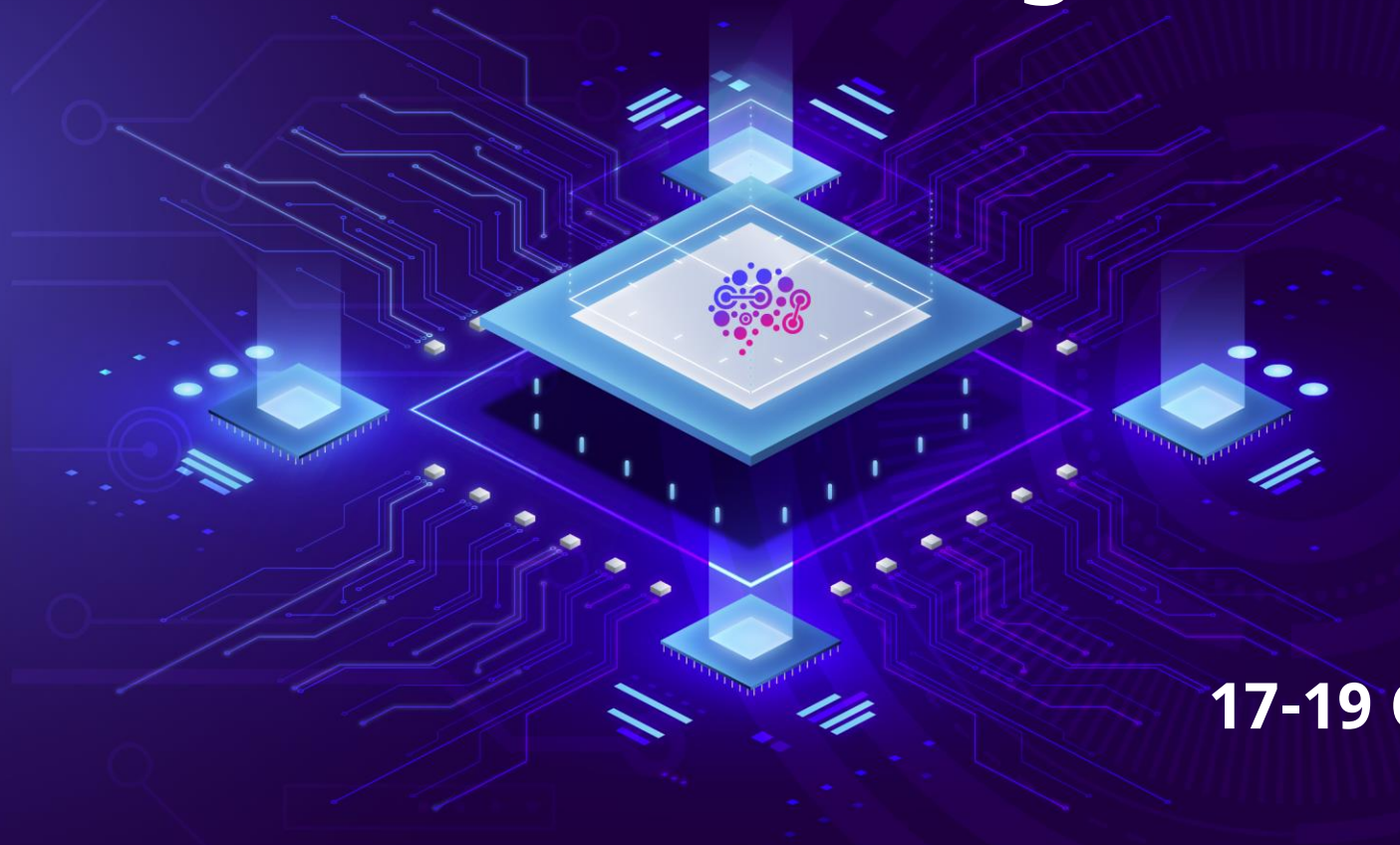
European Conference on EDGE AI Technologies and Applications - EEAI



Advancing emerging edge AI technologies and driving next generation intelligent applications.

17-19 October 2023, Athens, Greece

European Conference on EDGE AI Technologies and Applications - EEAI

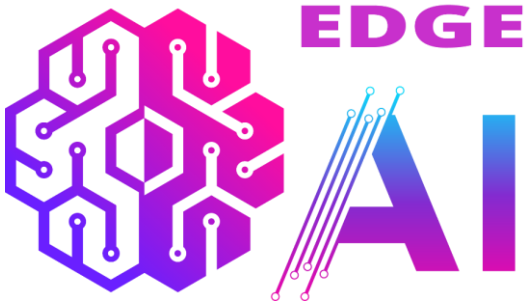


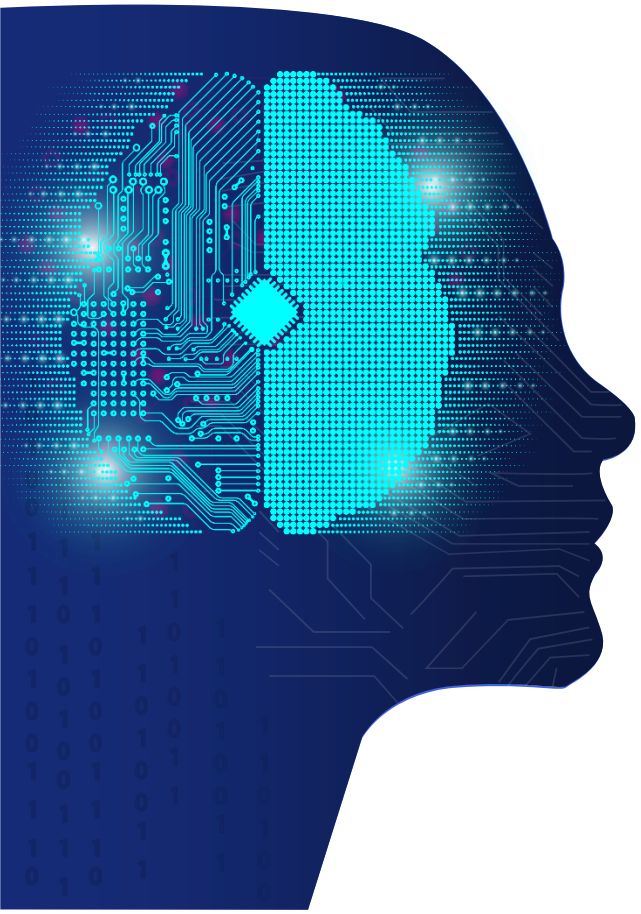
Image Signal Processor (ISP) Tunning using Machine Learning (ML) methods

Sepehr Bijani, MSc, Germany

17-19 October 2023 Athens, Greece

Copyright © 2023

Presentation Outline



- Introduction
- Static and Dynamic Parameters
- Challenges
- State-of-the-art
- Research Goal
- Methodology
- Design and Implementation
- Experimental Results
- Future Research
- Conclusion

Copyright © 2023

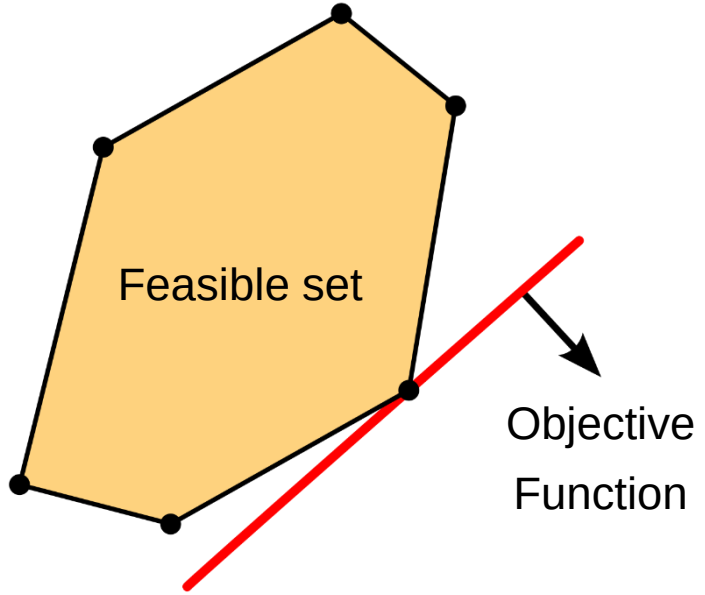
Introduction

Tuning: improve system performance respect to predefined KPI considering the constraints.

Mathematical Optimization Problem(OP):

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

→ Constrains (feasible set X)



Defining Tunning as OP:

1. **Objective measure for each image distortion: e.g., noise: PSNR color: Delta E, …**
2. **Constraints on variables. E.g., 0.0<gain<1.0**
3. **The Objective function for tunning can block!**

Copyright © 2023

Static and Dynamic Parameters in ISP

- Static Parameters: Sensor related ISP parameters should be tunned offline for each camera type. E.g., CCM
- Dynamic Parameters: related to environment such as white balancing gains should be fine-tuned in the field during the runtime.

The complexity can be divided to two parts:

- The offline phase is cumbersome and costly.
- The runtime for fine-tuning should be fast and accurate. E.g., noise reduction and AWB parameters

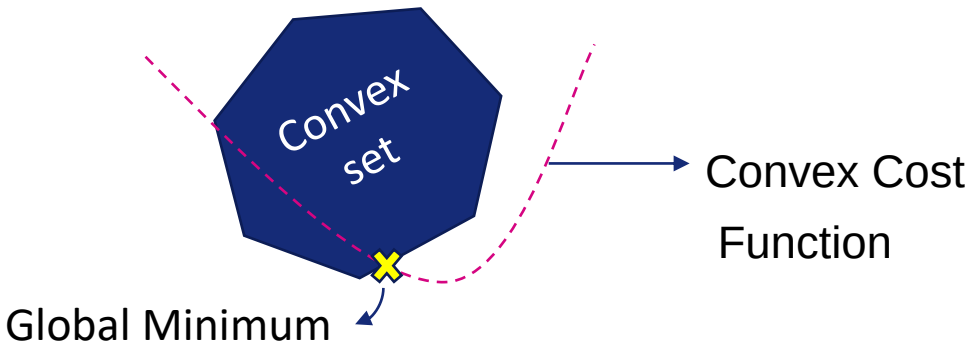
Copyright © 2023

Challenges

For solving tunning OP:

I. **OP is not Convex**

II. **OP is high dimensional (e.g., 10+)**



III. **Objective Function is Blocking: it takes a significant time to see KPI value for each parameter set.**

Runtime:

I. **Some parameters should be fine tunned. Solving OP during runtime is not possible**

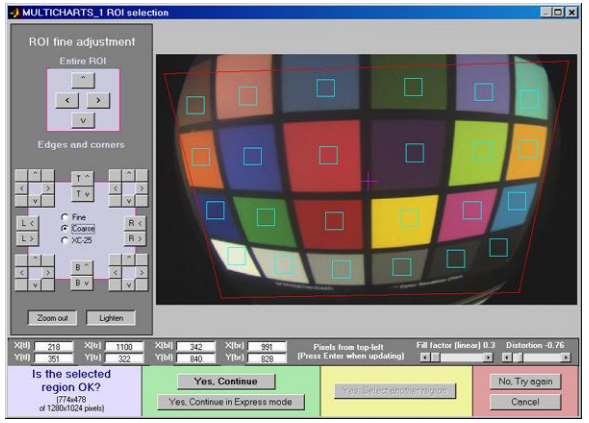
II. **The runtime should be light, we cannot use NNs as runtime since they are computationally expensive**

Copyright © 2023

State-of-the-art

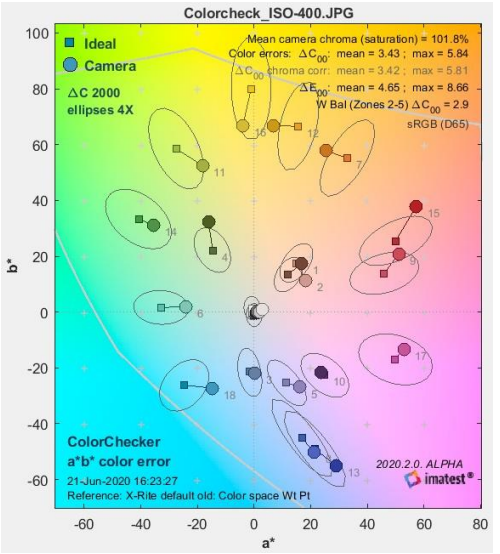
Offline tunning:

- The tunning process is done manually by experts.
- Tuning each part of ISP needs different expertise
- The process is time consuming and inaccurate



Runtime:

- Tuning for online parameters is done using small state machines which are designed and tuned for each ISP algorithm
- The state machine decides which parameter sets could be good enough based on environmental condition(statistical data provided by ISP)
- The runtime Oscillates or even fails



Copyright © 2023

Research Goal

Tuner needs to achieve two goals:

1. Tune the sensor related parameters automatically in lab
2. Create a data set for different scenarios in lab and train a Machine Learning (ML) runtime for fine-tunning ISP in the field.

For optimizer followings should be defined based on artifact we want to improve:

1. KPI which measures the artifact: such as SNR or color difference (detaE)
2. ISP algorithms which reduces artifact and parameters boundaries

Copyright © 2023

Methodology

- Automate the tuning process for static parameters:
  - Convert tuning problem to OP
  - Solve OP with the developed solver
- Create a pipeline for tuning ISP in various environmental scenarios:
  - a runtime database will be available by running tunning pipeline
  - Having the database, a Xgboost ML model will be trained
  - Xgboost models will act as trainable state machines
  - Xgboost models are light enough as a runtime
  - The database for Xgboost training will be big enough to guarantee accurate results

Copyright © 2023

Design and Implementation

- Following SW components are/will be implemented in Rust:

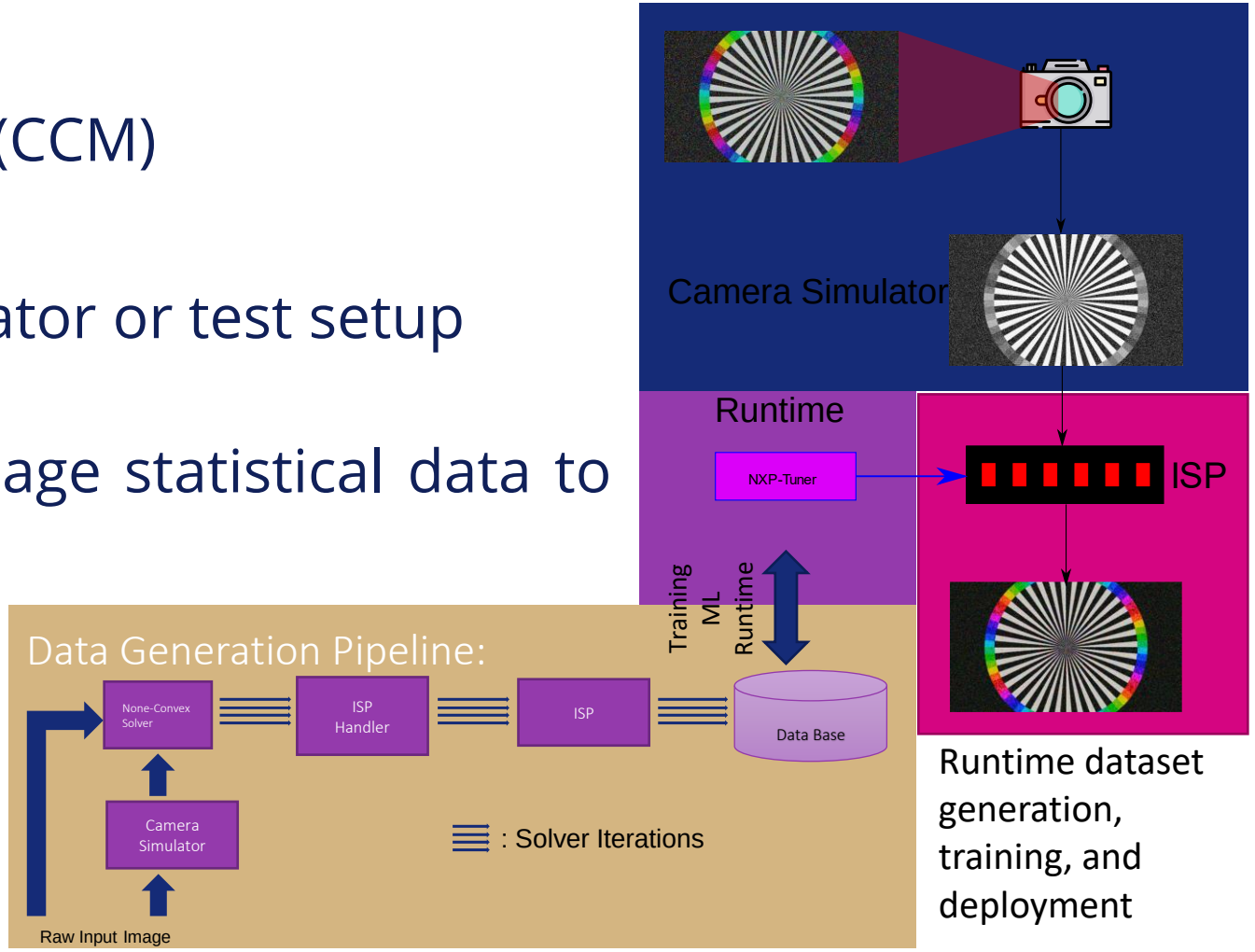
1. OP solver
2. ISP handler:
   a. Maps artifacts to ISP algorithms to be tuned
   b. Space conversion to unit space cube for OP solver
   c. Selects correct KPI for each artifact and defines feasibility set
3. Tuner:
   a. Runs solver and finds optimal parameters for static configurations
   b. Runs solver for different scene scenarios and create a tuning dataset for runtime for dynamic parameters
4. Runtime(ongoing):

   Using dataset made by tuner, an Xgboost model will be trained as a runtime

Copyright © 2023

Offline Tunning and Data Generation

- For Static Parameters:
  - Camera is fixed
  - Run OP solver single time
  - Example: Color Correction Matrix(CCM)
- Data generation for Runtime:
  - Create artifact image using simulator or test setup
  - Run solver for all artifact images
  - Create a dataset for mapping image statistical data to optimal parameter
- Training Xgboost model(s) for Runtime finetuning

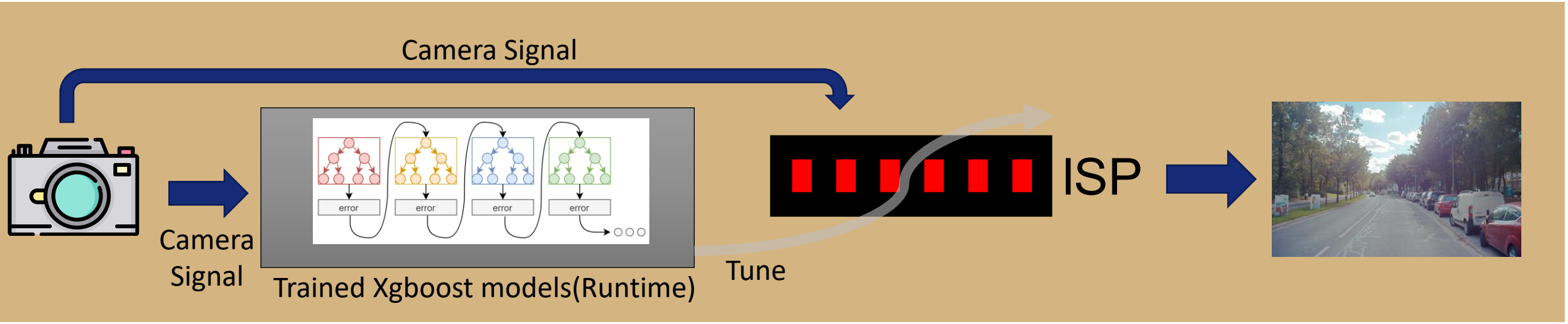


Runtime dataset generation, training, and deployment

Copyright © 2023

Runtime

- Having tuning Dataset, we will train Xgboost model for runtime:
  - Gradient boost are learnable decision trees
  - Computationally cheap as runtime



Copyright © 2023

Experimental Results

- The color correction matrix tunning process for improving precepted image color

Copyright © 2023

Future Research

- Implementation of runtime with other ML models
- Improving techniques for simulating artifacts in image
- Selecting all necessary KPI for different ISP blocks
- Improving optimization techniques for solving tuning problem

Copyright © 2023

Conclusion

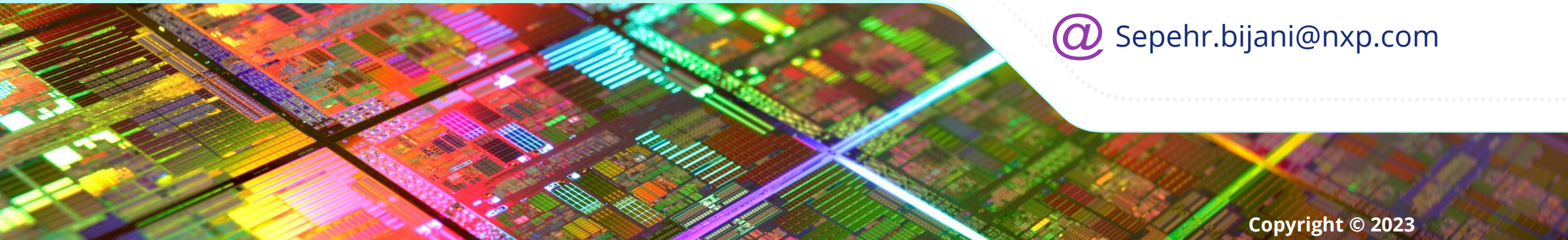
- Effect of our OP solver is proven for tuning
- Correct KPIs for different image artifacts are selected
- SW stack is implemented for automating tunning process
- SoftISP from Solectrix is used for demonstrating the results
- Unique Technologies used for this project:
  - Xgboost
  - Rust language
- Next steps:
  - Training Xgboost model for runtime
  - Implementing SW blocks for handling runtime

Copyright © 2023

EDGE AI

Thank You

For your attention



@ Sepehr.bijani@nxp.com

Copyright © 2023