Mosaic-SR: An Adaptive Multi-Step Super-Resolution Method For Low-Resolution 2D Barcodes

Enrico Vezzali[1,2], Costantino Grana[1], Lorenzo Vorabbi[2], and Federico Bolelli[1]

[1]*Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Italy*
[2]*Datalogic, S.p.A, Bologna, Italy*
enrico.vezzali@unimore.it

Link to repository

1 – Introduction

In many real-world applications, barcodes must be scanned **from a distance**.

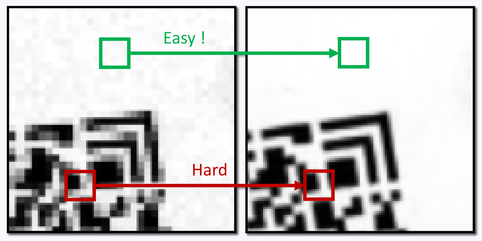
In **warehouses**, parcels may sit on **high shelves**; in **production pipelines**, **small parts move rapidly** along conveyor belts.

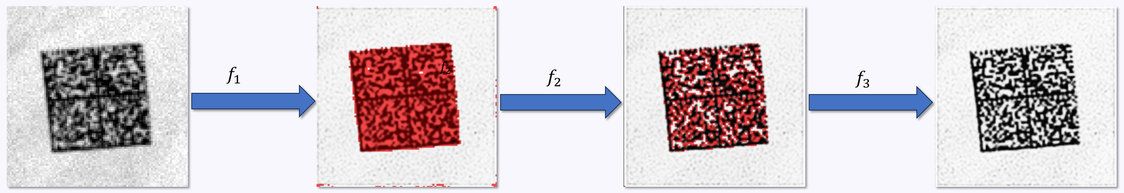
These conditions often yield **low-resolution images** where **black and white modules blur together**, making the codes **unreadable**.

*We need an upscaling algorithm that is **fast and accurate**.*

2 – Key Idea: Mosaic-SR

- Not all areas of the image are equally difficult to upscale.
- Upscaling **sharp edges and intricate corners** – critical for barcode readability – is the most challenging part.
- **Mosaic-SR iteratively refines only the regions that need it, while cutting computation on uniform backgrounds.**





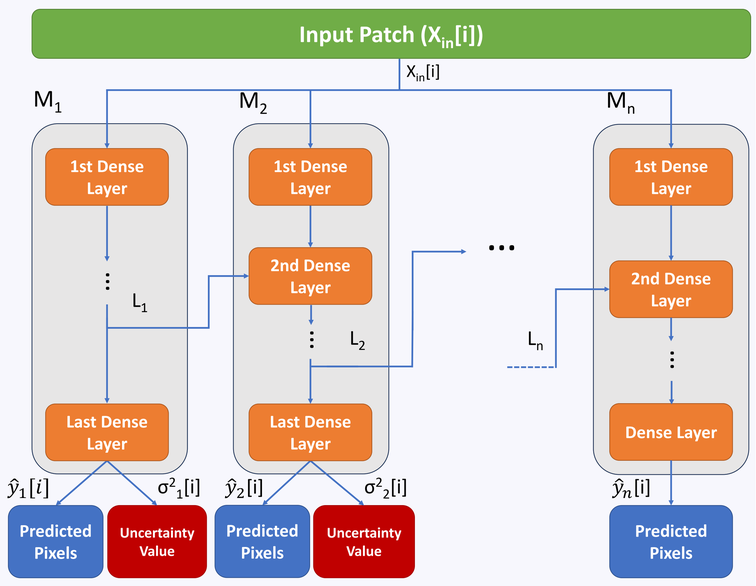
3 – Architecture and Algorithm

Patch-wise: split image into patches of $k \times k$ pixels, centered on each input pixel; flatten to $X_{in}[i]$.

Step Chain: Each step M_j predicts **upscaled patch $\hat{y}_j[i]$, uncertainty σ^2_j[i],** and a **latent L_j**, with increasing precision.

Stop/continue:

- **CASE 1:** $\sigma^2_j[i] \leq Th_j \rightarrow$ The approximation is good enough, we **stop here**;
- **CASE 2:** $\sigma^2_j[i] > Th_j \rightarrow$ The uncertainty is too high! **Refine the estimate further.**



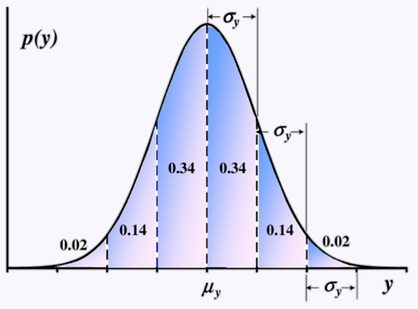
4 – Variance Prediction and Training

Variance Prediction
Each refinement step predicts the **variance (σ^2)** of its output, modeling the error as a Gaussian with zero mean. Variance acts as a measure of confidence: if it falls below a fixed threshold, the refinement stops.

Loss Function
Training minimizes a combined loss that balances reconstruction and variance estimation. This is the loss for a single model:

$$\mathcal{L}_{patch} = \|\hat{y} - y\|_2^2 + \alpha\left(\frac{\|\hat{y} - y\|_2^2}{\hat{\sigma}^2} + 4\ln\hat{\sigma}^2\right)$$

Training Strategy
Models are trained sequentially: first M_1, then M_2, etc. Finally, all are fine-tuned together end-to-end.



5 – Results

Architecture Used
Three predictors M_1, M_2 and M_3 with 1 154, 3 778 and 14 274 parameters, respectively. Patch size 7×7.

Dataset
BarBeR open-source dataset (QR/Datamatrix/Aztec). We generate LR crops with **1.0–1.6 PPM** (step 0.1) and HR at **2.0–3.2 PPM for a** total of **9,562** LR/HR pairs. Decoding via **pyzbar**; timings single-thread on a PC and a **Raspberry Pi 3B+**.

Link to dataset

Results
Across latency tiers, Mosaic-SR achieves **higher PSNR and more decodings at equal or lower time** on PC and Raspberry Pi. The method increases the barcode reader's range by 14.2% more compared to bicubic upscaling.

Method	PSNR (dB, ↑)	SSIM (↑)	# Decodings (↑)	Time PC (ms, ↓)	Time Raspberry (ms, ↓)
Bicubic (openCV)	15.35	0.705	2 336	0.046	1.13
eSR-MAX K5C8	15.62	0.772	2 657	0.887	26.60
Ours (0.25^2, 0.25^2)	**15.77**	**0.782**	**3 330**	**0.883**	**23.29**
ECBSR M4C8	15.81	**0.788**	3 502	2.195	62.83
RT4KSR XXS	15.64	0.775	2 900	2.182	55.73
RT4KSR S	15.64	0.776	2 901	3.413	86.70
Ours (0.13^2, 0.13^2)	**15.83**	0.785	**3 517**	**1.688**	**44.29**
AsConvSR	15.59	0.772	2 815	8.778	200.2
ESPCN	15.84	0.784	3 596	7.121	198.2
eSR-TM K7C16	15.65	0.777	2 940	9.458	238.7
eSR-TR K7C16	15.74	0.781	3 208	9.938	318.6
FSRCNN	15.85	0.787	3 602	13.45	442.9
IMDN RTC	15.83	**0.790**	3 695	10.97	356.6
RT4KSR XL	15.64	0.776	2 893	6.586	178.9
QRCNN	15.77	0.780	3 488	172.1	3 162
Ours (0.05^2, 0.07^2)	**15.90**	0.788	**3 714**	**4.221**	**109.9**