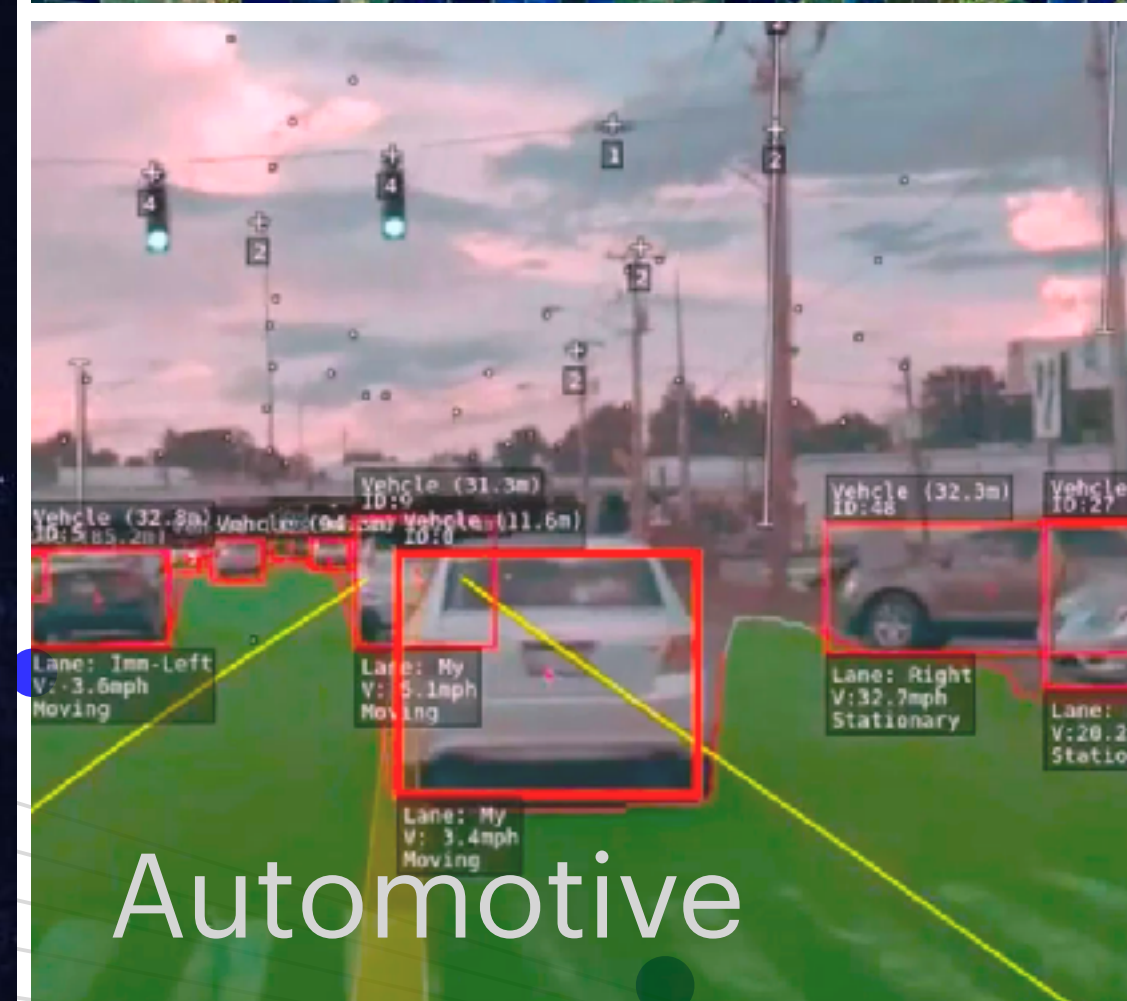
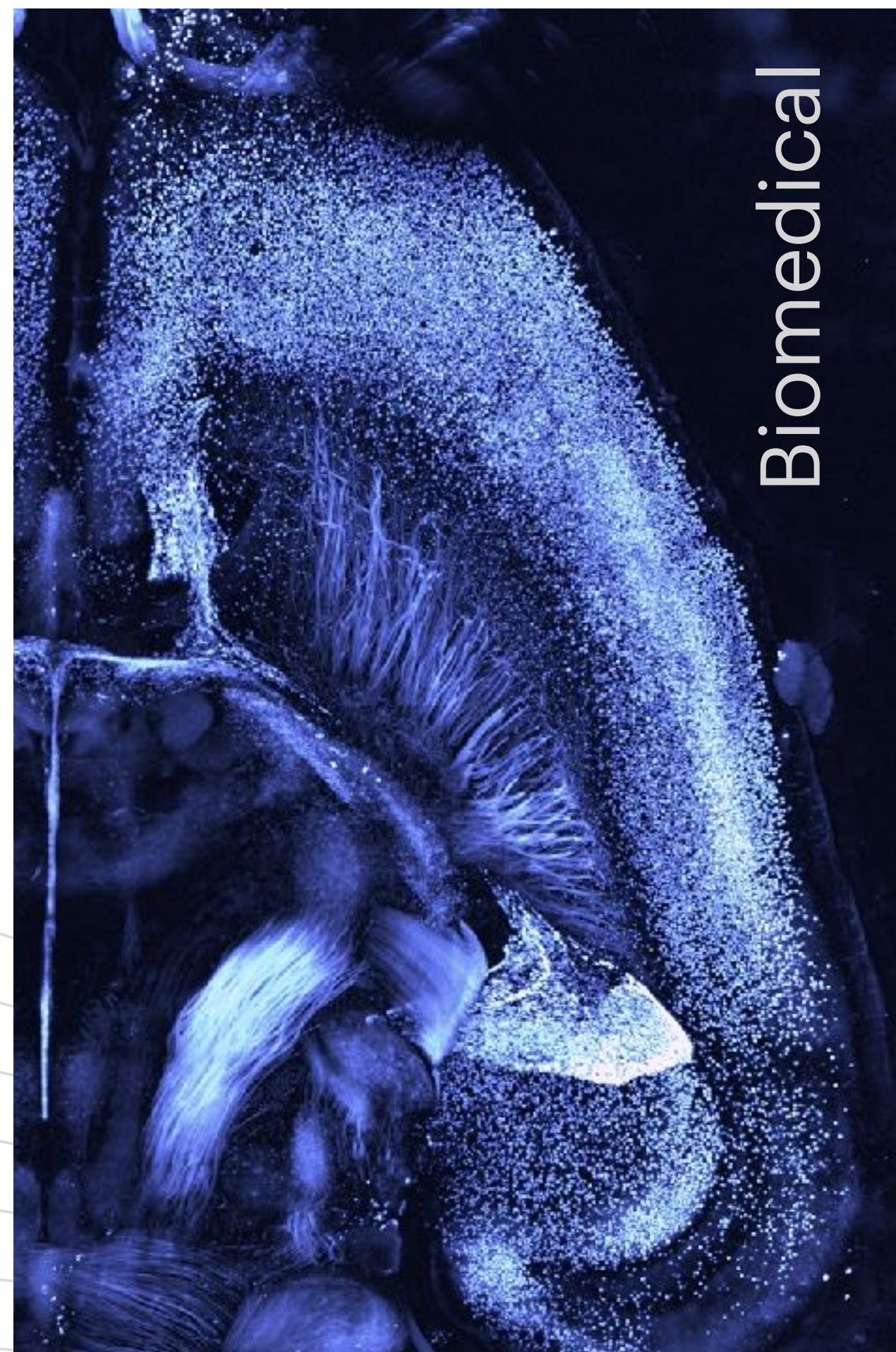
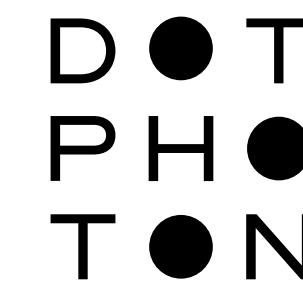


Dotphoton

EIPICSA 2



What we had: the Jetraw algorithm

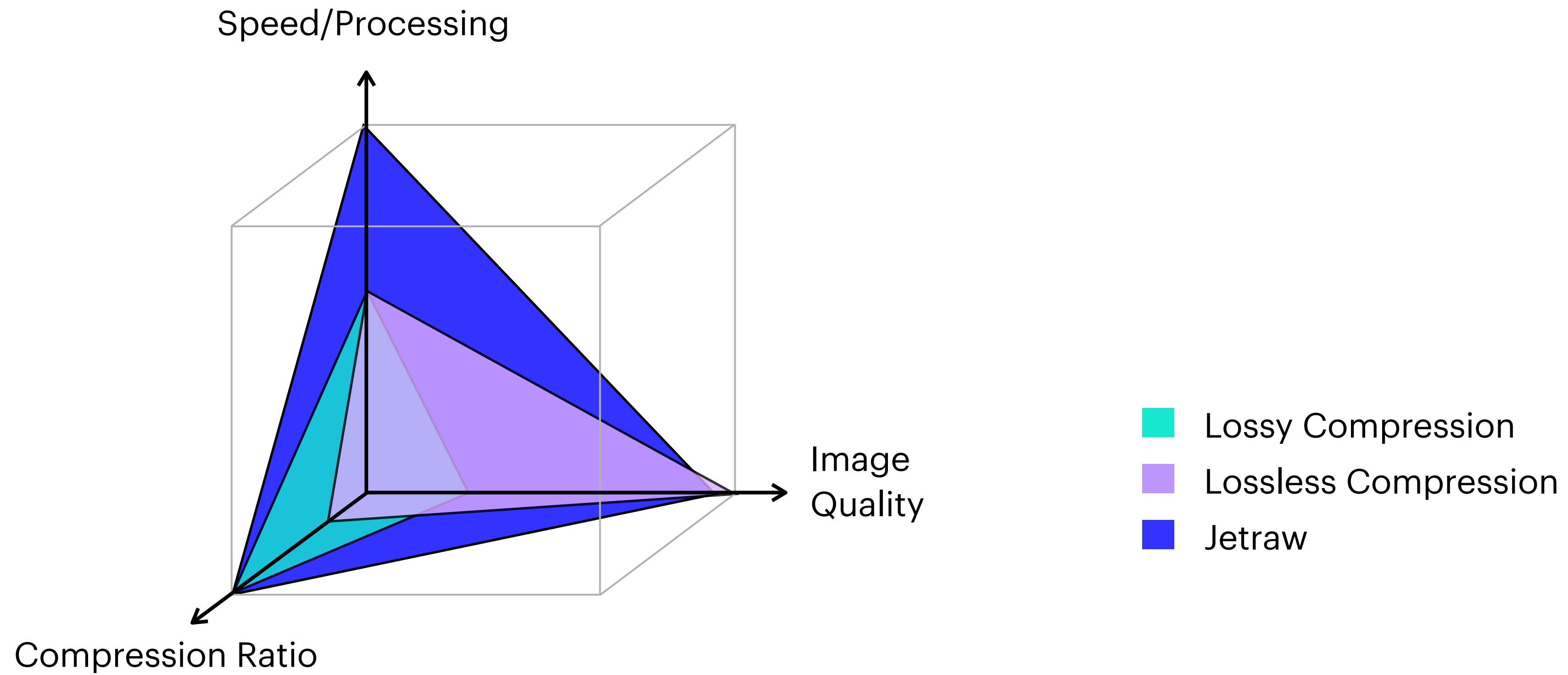
- A compression algorithm that treats images as measurements.
- Market growth due especially to an increase in Machine Learning applications.
- A proof-of-concept for the space implementation (GSTP Element 1).

What we wanted to achieve

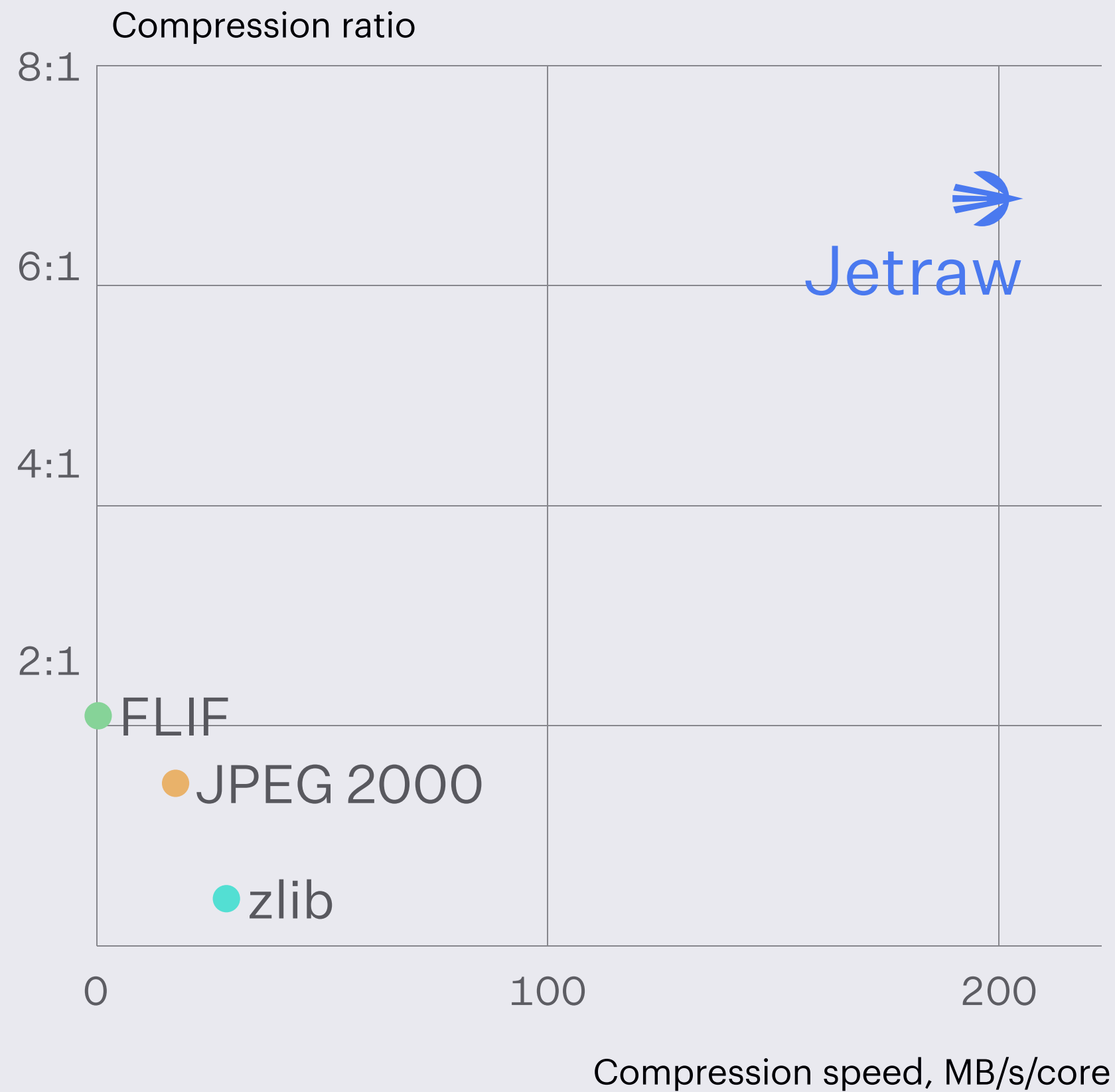
- An FPGA implementation, suitable for being integrated in space missions.
- Test and validate image compression before the mission (quality, compression ratio)
- Improve our development process and tools
- Make a portable demonstrator camera with compression for trade shows
- Demonstrate the increased performance that can be had when ML algorithms are used with raw data.
- Find customers for the products

The Jetraw algorithm

Image compression is a trade-off between quality, speed and size



Jetraw has the best in class performance in compression ratio and speed thanks to embedded sensor noise model



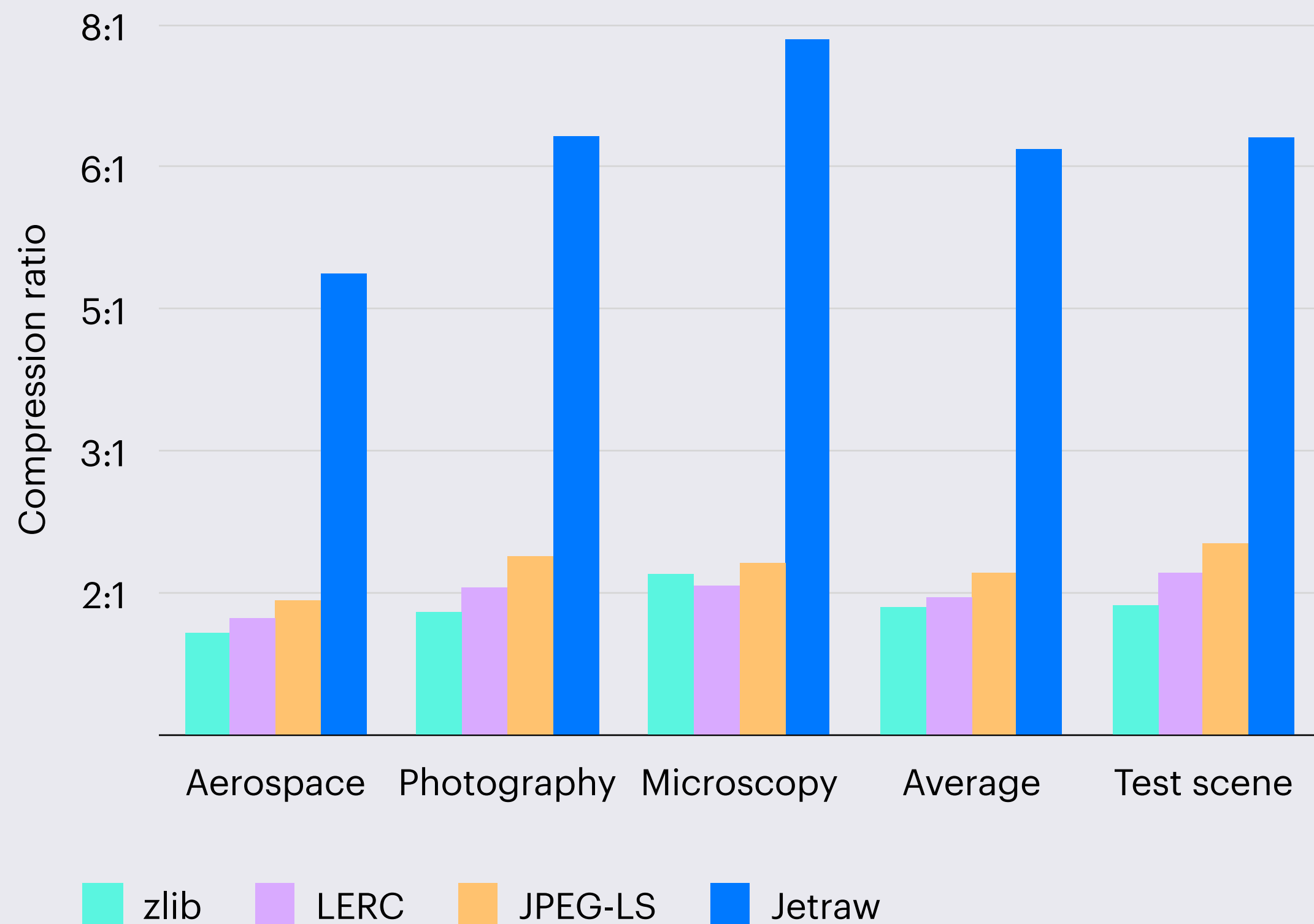
“ It was extremely appealing from PCO's point of view to obtain a compression method that exploits the individual image capture chain of each camera model”



Gerhard Holst,
Head of Science &
Research at PCO

Jetraw has the best in class performance in compression ratio and speed thanks to embedded sensor noise model

Average compression ratio by application



“ We selected the Jetraw by Dotphoton due to the combination of the method’s tight control on the maximum compression error, the compression ratio achieved and the algorithm speed”

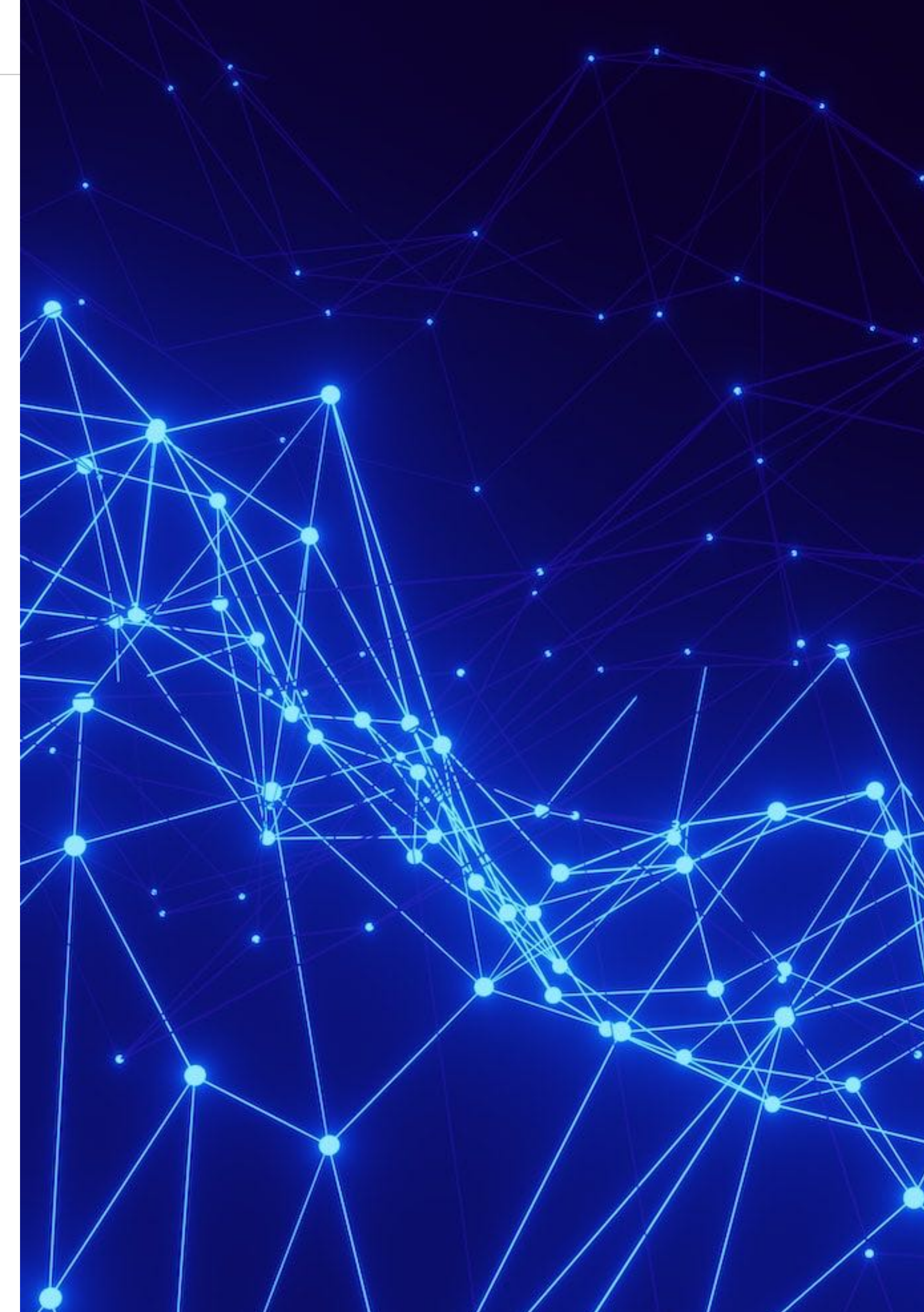


Chris Dunsby,
Imperial College
London

**Imperial College
London**

Reasons for the difference. Jetraw has been developed for raw data, seen by algorithms.

- 1 Other compression methods have been developed for the human eye, To maximise PSNR, to reduce mean-square difference between original and compressed file.
- 2 PSNR assumes that the original image is noise-free.
- 3 PSNR only depends on signal amplitude, not correlations.
- 4 For modern sensors and applications, Jetraw is more suitable.



Noise plays an important role in modern images.
Other compressors allocate more storage to noise
than to information



Original 2: PSNR 52dB

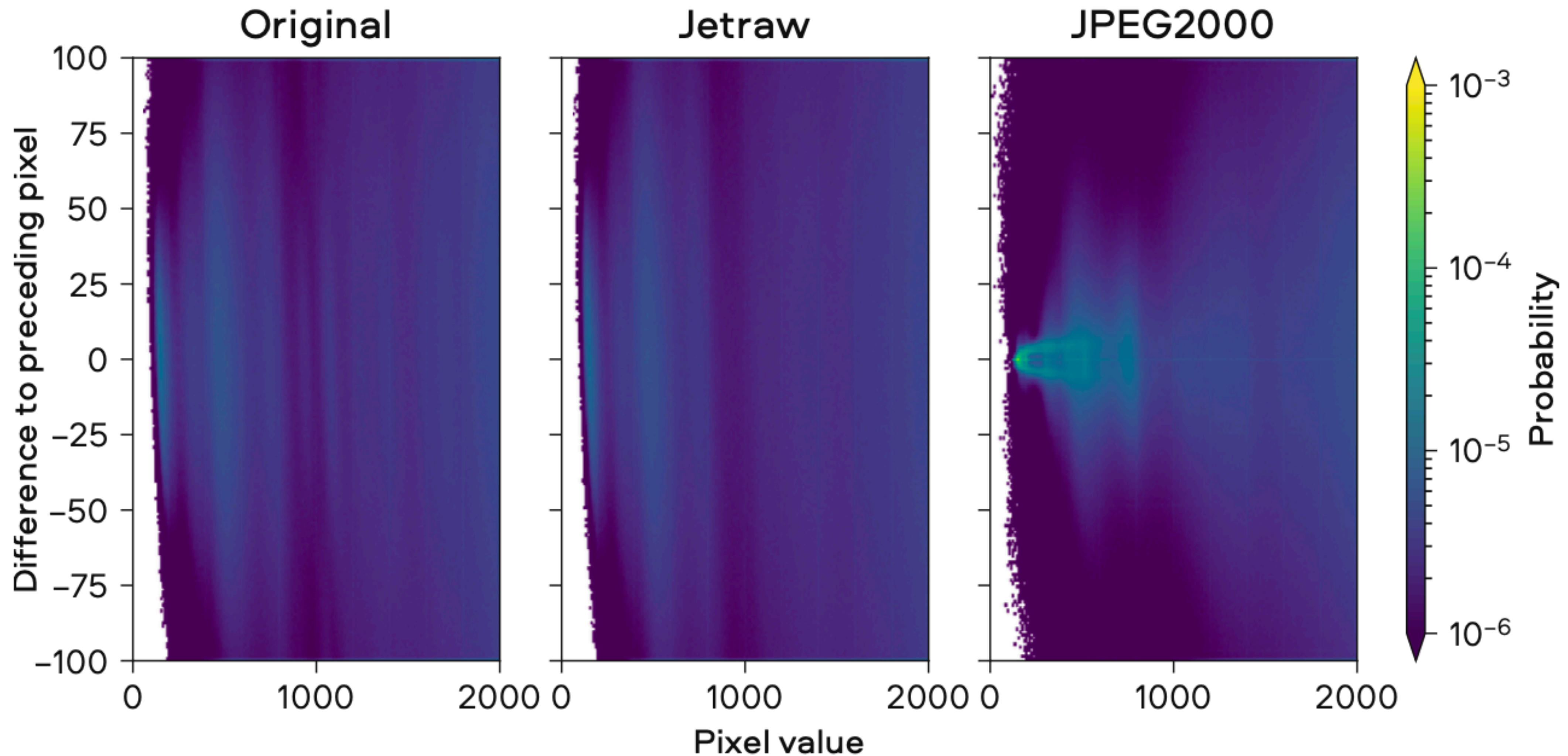


Original



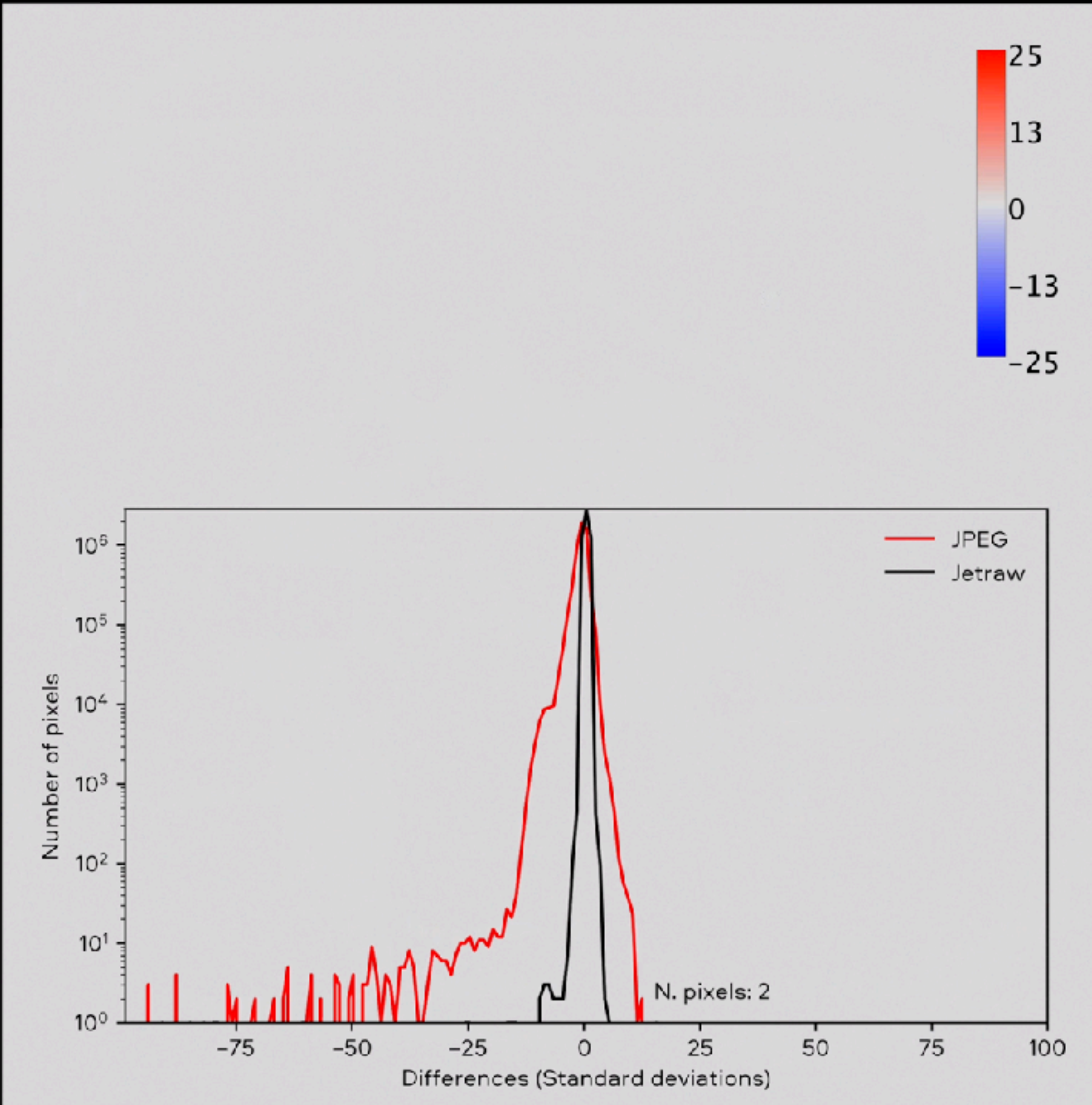
JPEGXL: PSNR 53dB

Other compression methods destroy correlations

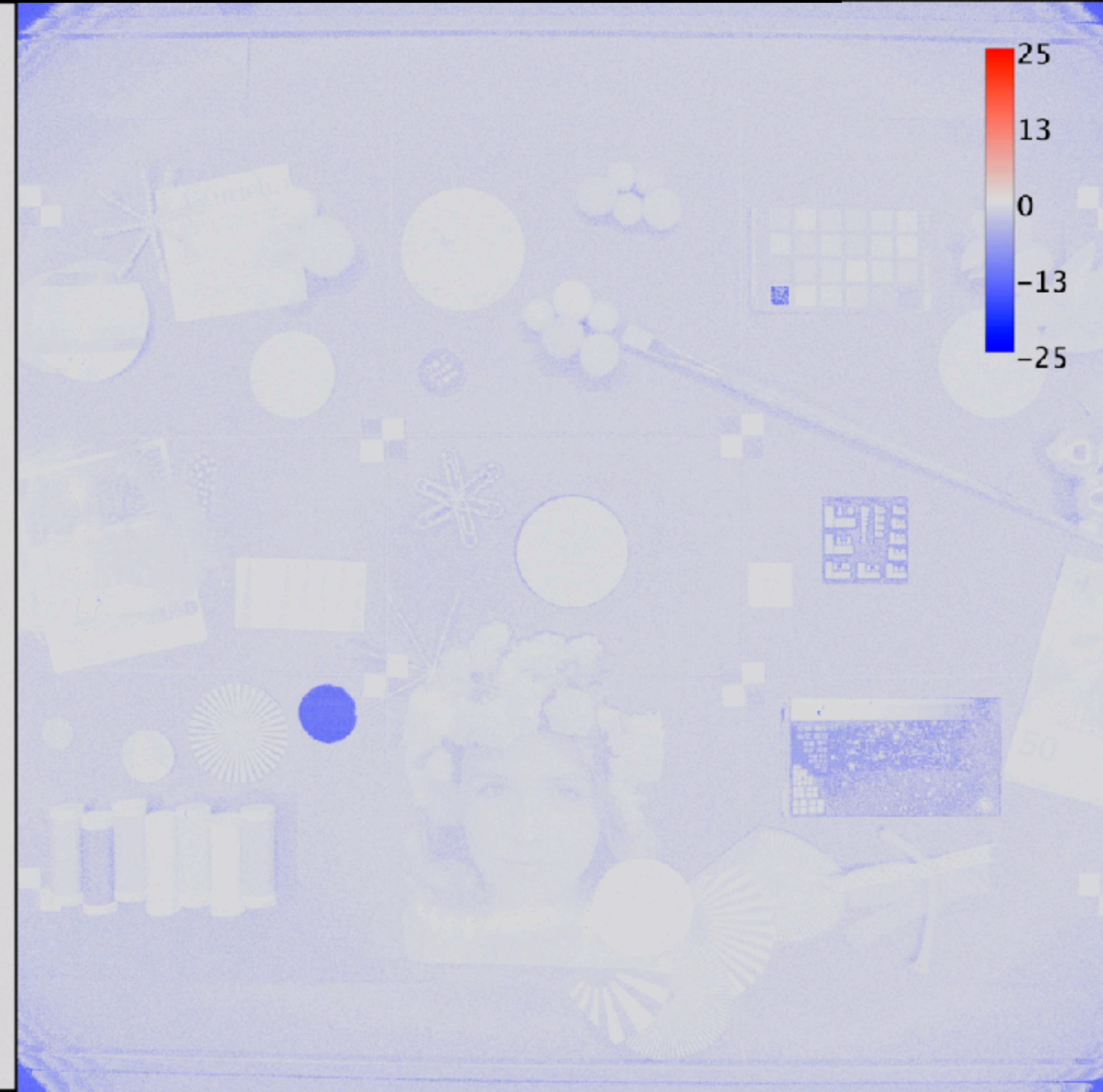


Other compression methods are biased

Jetraw



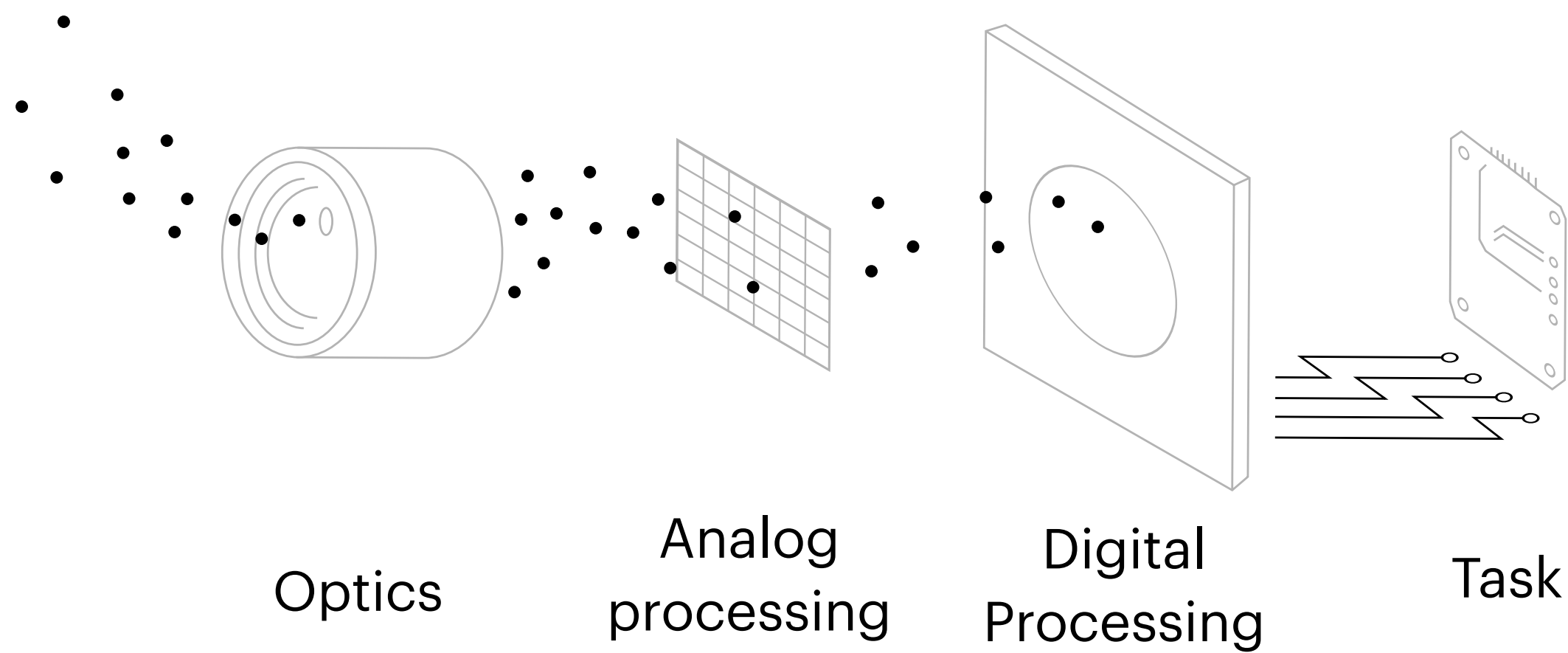
JPEG



Raw is a well defined measurement

Digital image generation

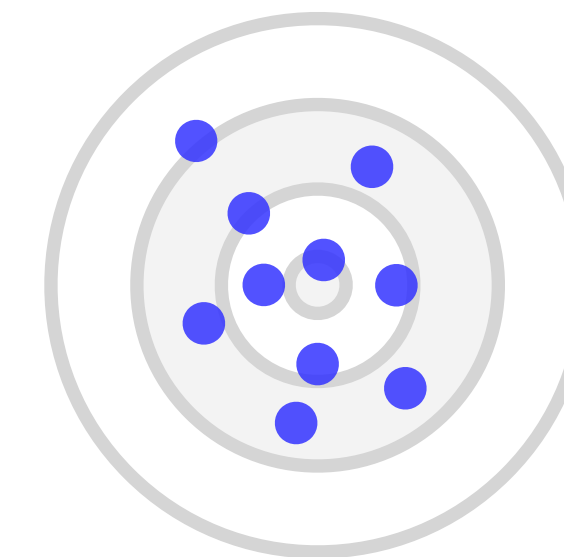
Requirement: Traceability of the image generation process (camera or sensor)



Errors are:

- ✓ Specified
- ✓ Independent
- ✓ Unbiased

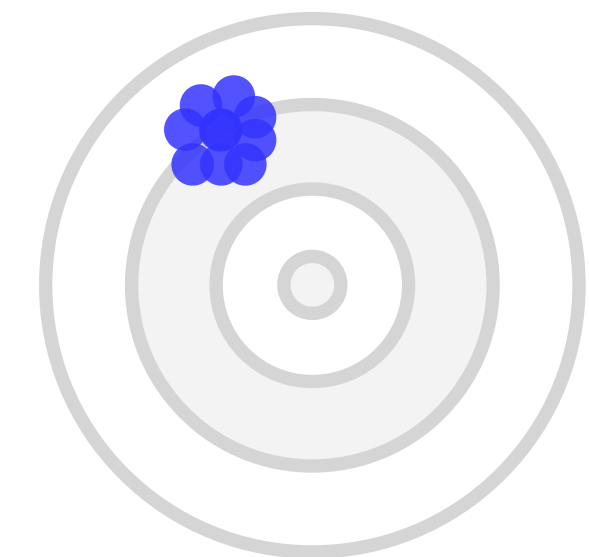
Precision & accuracy



Raw data

Low precision

High accuracy

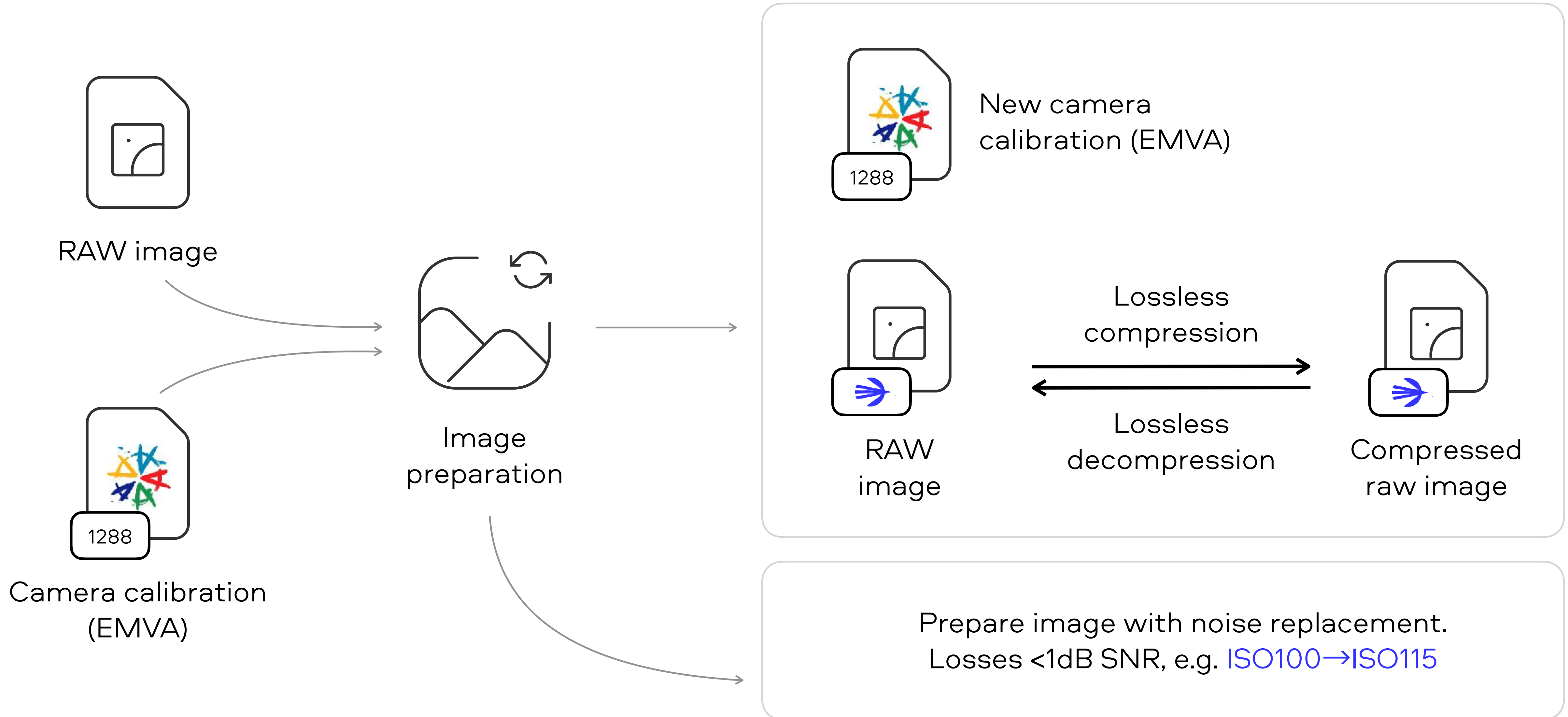


Processed data

High precision

Low accuracy

Steps to compression and decompression



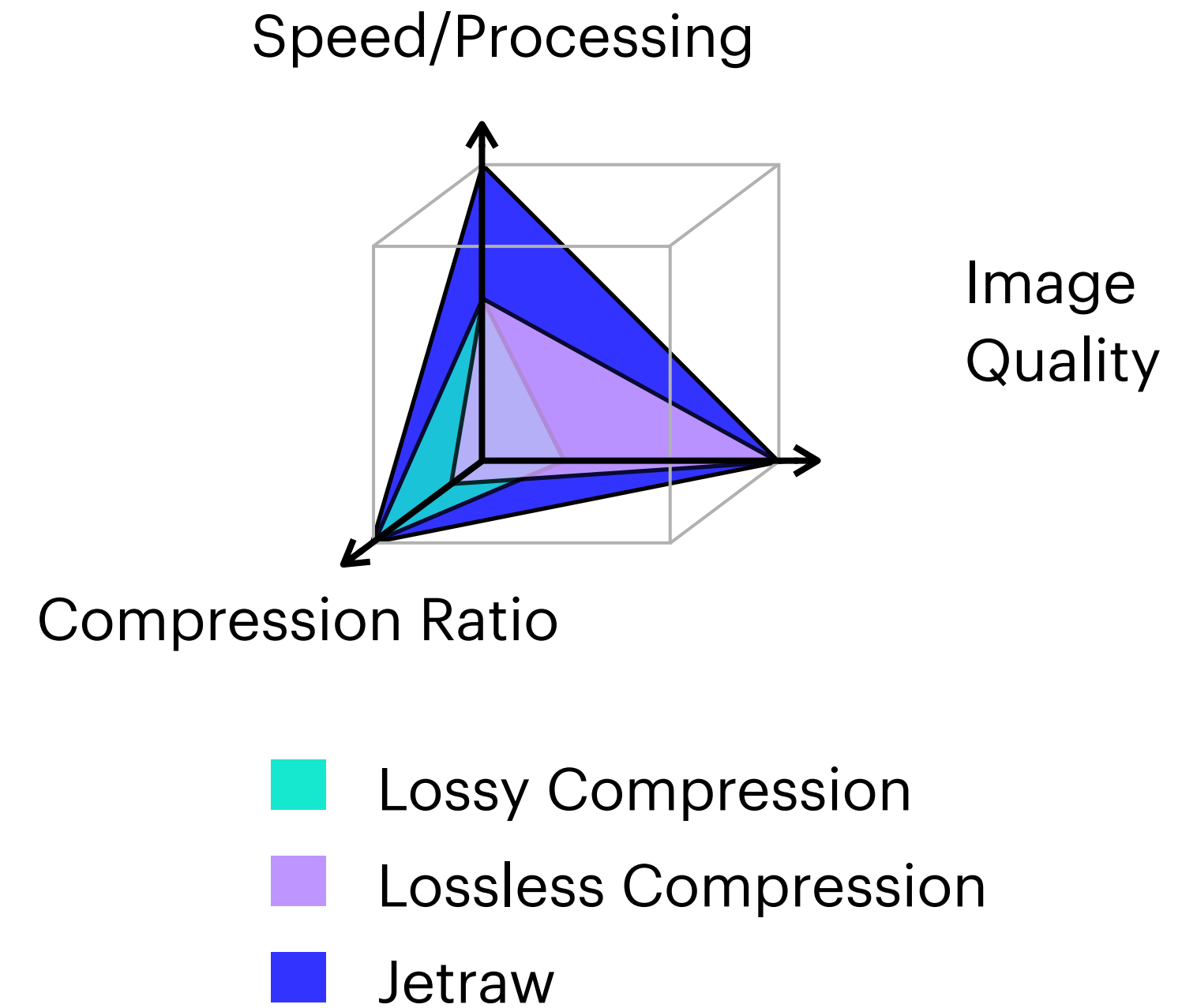
The tradeoff

Disadvantages of Jetraw

- ✓ Only works for raw data
- ✓ The sensor and settings must be known at compression time
- ✓ Several complexities in implementing the algorithm

Further advantages of Jetraw

- ✓ Engineering with accurate tolerances
→ easier testing and failure detection
- ✓ Generate accurate synthetic data
→ test system "in silico"
- ✓ Use uncertainty data to improve algorithm performance
→ "better than lossless".



Summary

Loss level	Tightly-controlled 1.2dB SNR (equiv. increase ISO100→ISO115)
Typical compression ratio	7:1 (always in the range 4:1 to 9:1)
Typical speed	FPGA → up to 100Gbps (6.25 Gpx/s) CPU (Intel/AMD/ARM) → 1.6Gbps/core (100Mpx/s/core)
Data formats	Any uncompressed format, typically TIFF or proprietary raw formats

Practical example 1: one 8Mpx camera, 30 fps

	Data rate	Disk video capacity time (4TB ssd)	Upload time (8h of video, over 1Gbps)
Without Jetraw	480 MB/s 1.7 TB/h	2.3 hours	16h
With Jetraw	68 MB/s 0.25 TB/h	16 hours	3h

Deployment possibilities

	Advantages	Development status
FPGA (on instrument)	Improve performance from the start	Ready
FPGA (datacenter)	High throughput with low CPU and network load	In development
Software library	Easy deployment, fast enough	Ready
Transparent compression proxy	Easy large scale deployments	Beta

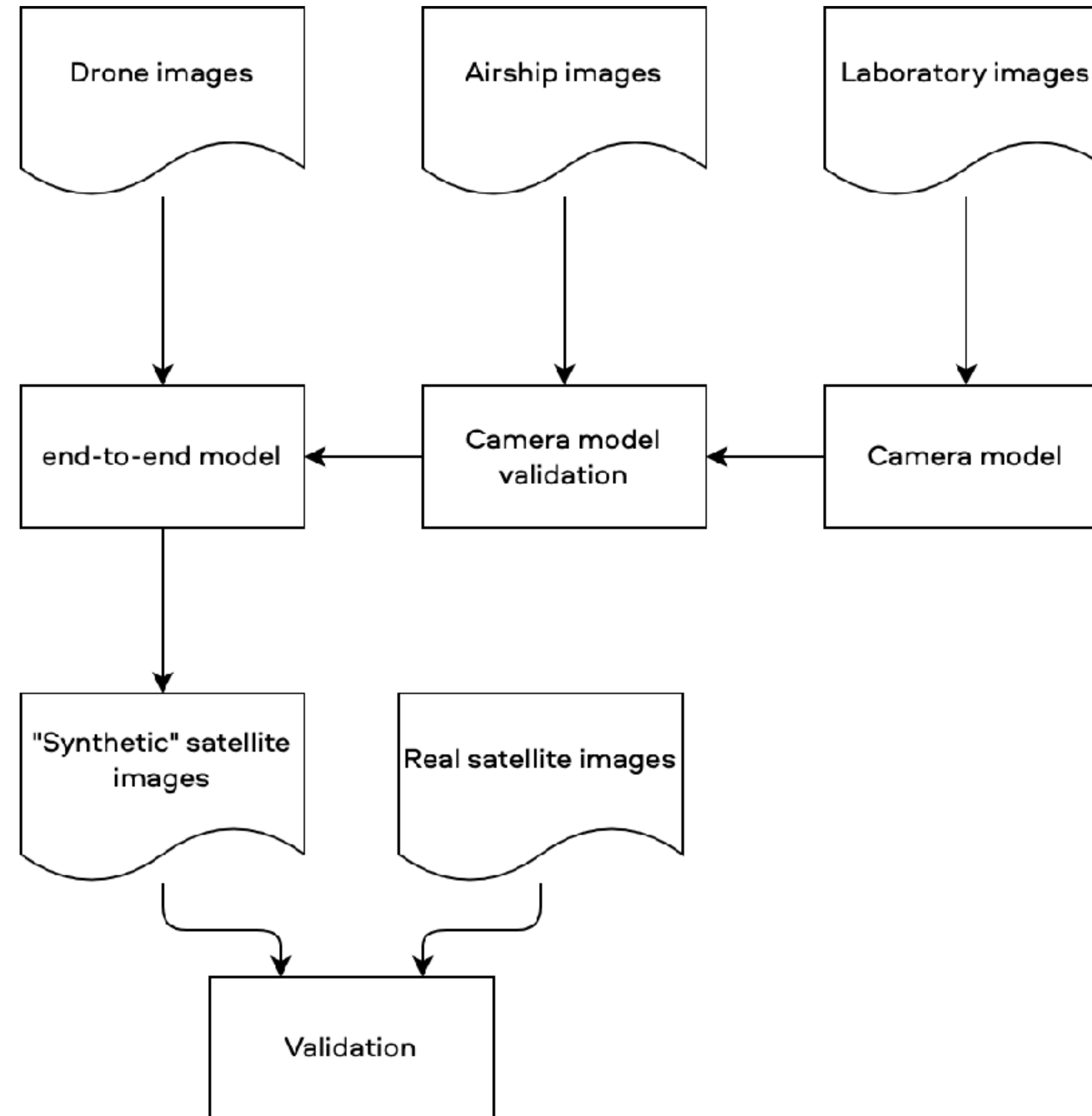
FPGA performance and resource usage

Resource usage (for an image 5000x5000 pixels, 16bpp) Vivado 2020.2, ZCU102, 200MHz.						
Pixels/clock	LUT	FF	BRAM	DSP	Mpx/s	Power (W)
1	3968	1874	7.5	16	200	0.149
2	5608	3063	9	32	400	0.225
4	9612	5621	14	64	800	0.448
8	17644	10718	24	128	1600	0.935
16	33944	20814	39.5	256	3200	1.528
32	70790	41828	70.5	512	6400	3.255

Validation tools

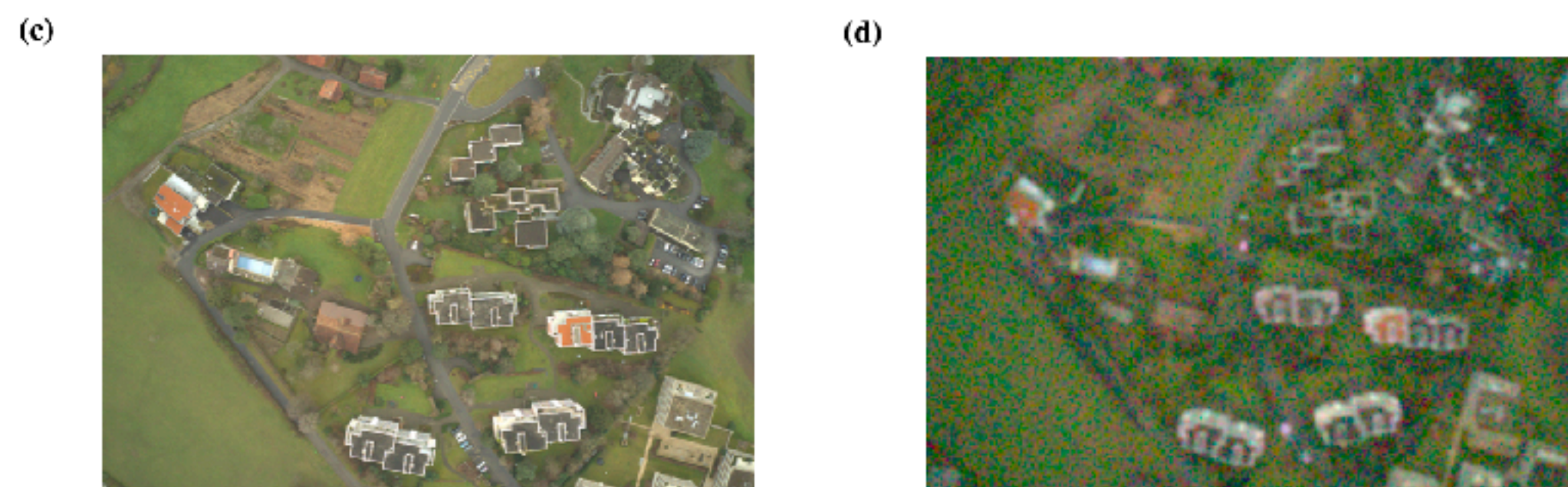
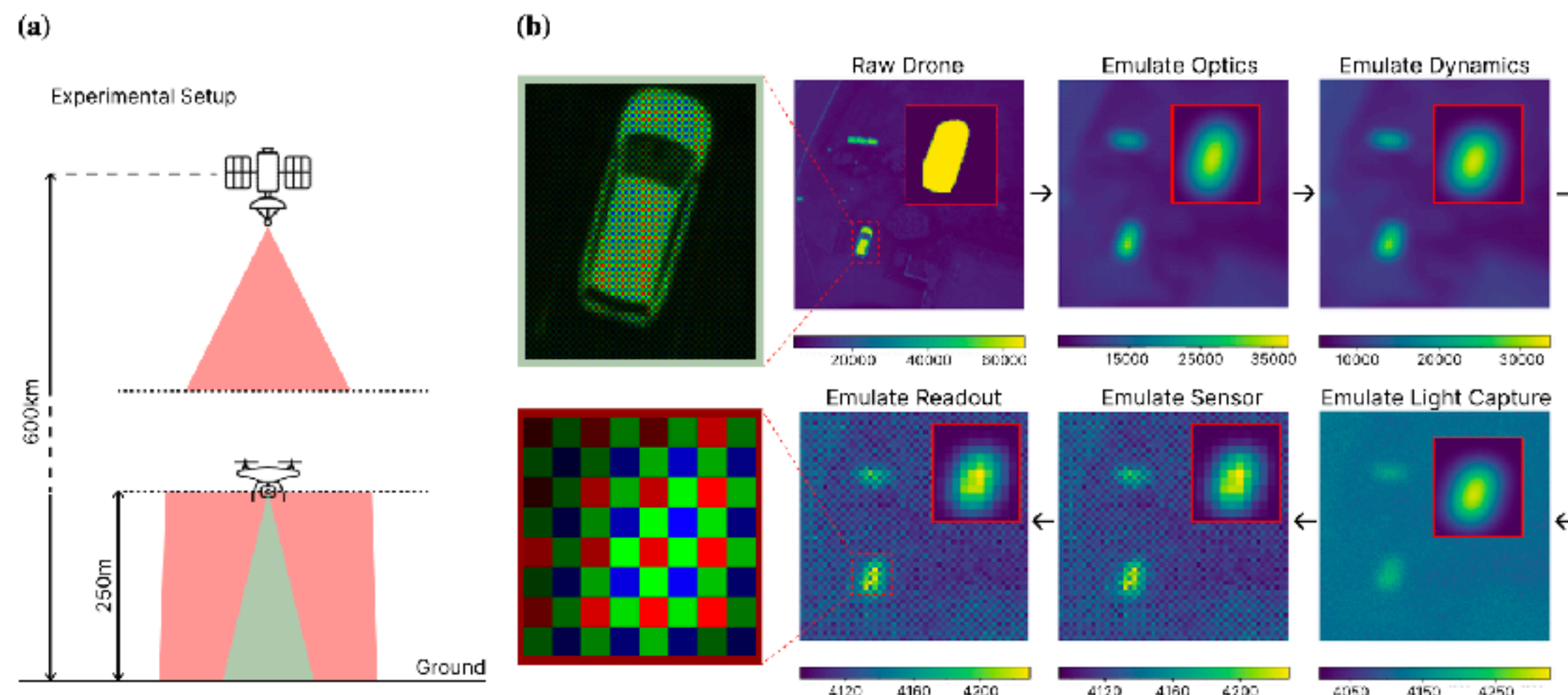
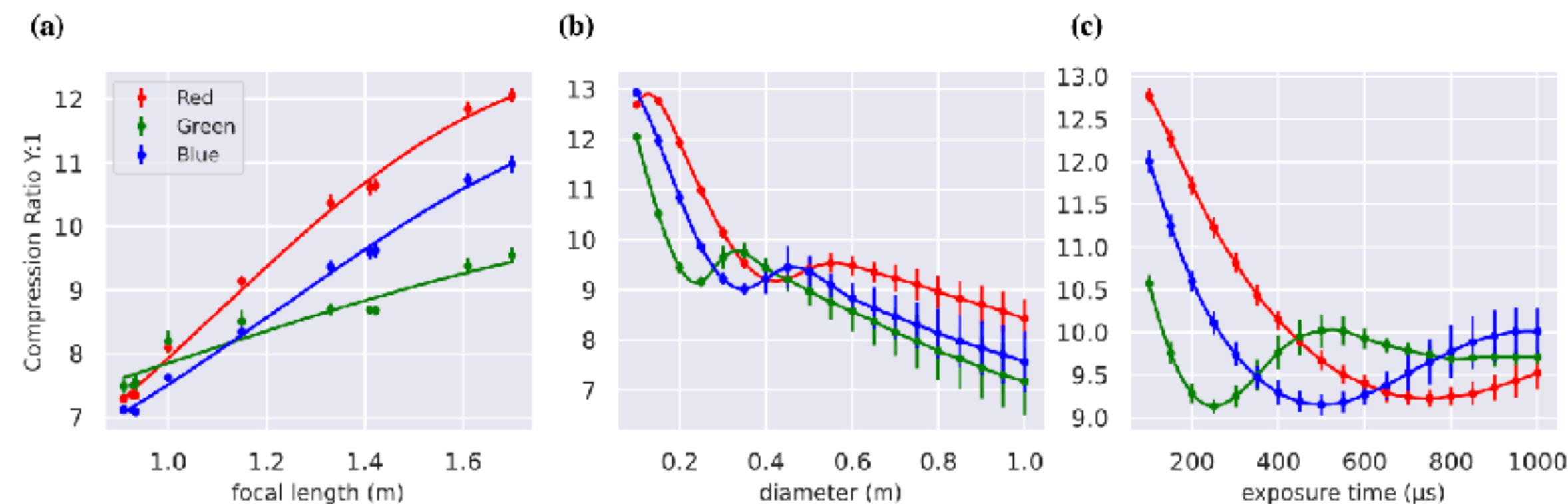
Testing, verification and validation tools

End-to-end model overview



End-to-end model results

- Inspired from "Pyxel", using the "Poppy" physical optics propagation code
- Tolerances discussed with MediaLario



Physical Data Models in Machine Learning Imaging Pipelines

Marco Aversa
University of Glasgow and Dotphoton AG
Glasgow, United Kingdom
marco.aversa@glasgow.ac.uk

Luis Oala
Fraunhofer HHI
Berlin, Germany
luis.oala@hhi.fraunhofer.de

Christoph Clausen
Dotphoton AG
Zug, Switzerland
christoph.clausen@dotphoton.com

Roderick Murray-Smith
University of Glasgow
Glasgow, United Kingdom
roderick.murray-smith@glasgow.ac.uk

Bruno Sanguinetti
Dotphoton AG
Zug, Switzerland
bruno.sanguinetti@dotphoton.com

Data-centric AI workflow based on compressed raw images

Marco Aversa^{1,2}, Ziad Malik¹, Phillip Geier^{1,3}, Fabien Droz⁴, Andres Upegui¹, Roderick Murray-Smith², Christoph Clausen¹, Bruno Sanguinetti^{1,*}

¹Dotphoton AG, Zug, Switzerland
²University of Glasgow, Glasgow, United Kingdom
³HEPIA, Geneva, Switzerland
⁴CSEM, Neuchatel, Switzerland
*bruno.sanguinetti@dotphoton.com

26 September, 2022

Abstract

In order to extract the full potential of the high volume of image data coming from earth observation, image compression is a key step in the pipeline. This paper presents a data-centric AI workflow based on compressed raw images. The workflow is designed to be efficient and accurate, and it is implemented in a modular way. The results show that the proposed workflow can achieve a high compression ratio while maintaining a high level of accuracy. The workflow is implemented in a modular way, and it can be adapted to different use cases. The results show that the proposed workflow can achieve a high compression ratio while maintaining a high level of accuracy.

create an
ex image
h human
learning
w format.
machine
following
emulate
d process
rameters
time and
in neural
le image
d up and
accuracy.





FPGA validation

- Questasim
- “Design-tester” automated AXI4-stream design tool in python
- vunit
- Test platform 1, based on KU105, images sent from computer (quality testing)
- Test platform 2, based on KU105, artificial test patterns generated on FPGA (perf testing)
- Demonstrator, based on KD260 (“Krya Vision”)

New development process and tools

- “Notion” did finally not work out, difficult to do configuration management.
- New development process based on integrating the following:
 - Github (version tracking)
 - cmake (configuration management)
 - Sonarqube (CI/CD, static code analysis)
 - Grist (Requirements management, traceability and project management)
 - Latex (Documentation)



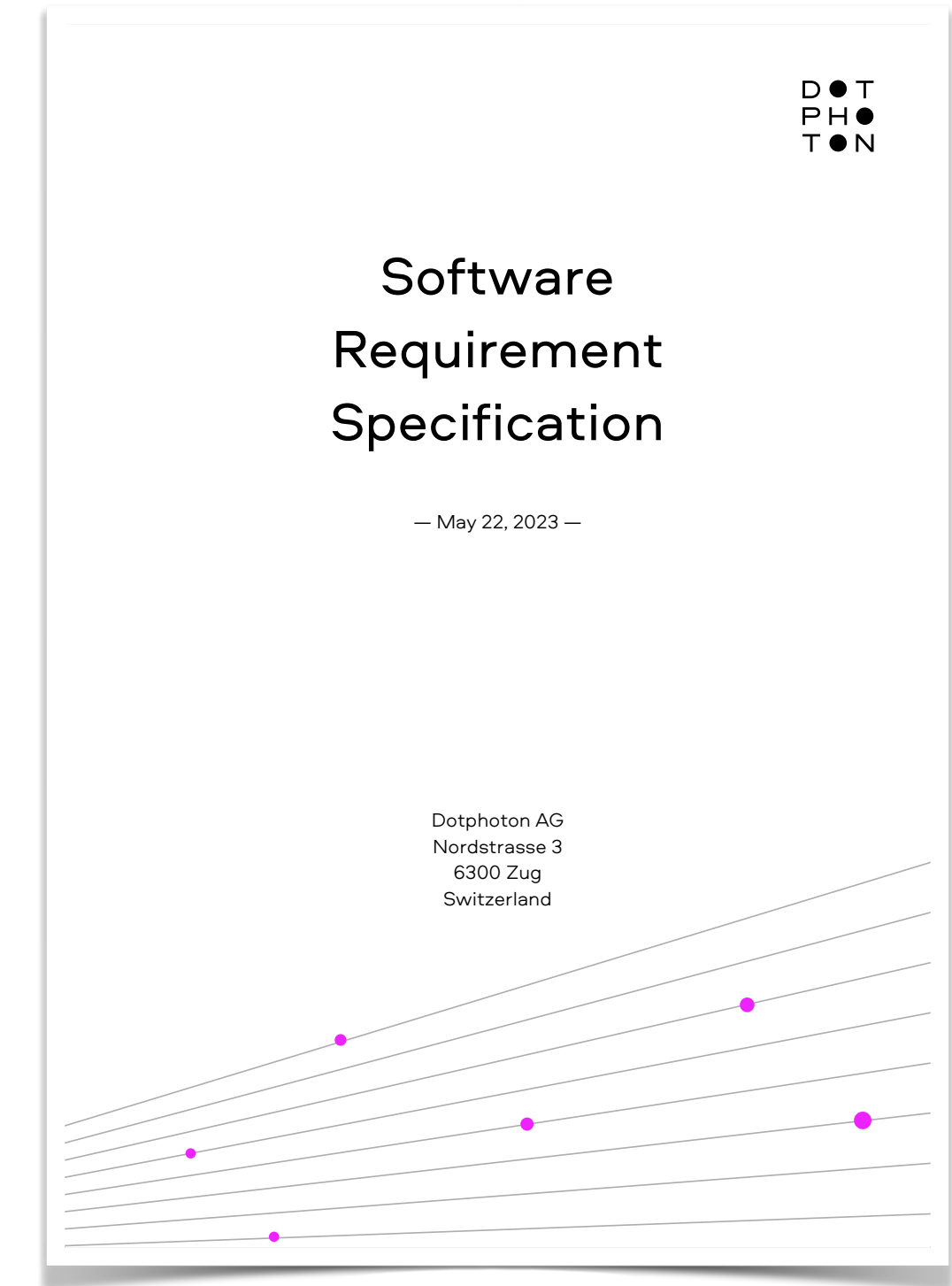
grist.dotphoton.com

Start Page

philo.gier / EPICSA2 / HW-ARS

HW-ARS

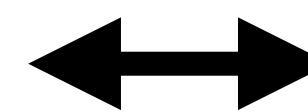
ARS-ID	Summary	Verified by	Validated by	Defined by
1 ARS-01-01	The IP shall have an IDLE operating mode	HW-VER-001 HW-VER-002	HW-VAL-01 HW-VAL-02 HW-VAL-03 HW-VAL-04	GEP
2 ARS-01-02	The IP shall have a COMPRESSING operating mode	HW-VER-003 HW-VER-004	HW-VAL-01 HW-VAL-02 HW-VAL-11	GEP
3 ARS-01-03	The IP shall have a BYPASS operating mode	HW-VER-005 HW-VER-006	HW-VAL-03 HW-VAL-12	GEP
4 ARS-01-04	The IP shall have an AUTOTEST operating mode	HW-VER-007 HW-VER-008	HW-VAL-04 HW-VAL-05	GEP
5 ARS-01-05	The IP core shall expose parameters that are configured at synthesis	HW-VER-009	HW-VAL-09	GEP
6 ARS-01-06	The IP core shall expose parameters that are configured at runtime	HW-VER-010	HW-VAL-02 HW-VAL-05	GEP
7 ARS-01-07	The IP core shall be divided into smaller sub-modules	HW-VER-011		GEP
8 ARS-02-01	The data input interface shall implement an AXI4-Stream compliant slave interface	HW-VER-012 HW-VER-013	HW-VAL-01 HW-VAL-02 HW-VAL-03 HW-VAL-09 HW-VAL-11 HW-VAL-12	GEP
9 ARS-02-02	The compressed data output interface shall implement an AXI4-Stream compliant master interface	HW-VER-014 HW-VER-015	HW-VAL-01 HW-VAL-02 HW-VAL-03 HW-VAL-09 HW-VAL-11 HW-VAL-12	GEP
10 ARS-02-03	The configuration interface shall implement an AXI4-Lite compliant slave interface	HW-VER-016 HW-VER-017 HW-VER-018	HW-VAL-02 HW-VAL-05 HW-VAL-09 HW-VAL-07 HW-VAL-08 HW-VAL-14	GEP
11 ARS-02-04	The configuration interface shall expose a set of memory mapped registers to allow the configura...	HW-VER-018	HW-VAL-02 HW-VAL-05 HW-VAL-09	GEP
12 ARS-02-01	The operating frequency for the IP core shall be at least 100MHz	HW-VER-019	HW-VAL-06 HW-VAL-08 HW-VAL-11 HW-VAL-12	GEP
13 ARS-03-02	The IP core should have a throughput of 1 pixel/clock cycle	HW-VER-020	HW-VAL-09 HW-VAL-10 HW-VAL-11 HW-VAL-12	GEP
14 ARS-03-03	The IP core shall use a single clock domain	HW-VER-021	HW-VAL-08 HW-VAL-09 HW-VAL-11 HW-VAL-12	GEP
16 ARS-05-01	Input pixel data format shall be configurable at synthesis through generic parameters	HW-VER-022	HW-VAL-08 HW-VAL-09 HW-VAL-11 HW-VAL-12	GEP
16 ARS-05-02	Image dimensions shall be configurable at synthesis through generic parameters	HW-VER-023	HW-VAL-08 HW-VAL-09 HW-VAL-11 HW-VAL-12	GEP
17 ARS-05-03	Sensor model parameters shall be configurable at runtime through configuration registers	HW-VER-024	HW-VAL-02 HW-VAL-05 HW-VAL-14	GEP
18 ARS-05-04	Encoder parameters (e.g. bits) shall be configurable at runtime through configuration registers	HW-VER-025		GEP
19 ARS-05-05	The IP core shall implement the latest image compression algorithm	HW-VER-026 HW-VER-027	HW-VAL-01 HW-VAL-02 HW-VAL-03 HW-VAL-11	GEP
20 ARS-06-06	The output stream shall contain all the necessary information for decompression.	HW-VER-028	HW-VAL-01 HW-VAL-02 HW-VAL-11	GEP
21 ARS-06-07	The IP core should be able to account for corrected (pixel linearization) values instead of raw pix...	HW-VER-029		GEP
22 ARS-06-08	The operating mode shall be set accordingly based on user configuration and the current context	HW-VER-030	HW-VAL-07 HW-VAL-09 HW-VAL-13	GEP
23 ARS-06-09	The error bound shall be configurable at runtime.	HW-VER-031		GEP



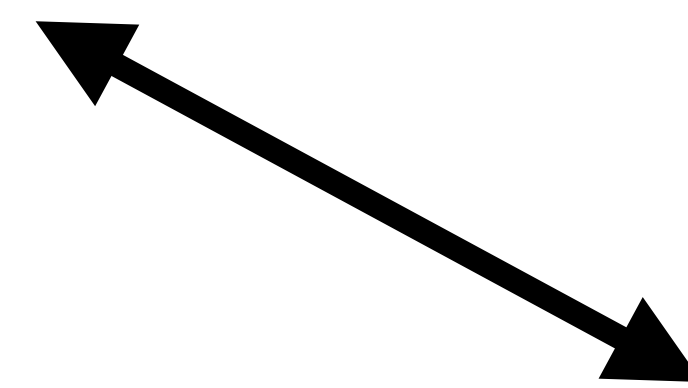
CMake



GitHub

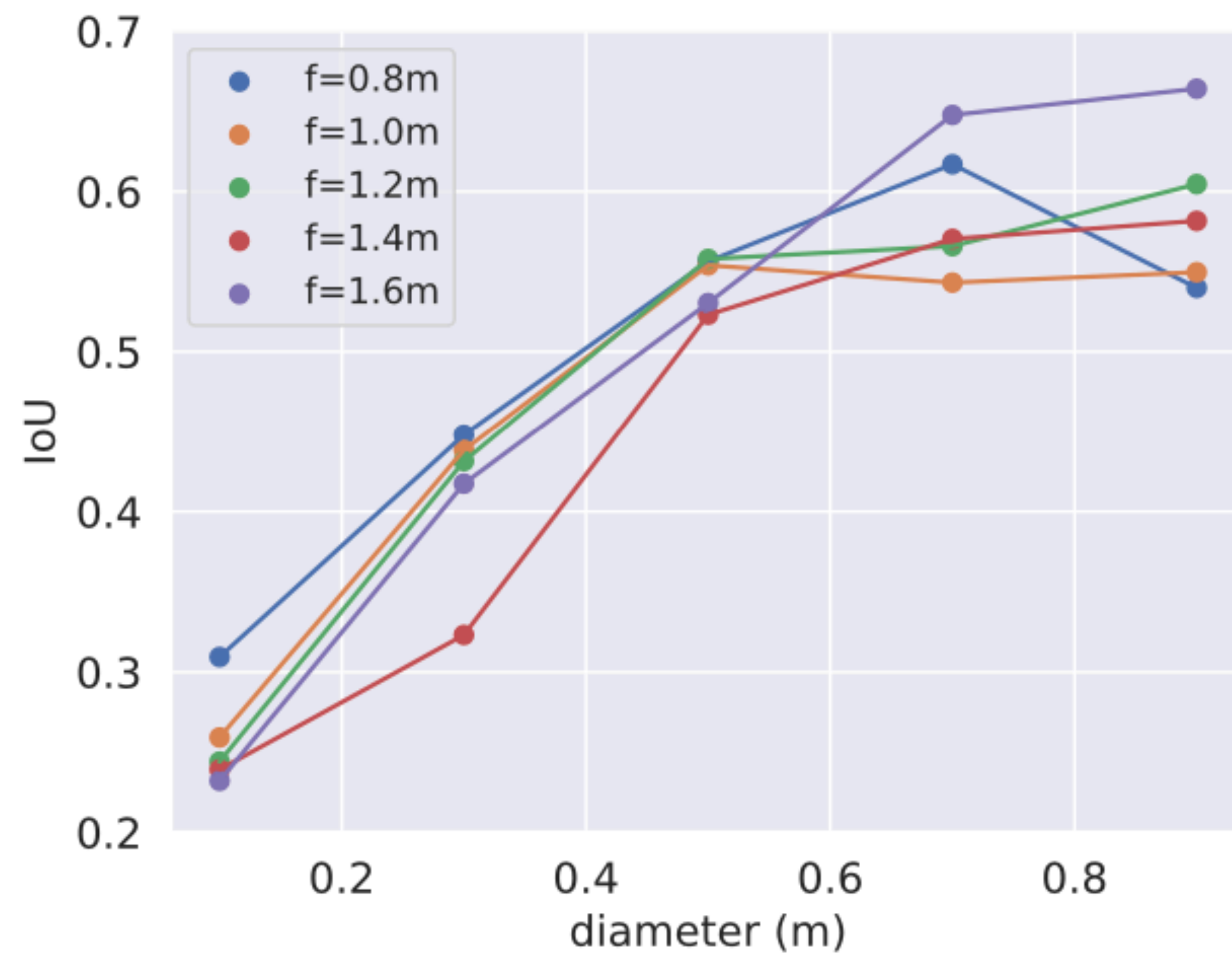
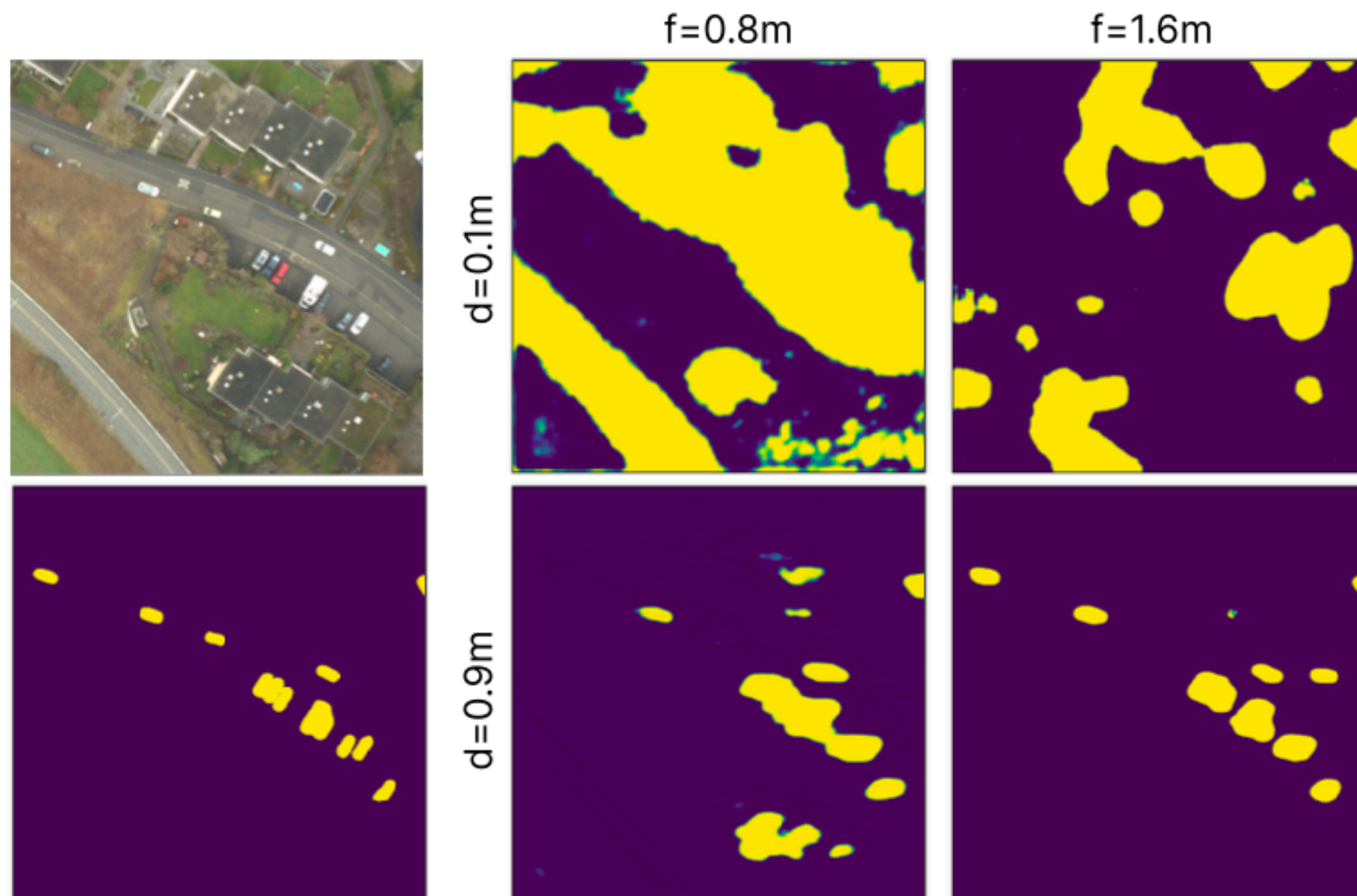


sonarqube



Demonstrate improved performance of
ML algorithms with RAW data

OBPDC 2022, NeurIPS 2022



OBPDC 2022, NeurIPS 2022

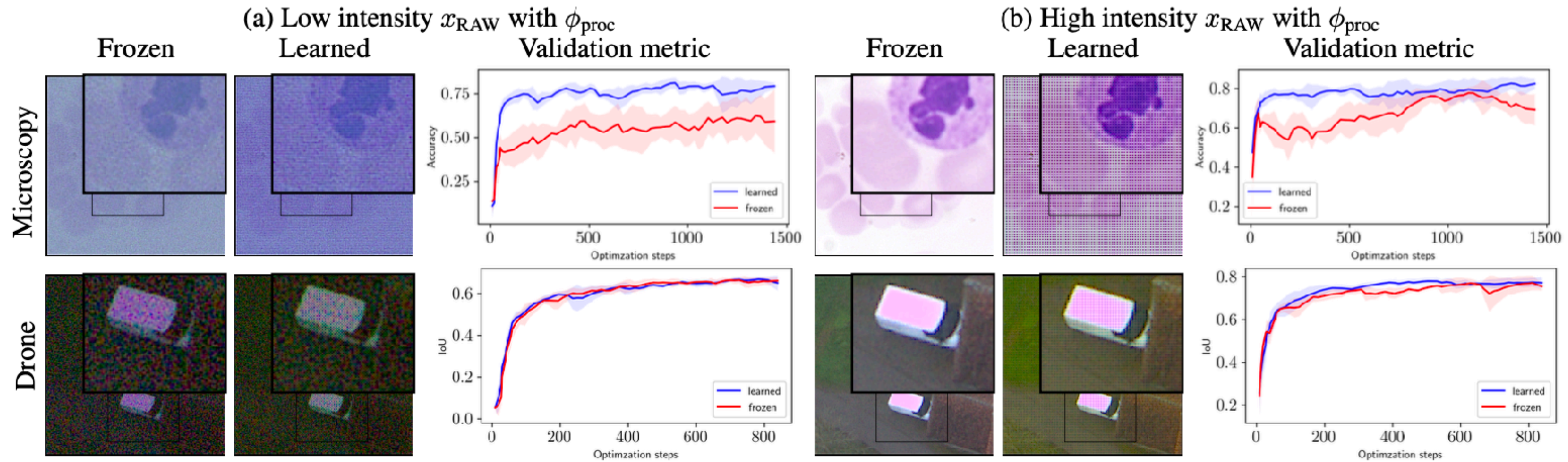


Figure 3: Low (a) and high (b) intensity images processed by a *frozen* and a *learned* pipeline. This dataset drift adjustment would not be possible with processed data typically used for machine learning experiments. The plots in the rightmost column of each block display the mean of validation metrics over five cross-validation runs. Error bars are reported as one standard deviation. Optimization steps 1439 and 915 correspond to epoch 60 into training.

Demonstrator: Live!