

ARTICLE

Designing a JPEG-compatible receiver-side residual refinement pipeline for unmanned aerial vehicle image transmission and object detection

Le Wang and Hong Zhang*

Transportation Institute of Inner Mongolia University, No. 49, Xilingol South Road, Yuquan District, Hohhot, China

Abstract

Unmanned aerial vehicle (UAV) image transmission commonly relies on JPEG because of its interoperability and low deployment cost, yet compression can weaken structures used by downstream object detectors. This study presents JPEG-compatible receiver-side residual refinement (JCRR), a post-decoding pipeline that preserves the transmitter, JPEG syntax, and transmitted payload. JCRR comprises bounded residual reconstruction (BRR) and structure-prior residual refinement (SPRR), which adds a normalized Sobel edge prior. Both variants refine JPEG-decoded images before a fixed YOLO26s detector. An additional robustness check was conducted with a separately trained YOLO26n detector without retraining the receiver-side refinement models. Experiments used 548 VisDrone validation images at JPEG Q15, Q20, and Q25. BRR and SPRR were each trained with three seeds, and Artifact Reduction Convolutional Neural Network (ARCNN) was trained as an external post-decoding baseline under the same Q20-pair protocol. At the same transmitted JPEG payload, BRR increased mAP50 by 1.67–2.14 percentage points and SPRR by 1.67–2.22 percentage points. Under strict size bins defined after 640 × 640 letterboxing, SPRR improved small-target matched-target recall by 0.85–1.18 percentage points over JPEG; seed-specific paired bootstrap intervals were positive at every operating point. On the 300-image Q20 reconstruction holdout, SPRR achieved 37.75 ± 0.02 dB PSNR and 0.9779 ± 0.0002 SSIM, compared with 37.18 dB and 0.9763 for JPEG. Under the FP32 receiver-side forward-only timing protocol, BRR and SPRR processed images at approximately 62.0 and 61.6 frames per second, respectively, excluding JPEG decoding, detector inference, communication, and file I/O. WebP remained competitive at nearby, rather than bitrate-matched, operating points. The results support JCRR as a compatibility-oriented receiver-side enhancement for UAV systems that must retain JPEG transmission, while not implying bitrate reduction or end-to-end real-time flight deployment.

Keywords: Unmanned aerial vehicle image transmission; JPEG-compatible refinement; Receiver-side processing; Object detection; Small-target detection; Machine vision compression

***Corresponding author:**
Hong Zhang
(imu_hongzhang@outlook.com)

Citation: Wang L, Zhang H. Designing a JPEG-compatible receiver-side residual refinement pipeline for unmanned aerial vehicle image transmission and object detection. *Design+*. doi: 10.36922/DP026260029

Received: June 28, 2026

Revised: July 16, 2026

Accepted: August 6, 2026

Published online: August 24, 2026

Copyright: © 2026 Author(s). This is an Open-Access article distributed under the terms of the Creative Commons AttributionNonCommercial 4.0 International (CC BY-NC 4.0), which permits all non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Publisher's Note: AccScience Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

Unmanned aerial vehicles (UAVs) have become important platforms for traffic observation, infrastructure inspection, environmental sensing, public-safety monitoring, and remote visual analytics. In these applications, images or video frames are commonly acquired at a front-end platform and transmitted to a receiver for subsequent interpretation by detection, tracking, segmentation, or other machine-vision models. The effectiveness of this pipeline depends not only on the downstream model but also on the properties of the transmitted visual signal. UAV imagery is particularly challenging because targets are often small, densely distributed, partially occluded, and embedded in complex backgrounds. These characteristics have motivated dedicated UAV vision studies and benchmark datasets, including VisDrone, large-scale aerial-object benchmarks, and maritime drone datasets.¹⁻⁵

Alongside conventional visual transmission, semantic communication studies illustrate an alternative paradigm in which task-relevant representations are transmitted instead of full image signals.⁶ However, many operational image links still rely on conventional image formats, particularly JPEG. JPEG remains widely adopted because of its interoperability, mature software ecosystem, and low deployment cost.⁷ Its quantization process can weaken fine boundaries, low-contrast target contours, and local structures that may be important for machine analysis. UAV pipeline studies have shown that image degradation can influence object-detection performance, particularly when targets are small or scenes are crowded.² This motivates the broader question of how imaging, coding, and transmission decisions should be evaluated when the final consumer of an image is a machine rather than a human observer.

Fast object detectors make receiver-side UAV analytics practically attractive, but detector speed alone does not remove the effect of transmission-induced image degradation. Prior UAV detection-pipeline studies have therefore evaluated both recognition accuracy and throughput, and modern YOLO-family detectors are commonly used as fast downstream evaluators in such pipelines.^{2,8} In this study, the detector is not redesigned; instead, our research question is whether a receiver-side refinement stage can improve the visual input supplied to a fixed fast detector under an unchanged JPEG payload.

Earlier work has shown that computer-vision performance can be influenced by the image formation and acquisition pipeline. Reconfigurable imaging systems, optimized camera designs, and detector-aware acquisition strategies have demonstrated that the input representation

supplied to a vision model is not neutral and can affect downstream recognition accuracy.⁹⁻¹¹ In the UAV setting, this observation is especially relevant because the captured image may experience compression, transmission, decoding, and resizing before it reaches the detector. Nevertheless, redesigning sensors, camera-processing pipelines, or front-end imaging modules may be difficult when a UAV platform relies on fixed hardware, embedded firmware, legacy software, or established JPEG-based communication procedures.

A growing body of research has explored neural processing around conventional image codecs. Preprocessing-enhanced image compression for machine vision uses learned preprocessing to improve recognition-oriented coding outcomes.¹² Sandwiched image compression wraps neural networks around a standard codec to improve decoded results while maintaining conventional-format interoperability.¹³ Auxiliary codec networks and deep pre-editing approaches likewise investigate how learned modules can enhance standard-compatible compression systems without abandoning existing codecs.^{14,15} Related work has examined perceptual preprocessing, content-adaptive optimization, and decoder-side adaptation around conventional coding systems.¹⁶⁻²¹ These studies establish that conventional codecs can be augmented by learned processing, but many designs intervene before encoding, jointly around the codec, or within a learned compression workflow. Such arrangements may require transmitter-side neural computation, encoder modification, proxy optimization for non-differentiable coding operations, or a revised deployment stack.

Learned image compression and coding-for-machines research has further advanced end-to-end transform coding, scale hyperpriors, autoregressive entropy models, context-adaptive probability estimation, and attention-based designs.²²⁻²⁸ It has also explored representations serving both human and machine vision, feature-oriented coding, task-oriented adaptation, semantic coding, and collaborative video analytics.²⁹⁻⁴⁷ These methods can be valuable when an entire coding system can be redesigned, jointly optimized, or trained with task-level objectives. Their assumptions differ from a conventional UAV image link in which standard JPEG encoding and the transmitted file format must remain unchanged.

This study addresses the narrower setting in which a UAV image must retain a standard JPEG transmitter and a conventional JPEG payload. The question is not whether a new codec can outperform JPEG under a redesigned communication system. Instead, it is whether a lightweight receiver-side refinement stage can improve downstream

object detection after standard JPEG decoding, without modifying the JPEG bitstream, adding transmitted bits, or introducing detector-guided training. The proposed approach is a compatibility-oriented refinement layer rather than a learned codec, semantic communication system, or end-to-end rate-task optimization framework.

To address this question, we propose a JPEG-Compatible Receiver-Side Residual Refinement (JCRR) pipeline for UAV image transmission and object detection. The transmitter follows a standard JPEG workflow and transmits a conventional JPEG file. At the receiver, the JPEG-decoded image is refined before evaluation by a fixed YOLO26s detector. Two controlled variants are investigated. The first, bounded residual reconstruction (BRR), predicts a constrained image residual. The second, structure-prior residual refinement (SPRR), extends the same residual framework with a normalized Sobel edge prior. The final evaluation includes an external Artifact Reduction Convolutional Neural Network (ARCNN) post-decoding artifact-removal baseline, three-seed training, strict size-bin analysis, paired bootstrap confidence intervals, reconstruction metrics, forward-only receiver-side timing, an additional YOLO26n detector robustness check, and explicit implementation metadata.

The study makes four contributions. First, it presents a JPEG-compatible receiver-side refinement design that leaves the transmitter workflow and transmitted JPEG bits per pixel unchanged. Second, it compares bounded residual reconstruction with a structure-prior extension under the same backbone and objective. Third, it evaluates both variants against a classic post-decoding artifact-removal baseline at three JPEG operating points using global detection metrics, three-seed variability, strict

size-bin recall, and an additional fixed YOLO26n detector robustness check. Fourth, it characterizes reconstruction quality, forward-only receiver-side speed, codec settings, reproducibility materials, and the operating boundary relative to nearby WebP points. The aim is not to claim universal superiority over alternative codecs or restoration methods, but to establish the value and limits of receiver-side refinement when an existing JPEG-based UAV transmission link must remain unchanged.

2. Materials and methods

2.1. Design objectives and system architecture

This study considers a practical UAV image-transmission setting in which the transmitter already relies on a conventional JPEG workflow. Rather than replacing the codec or modifying the transmitted bitstream, JCRR operates after standard JPEG decoding and before downstream object detection. The objective is to preserve detection-relevant structure under fixed transmitted JPEG payloads, with attention to compact aerial objects.

The design problem is therefore not to invent a new codec, but to reconfigure the receiver-side processing stage under three practical constraints: the transmitter should remain unchanged, the transmitted JPEG payload should define the communication cost, and the downstream detector should remain frozen during refinement training. This framing links the proposed method to a compatibility-constrained design problem rather than to a purely incremental restoration-network modification.

As illustrated in [Figure 1](#), a source reference image is encoded at the transmitter using JPEG and transmitted as a conventional JPEG file. At the receiver, the file is decoded

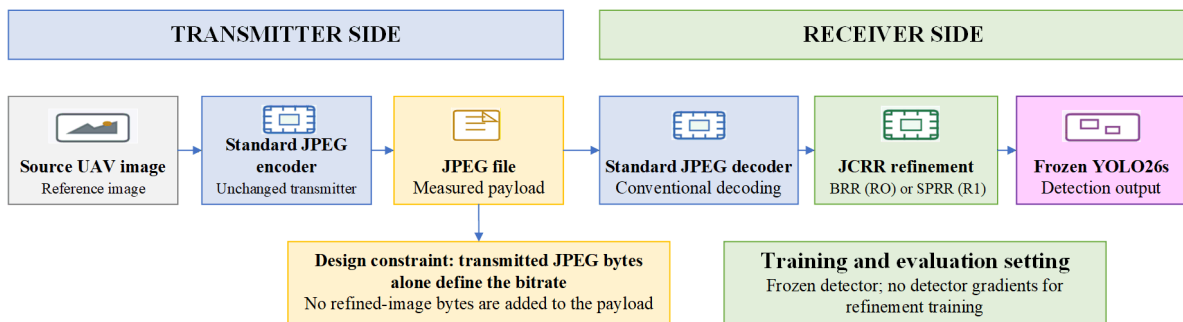


Figure 1. JPEG-compatible receiver-side refinement framework. A source UAV image is encoded and transmitted as a conventional JPEG file. After standard JPEG decoding, BRR (R0) or SPRR (R1) refines the received image before downstream evaluation by the fixed YOLO26s detector. Bpp is calculated from the transmitted JPEG file only; no refined-image bytes are added to the communication payload.

Abbreviations: Bpp: Bits per pixel; BRR: Bounded residual reconstruction; JCRR: JPEG-compatible receiver-side residual refinement; SPRR: Structure-prior residual refinement; UAV: Unmanned aerial vehicle.

by a standard JPEG decoder and processed by BRR (R0) or SPRR (R1) before the refined image is supplied to a fixed YOLO26s detector. No neural network is inserted at the transmitter, and no modification is made to JPEG syntax, coding, or payload.

The refinement stage is separated from the detector. The detector is a fixed downstream evaluator; no detector gradients, detection losses, feature-alignment losses, or task-saliency losses are used during refinement training. Let I denote a source reference image and let $X_{J,q}$ denote its JPEG-decoded counterpart at quality factor q . The receiver-side model f_θ produces:

$$\hat{X}_{J,q} = f_\theta(X_{J,q}), X_{J,q} = D_q(E_q(I)) \quad (1)$$

where the refined image is used for detector evaluation. The transmitted information remains the JPEG file only; therefore, the refined output is not counted as an additional communication payload.

For JPEG and WebP, transmitted bits per pixel (Bpp) for image i were calculated from the actual encoded file size:

$$Bpp_i = \frac{8B_i}{W_i H_i} \quad (2)$$

where B_i is the encoded size in bytes and I and I_i are the original image width and height. The reported $\overline{Bpp}$ at each operating point is the mean across all evaluated images:

$$\overline{Bpp} = \frac{1}{N} \sum_{i=1}^N Bpp_i \quad (3)$$

Locally generated refined images were excluded from Bpp calculations.

2.2. Bounded residual reconstruction

BRR receives the three-channel JPEG-decoded image X_J and predicts a residual image $R_\theta(X_J)$. Its bounded output is:

$$\hat{X} = \text{clip}(X_J + 0.20 \tanh(R_\theta(X_J)), 0, 1) \quad (4)$$

The hyperbolic tangent and fixed 0.20 scale constrain the residual correction and reduce the risk of excessive color shifts or artificial high-frequency content. BRR uses a convolutional stem, four depthwise residual blocks, a lightweight channel gate, and a three-channel reconstruction head. The formal model width is 32 channels. BRR contains 22,307 trainable parameters. Figure 2 summarizes the architecture.

2.3. Structure-prior residual refinement

SPRR extends BRR with an explicit edge prior derived from X_J . A grayscale image is computed as:

$$Y_J = 0.299R_J + 0.587G_J + 0.114B_J \quad (5)$$

The horizontal and vertical Sobel responses are:

$$G_x = K_x * Y_J, G_y = K_y * Y_J \quad (6)$$

where K_x and K_y are conventional 3×3 Sobel kernels. The edge magnitude is:

$$E_{u,v} = \sqrt{G_{x,u,v}^2 + G_{y,u,v}^2} \quad (7)$$

The map is normalized independently within each image as:

$$\tilde{E}_{u,v} = \frac{E_{u,v}}{\max_{u,v} E_{u,v} + \epsilon}, \epsilon = 10^{-6} \quad (8)$$

SPRR concatenates the normalized edge map with X_J :

$$Z = \text{concat}(X_J, \tilde{E}) \quad (9)$$

The resulting four-channel input is processed by the same stem, four depthwise residual blocks, channel gate, and reconstruction head used in BRR. The output is:

$$\hat{X} = \text{clip}(X_J + 0.20 \tanh(R_\theta(Z)), 0, 1) \quad (10)$$

The Sobel prior is an input-level augmentation rather than an attention branch, semantic module, or detector-guided pathway. SPRR contains 22,595 trainable parameters, only 288 more than BRR. Both variants use the same reconstruction objective. The pixel term is a channel-normalized Charbonnier loss:⁴⁸

$$\mathcal{L}_{\text{rec}} = \frac{1}{3HW} \sum_{c,u,v} \sqrt{(\hat{X}_{c,u,v} - X_{c,u,v})^2 + \epsilon^2}, \epsilon = 10^{-3} \quad (11)$$

An edge-consistency term is shared by both variants:

$$\mathcal{L}_{\text{edge}} = \frac{1}{HW} \sum_{u,v} |S(\hat{X})_{u,v} - S(X)_{u,v}| \quad (12)$$

where $S(\cdot)$ denotes the Sobel edge-magnitude operator and I is the corresponding source reference image. The total objective is:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + 0.10 \mathcal{L}_{\text{edge}} \quad (13)$$

Thus, the edge-consistency term is shared by BRR and SPRR, whereas the normalized Sobel prior is specific to SPRR.

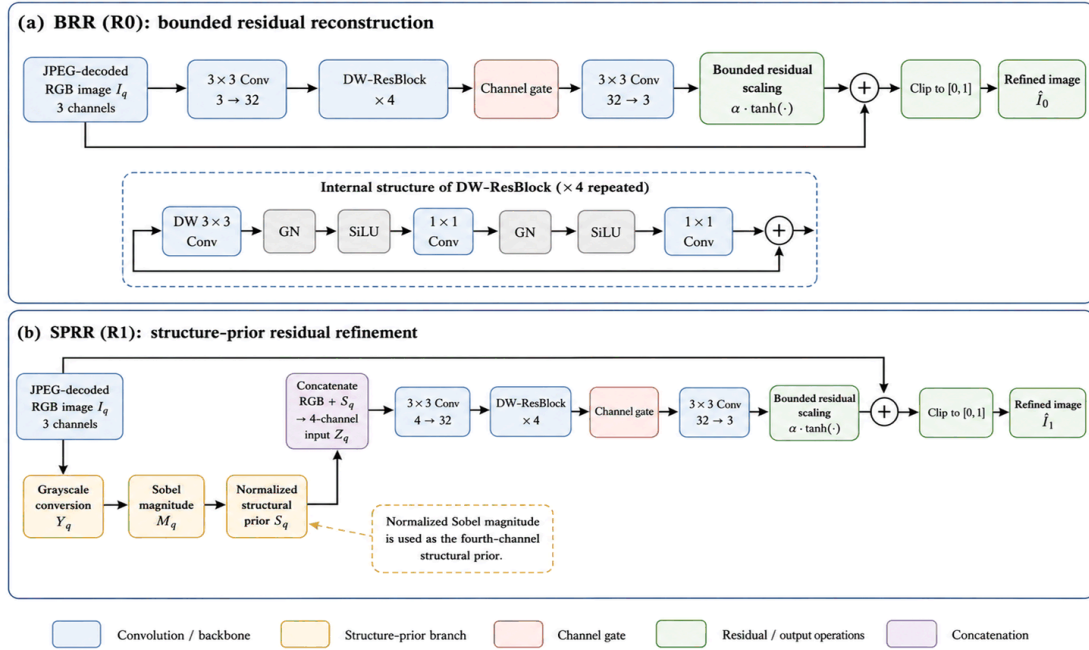


Figure 2. Architectures of bounded residual reconstruction and structure-prior residual refinement. (A) BRR receives the JPEG-decoded RGB image and predicts a bounded residual correction through the residual backbone, channel gate, and reconstruction head. (B) SPRR augments the RGB input with a normalized Sobel structural prior while retaining the same residual backbone, channel gate, and bounded output design. Both variants use the same reconstruction and edge-consistency supervision.

Abbreviations: BRR: Bounded residual reconstruction; SPRR: Structure-prior residual refinement.

2.4. Dataset, baselines, and evaluation protocol

Experiments used the VisDrone benchmark³, with its 548-image validation split for formal detection evaluation. The configuration contains 10 categories: pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, and motor. The 38,759 annotated targets are referred to as objects in reference images; no claim is made that the source images are uncompressed.

BRR and SPRR were trained on the VisDrone training split. A deterministic split generated with seed 42 reserved 300 images as a reconstruction holdout and used the remaining 6,171 images for optimization. Training pairs were generated online by an in-memory JPEG round trip at Q20, followed by bicubic letterboxing to 640×640 . Each model was trained independently with seeds 42, 123, and 2026 for 12 epochs using AdamW⁴⁹, batch size 8, learning rate 2×10^{-4} , weight decay 1×10^{-4} , and eight data-loader workers. The same fixed training and holdout partitions were used for all seeds.

An external post-decoding JPEG artifact-removal baseline was implemented with the four-layer ARCNN architecture of Dong *et al.*⁵⁰: 9×9 feature extraction (3 to 64 channels), 7×7 enhancement (64 to 32), 1×1 mapping (32 to 16), and 5×5 reconstruction (16 to 3). ARCNN was

trained on the same 6,171 Q20 pairs under the same three seeds for 40 epochs using Adam, batch size 24, learning rate 1×10^{-4} , and mean squared error; the checkpoint with the lowest reconstruction-holdout mean squared error (MSE) was retained for each seed. ARCNN contains 117,731 trainable parameters. It serves as a matched-data external receiver-side restoration baseline rather than an exhaustive comparison with all artifact-removal methods.

JPEG files were generated with Pillow 12.2.0 in RGB mode using quality values 15, 20, and 25, optimize = True, progressive = False, and 4:2:0 chroma subsampling (Pillow subsampling value 2). WebP was encoded with lossless = False and method = 6. For the Q15, Q20, and Q25 JPEG references, WebP quality values 15, 25, and 36 were selected after calibration on 80 images by searching quality values 1-100 and minimizing the absolute difference in mean Bpp. Accordingly, WebP is reported only as a nearby operating-point comparison, not as a bitrate-matched codec comparison.

All source-reference, JPEG, ARCNN, BRR, SPRR, and WebP images were evaluated with the same fixed YOLO26s checkpoint.⁸ The checkpoint was fine-tuned on the VisDrone training split for 100 epochs at 640 pixels with batch size 16, seed 0, automatic optimizer selection,

mixed-precision training, and the recorded Ultralytics augmentation schedule. Evaluation used Ultralytics v8.4.24, input size 640, batch size 16, and the same detector checkpoint for every method.

To address detector-level dependence, an additional robustness check was conducted with a separately trained YOLO26n detector. The second detector was trained on the same VisDrone YOLO data configuration from the YOLO26n initialization for 80 epochs and then frozen for evaluation. The ARCNN, BRR, and SPRR restoration checkpoints were not retrained; only the downstream detector used for evaluation was changed. The same Q15, Q20, and Q25 JPEG operating points and the same WebP-nearby settings were used.

Primary metrics were precision, recall, mAP at IoU 0.5 (mAP50), and mAP averaged from IoU 0.5 to 0.95 (mAP50-95). Strict size-bin statistics were computed after detector letterboxing to 640×640 : small objects had area below 32^2 pixels, medium objects had area from 32^2 to below 96^2 pixels, and large objects had area at least 96^2 pixels. The validation set contained 35,827 small, 2,897 medium, and 35 large objects. A ground-truth object was counted as matched when a same-class prediction at confidence at least 0.25 achieved IoU at least 0.5; each prediction was assigned to at most one ground-truth object. For each trained checkpoint, paired image-resampled bootstrap intervals used 2,000 resamples. The base random-number-generator seed was 20260626, and seed-specific bootstrap streams were initialized as 20260626 plus the corresponding training seed. These intervals quantify image-level uncertainty conditional on the fixed trained checkpoint and do not capture training-seed variability. Table 1 summarizes the complete protocol.

For reproducibility, the Supplementary File provides the training and evaluation scripts, deterministic split information, JPEG/WebP encoding configuration, aggregated CSV metrics, checkpoint identifiers, and reproduction metadata. The full detector checkpoint hashes are reported in the Supplementary File. The archive does not redistribute VisDrone image files, JPEG/WebP derivatives, or restored image variants.

2.5. Implementation, reconstruction quality, and timing

Training and evaluation were conducted in Python 3.10.20 with PyTorch 2.5.1+cu124 on an NVIDIA GeForce RTX 4090 D GPU. Q20 reconstruction quality was measured on the 300-image holdout after 640×640 letterboxing using peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). These reconstruction measurements complement, rather than replace, the detection-oriented evaluation.

Table 1. Experimental setup and reproducibility protocol

Item	Final setting
Dataset and validation set	VisDrone2019-DET; 548 validation images; 38,759 annotated targets across 10 categories.
R0/R1 training data	Deterministic seed-42 split: 6,171 Q20 training pairs and 300 reconstruction-holdout images; online JPEG round trip and 640×640 letterboxing.
R0/R1 training	Three seeds (42, 123, 2026); 12 epochs; AdamW; batch size 8; learning rate 2×10^{-4} ; weight decay 1×10^{-4} .
ARCNN external baseline	Four-layer ARCNN; same 6,171 Q20 pairs; three seeds; 40 epochs; Adam; batch size 24; learning rate 1×10^{-4} ; best holdout-MSE checkpoint retained.
JPEG encoding	Pillow 12.2.0; RGB; Q15/Q20/Q25; optimize = True; progressive = False; 4:2:0 chroma subsampling.
WebP comparison	lossless = False; method = 6; qualities 15/25/36 for Q15/Q20/Q25 references; calibrated on 80 images; nearby operating points only.
Primary detector	One fixed YOLO26s checkpoint fine-tuned on the VisDrone training split; 100 epochs; 640 input; evaluation batch size 16; identical checkpoint for all variants. The detector was used only for evaluation and was not updated during refinement training.
Additional detector robustness check	A separately trained YOLO26n detector was used as a second fixed downstream detector. The BRR, SPRR, and ARCNN checkpoints were not retrained; only the detector used for evaluation was changed.
Strict size bins	After 640×640 letterbox: small $< 32^2$ px; medium 32^2 to $< 96^2$ px; large $\geq 96^2$ px.
Statistical reporting	JPEG/WebP/source-reference: fixed results. ARCNN/BRR/SPRR: mean $\pm$ SD across three seeds. Paired bootstrap: 2,000 image-resampled iterations; base RNG seed 20260626, with seed-specific streams initialized as 20260626 plus the training seed; conditional on each fixed checkpoint.
Reconstruction and timing	PSNR/SSIM on the 300-image Q20 holdout. FP32 forward-only restoration timing used 20 warm-up images and 120 measurements on preloaded GPU tensors; reported FPS is model-forward restoration FPS and excludes JPEG decoding, data transfer, detector inference, communication, and file I/O.
Environment	Python 3.10.20; PyTorch 2.5.1+cu124; Ultralytics v8.4.24; NVIDIA GeForce RTX 4090 D GPU.
Reproducibility materials	The Supplementary File includes training and evaluation scripts, deterministic split information, JPEG/WebP encoding settings, trained BRR/SPRR/ARCNN checkpoints, detector checkpoint identifiers, aggregated metrics, and reproduction metadata. VisDrone image files and derivative image sets are not redistributed.

Abbreviations: ARCNN: Artifact reduction convolutional neural network; Bpp: Bits per pixel; BRR: Bounded residual reconstruction; FPS: Frames per second; JPEG: Joint Photographic Experts Group; MSE: Mean squared error; R0/R1: BRR/SPRR variants; SPRR: Structure-prior residual refinement.

Forward-only restoration timing used FP32, 20 warm-up images, and 120 measured images with preloaded GPU tensors. CUDA synchronization was applied around the forward call. The measurement includes the Sobel-prior computation inside the SPRR forward pass but excludes JPEG decoding, letterboxing, host-to-device transfer, resize-back, file I/O, detector inference, communications, and wireless latency. ARCNN has no Sobel prior. The reported frames per second (FPS) is therefore receiver-side restoration model-forward FPS rather than end-to-end UAV-system FPS. JCRR is not proposed as a standalone fast object detector; it is a receiver-side refinement stage that can be paired with a fast detector when receiver-side computing resources are available.

3. Results

3.1. Rate-detection performance at fixed JPEG payloads

Table 2 and Figure 3 report the final global detection results. JPEG, WebP, and source-reference values are fixed results, whereas ARCNN, BRR, and SPRR are reported as mean $\pm$ standard deviation (SD) across three training seeds. At the same transmitted JPEG Bpp, BRR increased mAP50 by 2.00, 1.67, and 2.14 percentage points at Q15, Q20, and Q25, respectively. SPRR increased mAP50 by 2.03, 1.67, and 2.22 percentage points at the same operating points. The gains remained evident in mAP50-95: BRR improved the fixed JPEG result by 1.41, 1.36, and 1.73 percentage points, while SPRR improved it by 1.42, 1.35, and 1.78 percentage points.

The three-seed variation of BRR and SPRR was small relative to the observed improvement over JPEG. For example, at Q20, BRR obtained mAP50 of 0.3889 ± 0.0003 and SPRR obtained 0.3889 ± 0.0005 , compared with 0.3722 for JPEG. At Q25, BRR and SPRR reached 0.3950 ± 0.0007 and 0.3958 ± 0.0007 , respectively, compared with 0.3737 for JPEG. These findings indicate stable detection preservation gains under the fixed JPEG-payload condition, while the global difference between BRR and SPRR remains limited.

3.2. External JPEG artifact-removal baseline

ARCNN improved mAP50 over JPEG at every evaluated quality, by 1.76, 1.59, and 1.94 percentage points at Q15, Q20, and Q25, respectively. Thus, a conventional receiver-side artifact-removal network can improve the fixed JPEG detection baseline under matched training data. However, ARCNN remained below BRR and SPRR in mAP50 and mAP50-95 at each same-payload JPEG point. At Q20, ARCNN reached mAP50 of 0.3880 ± 0.0003 , compared with 0.3889 ± 0.0003 for BRR and 0.3889 ± 0.0005 for SPRR.

The comparison is intentionally narrow. ARCNN was trained with the same Q20 image-pair protocol and three seeds, with a holdout-selected checkpoint, to provide an external post-decoding baseline. The results do not imply that all residual architectures outperform all artifact-removal networks; they show that the proposed bounded-residual designs were more effective than this implemented four-layer ARCNN baseline under the present data, detector, and training protocol.

3.3. Strict size-bin analysis and conditional bootstrap evidence

Table 3 and Figure 4 report strict size-bin results after 640×640 letterboxing. The strict-small bin contains 35,827 targets and therefore captures the dominant compact-object regime of this validation set. Relative to JPEG, BRR improved strict-small matched-target recall by 0.93, 0.64, and 1.00 percentage points at Q15, Q20, and Q25. SPRR improved the same metric by 1.08, 0.85, and 1.18 percentage points. ARCNN also improved strict-small recall on average, although its Q20 gain was comparatively small at 0.20 percentage points.

Across the three training seeds, SPRR exceeded BRR in strict-small matched-target recall by 0.15, 0.21, and 0.18 percentage points at Q15, Q20, and Q25, respectively. This is a small but consistent effect; it should not be generalized into a decisive global-accuracy advantage because the BRR-SPRR mAP50 differences were limited. For the SPRR-versus-JPEG comparison, all nine seed-specific paired image-resampled bootstrap intervals had positive lower bounds. At Q20, the observed gains were +0.94 pp (95% CI 0.67 to 1.23), +0.78 pp (0.49 to 1.06), and +0.85 pp (0.57 to 1.13) for seeds 42, 123, and 2026, respectively. These intervals are conditional on the particular trained checkpoints and complement, rather than replace, the three-seed mean $\pm$ SD results.

Medium-object recall was already high for JPEG and changed modestly across learned methods. The large-object bin contains only 35 annotations; accordingly, its apparent differences are descriptive and are not interpreted as stable evidence for large-object behavior.

3.4. WebP nearby operating points

WebP was evaluated as a practical conventional-format alternative. Its Q15-, Q20-, and Q25-referenced operating points used 0.3017, 0.3919, and 0.4924 Bpp, respectively, which are lower than the corresponding JPEG reference points. At those nearby points, WebP achieved mAP50 values of 0.3783, 0.3887, and 0.3924. It was stronger than the JCRR variants at the low-rate nearby point, close to them at the middle point, and below SPRR at the high-rate nearby point.

Table 2. Global detection performance on the VisDrone validation set

Quality	Method	Bpp	Precision	Recall	mAP50	mAP50-95	Δ mAP50 vs JPEG (pp)
Reference	Source-reference	—	0.5206	0.3923	0.4083	0.2424	—
Q15	JPEG	0.3190	0.4698	0.3476	0.3566	0.2066	—
Q15	WebP (nearby)	0.3017	0.4934	0.3653	0.3783	0.2220	—
Q15	ARCNN	0.3190	0.4713 ± 0.0014	0.3659 ± 0.0012	0.3742 ± 0.0004	0.2192 ± 0.0003	+1.76
Q15	BRR (R0)	0.3190	0.4758 ± 0.0047	0.3690 ± 0.0023	0.3766 ± 0.0003	0.2207 ± 0.0002	+2.00
Q15	SPRR (R1)	0.3190	0.4734 ± 0.0013	0.3699 ± 0.0007	0.3768 ± 0.0003	0.2208 ± 0.0002	+2.03
Q20	JPEG	0.4085	0.4883	0.3639	0.3722	0.2176	—
Q20	WebP (nearby)	0.3919	0.4998	0.3732	0.3887	0.2278	—
Q20	ARCNN	0.4085	0.5040 ± 0.0018	0.3737 ± 0.0015	0.3880 ± 0.0003	0.2304 ± 0.0005	+1.59
Q20	BRR (R0)	0.4085	0.5044 ± 0.0053	0.3746 ± 0.0031	0.3889 ± 0.0003	0.2313 ± 0.0005	+1.67
Q20	SPRR (R1)	0.4085	0.5025 ± 0.0022	0.3749 ± 0.0007	0.3889 ± 0.0005	0.2311 ± 0.0001	+1.67
Q25	JPEG	0.5192	0.4897	0.3676	0.3737	0.2176	—
Q25	WebP (nearby)	0.4924	0.5019	0.3766	0.3924	0.2316	—
Q25	ARCNN	0.5192	0.5098 ± 0.0046	0.3795 ± 0.0017	0.3930 ± 0.0008	0.2340 ± 0.0006	+1.94
Q25	BRR (R0)	0.5192	0.5121 ± 0.0031	0.3827 ± 0.0017	0.3950 ± 0.0007	0.2349 ± 0.0005	+2.14
Q25	SPRR (R1)	0.5192	0.5105 ± 0.0036	0.3840 ± 0.0009	0.3958 ± 0.0007	0.2353 ± 0.0002	+2.22

Notes: Values for ARCNN, BRR, and SPRR are mean $\pm$ SD across three training seeds. JPEG, WebP, and source-reference values are fixed results. WebP represents a nearby operating-point comparison rather than an exact bitrate-matched comparison.

Abbreviations: ARCNN: Artifact reduction convolutional neural network; Bpp: Bits per pixel; BRR: Bounded residual reconstruction; IoU: Intersection over union; mAP: Mean average precision; SPRR: Structure-prior residual refinement.

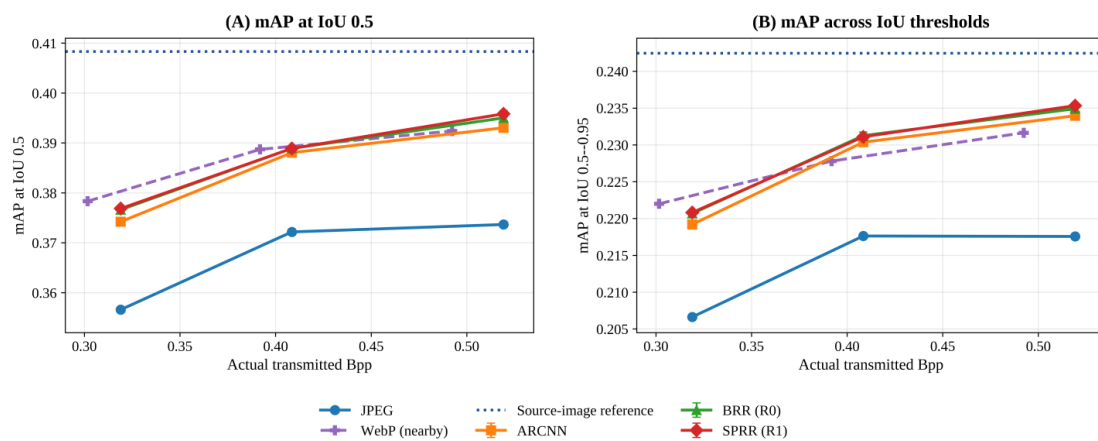


Figure 3. Rate-detection performance at actual transmitted operating points. (A) mAP at IoU 0.5. (B) mAP averaged over IoU thresholds from 0.5 to 0.95. Symbols and error bars for ARCNN, BRR, and SPRR show mean $\pm$ SD across three training seeds. JPEG, WebP, and source-reference results are fixed. WebP is shown at nearby operating points, not as an exact bitrate-matched comparison.

Abbreviations: ARCNN: Artifact reduction convolutional neural network; Bpp: Bits per pixel; BRR: Bounded residual reconstruction; IoU: Intersection over union; mAP: Mean average precision; SPRR: Structure-prior residual refinement.

Table 3. Strict size-bin matched-target recall and paired bootstrap evidence

(A) Strict matched-target recall by size bin after 640 × 640 letterboxing				
Quality	Method	Small recall	Medium recall	Large recall
Q15	JPEG	0.3980	0.8153	0.6571
Q15	ARCNN	0.4030 ± 0.0014	0.8207 ± 0.0017	0.6857 ± 0.0000
Q15	BRR (R0)	0.4073 ± 0.0010	0.8245 ± 0.0024	0.7048 ± 0.0165
Q15	SPRR (R1)	0.4088 ± 0.0009	0.8243 ± 0.0003	0.7048 ± 0.0165
Q20	JPEG	0.4162	0.8253	0.6571
Q20	ARCNN	0.4182 ± 0.0015	0.8334 ± 0.0008	0.7143 ± 0.0000
Q20	BRR (R0)	0.4226 ± 0.0010	0.8348 ± 0.0008	0.7048 ± 0.0165
Q20	SPRR (R1)	0.4247 ± 0.0008	0.8352 ± 0.0009	0.7143 ± 0.0000
Q25	JPEG	0.4170	0.8212	0.6571
Q25	ARCNN	0.4227 ± 0.0011	0.8362 ± 0.0014	0.6952 ± 0.0165
Q25	BRR (R0)	0.4270 ± 0.0017	0.8380 ± 0.0022	0.7143 ± 0.0000
Q25	SPRR (R1)	0.4288 ± 0.0000	0.8378 ± 0.0024	0.7143 ± 0.0000
(B) SPRR-JPEG paired image-resampled bootstrap intervals for strict-small matched-target recall				
Quality	Seed	Δ recall (pp)	95% CI (pp)	
Q15	42	+1.18	0.92 to 1.44	
Q15	123	+1.03	0.78 to 1.28	
Q15	2026	+1.04	0.77 to 1.29	
Q20	42	+0.94	0.67 to 1.23	
Q20	123	+0.78	0.49 to 1.06	
Q20	2026	+0.85	0.57 to 1.13	
Q25	42	+1.18	0.87 to 1.46	
Q25	123	+1.18	0.89 to 1.50	
Q25	2026	+1.18	0.88 to 1.49	

Notes: Small, medium, and large denote areas < 32² px, 32² to <96² px, and ≥96² px, respectively, after letterboxing. Values for ARCNN, BRR, and SPRR in panel A are mean ± SD across three seeds. Panel B intervals are paired image-resampled bootstrap intervals conditional on each fixed trained checkpoint and do not represent training-seed variability.

Abbreviations: ARCNN: Artifact reduction convolutional neural network; BRR: Bounded residual reconstruction; CI: Confidence interval; IoU: Intersection over union; pp: Percentage points; JPEG: Joint Photographic Experts Group; SPRR: Structure-prior residual refinement.

These results are not a strict codec ranking because the WebP operating points are not exactly bitrate matched to JPEG. The purpose of the comparison is to define the boundary of the claim: JCRR improves a retained JPEG workflow at the receiver; it is not presented as a universal replacement for alternative image formats when transmitter-side format choice is unconstrained.

3.5. Reconstruction quality and forward-only restoration time

Table 4 and Figure 4 summarize Q20 reconstruction fidelity. JPEG achieved 37.18 dB peak signal-to-noise ratio (PSNR) and 0.9763 structural similarity index measure (SSIM) on the 300-image holdout. ARCNN reached 37.03

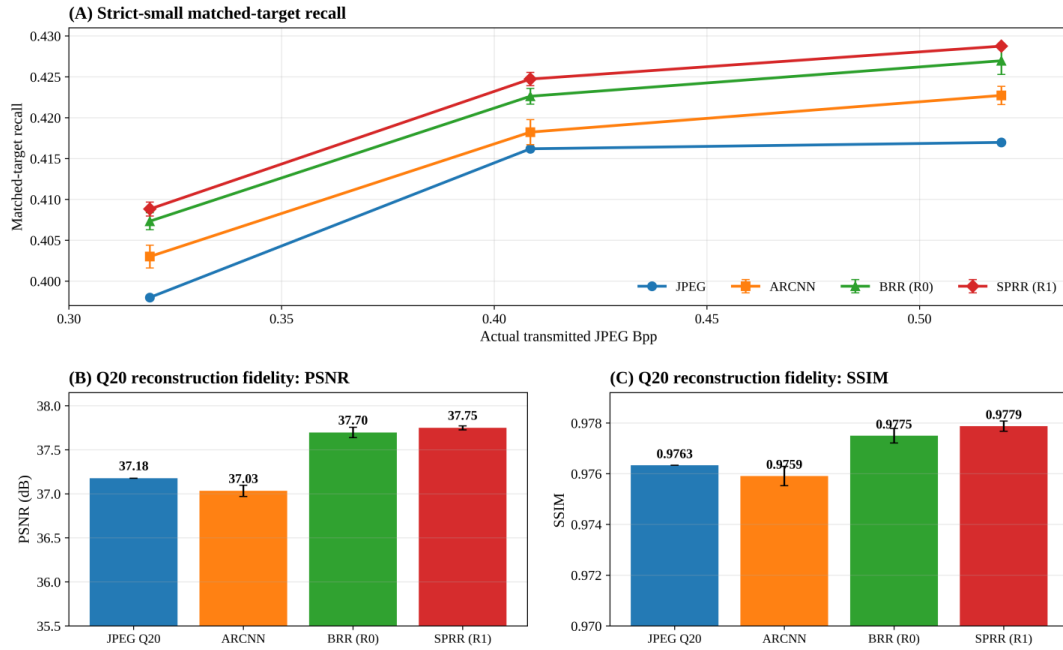


Figure 4. Strict-small preservation and reconstruction fidelity. (A) Strict-small matched-target recall after 640×640 letterboxing. (B) Q20 reconstruction peak signal-to-noise ratio. (C) Q20 structural similarity index measure. Error bars for learned methods show mean $\pm$ SD across three training seeds. Strict-small objects have areas below 32^2 pixels after letterboxing. Per-seed paired bootstrap confidence intervals are reported in Table 3.

Abbreviations: ARCNN: Artifact reduction convolutional neural network; BRR: Bounded residual reconstruction; JPEG: Joint Photographic Experts Group; PSNR: Peak signal-to-noise ratio; SPRR: Structure-prior residual refinement; SSIM: Structural similarity index measure.

± 0.06 dB and 0.9759 ± 0.0004 , whereas BRR reached 37.70 ± 0.06 dB and 0.9775 ± 0.0003 . SPRR achieved 37.75 ± 0.02 dB and 0.9779 ± 0.0002 . These measurements show that the bounded-residual variants improved the selected reference-image fidelity metrics under the Q20 protocol; they do not constitute a comprehensive human-perceptual assessment.

Under the explicitly bounded FP32 forward-only protocol, ARCNN required 2.93 ± 0.00 ms per image, corresponding to approximately 341.3 FPS. BRR required 16.12 ± 0.02 ms, corresponding to approximately 62.0 FPS, and SPRR required 16.24 ± 0.03 ms, corresponding to approximately 61.6 FPS on the RTX 4090 D. The 0.12 ms BRR-SPRR difference includes Sobel-prior computation for SPRR. These figures exclude all decoding, transfer, detector, storage, communication, and network components specified in section 2.5 and therefore should not be interpreted as end-to-end system latency or as evidence of embedded UAV deployment.

3.6. Additional detector robustness check

To address detector-level dependence, an additional robustness check was conducted using a separately trained YOLO26n detector while keeping the ARCNN, BRR,

and SPRR restoration checkpoints unchanged. Table 5 reports the second-detector results at Q15, Q20, and Q25. Under YOLO26n, all three receiver-side learned refinement methods improved mAP50 over JPEG at the same transmitted JPEG Bpp. BRR improved mAP50 by 1.09, 1.15, and 1.17 percentage points at Q15, Q20, and Q25, respectively. SPRR improved mAP50 by 1.10, 1.22, and 1.23 percentage points at the same operating points.

The second-detector results support the interpretation that the observed benefit is not restricted to the primary YOLO26s checkpoint. However, the analysis remains a detector-level robustness check on the same VisDrone validation split and should not be interpreted as cross-dataset generalization. WebP-nearby remained competitive under YOLO26n, reinforcing the same claim boundary as in section 3.4: JCRR improves a retained JPEG workflow at the receiver rather than replacing all alternative image formats.

3.7. Qualitative examples

Figures 5 and 6 provide representative Q20 restoration and detection examples generated with the seed 42 checkpoints. The restoration examples illustrate how the receiver-side outputs preserve visually continuous edges

Table 4. Q20 reconstruction quality and FP32 forward-only receiver-side restoration speed

Method	Params	Checkpoint (MB)	PSNR (dB)	SSIM	Mean forward (ms)	Approx. FPS	P95 (ms)
JPEG Q20	—	—	37.1774	0.9763	—	—	—
ARCNN	117,731	1.36	37.03 $\pm$ 0.06	0.9759 $\pm$ 0.0004	2.93 $\pm$ 0.00	341.3	2.94
BRR (R0)	22,307	0.30	37.70 $\pm$ 0.06	0.9775 $\pm$ 0.0003	16.12 $\pm$ 0.02	62.0	16.17
SPRR (R1)	22,595	0.31	37.75 $\pm$ 0.02	0.9779 $\pm$ 0.0002	16.24 $\pm$ 0.03	61.6	16.29

Notes: PSNR and SSIM were measured on the 300-image Q20 reconstruction holdout after 640×640 letterboxing. Learned-method values are mean $\pm$ SD across three training seeds. Timing uses FP32, 20 warm-up images, and 120 measurements with preloaded GPU tensors. FPS is calculated from the mean forward time and refers only to receiver-side restoration model forwarding. It excludes JPEG decoding, letterboxing, host-to-device transfer, resize-back, file I/O, detector inference, communication, wireless latency, and embedded-device deployment overhead.

Abbreviations: ARCNN: Artifact reduction convolutional neural network; BRR: Bounded residual reconstruction; FP32: 32-bit floating point; FPS: Frames per second; JPEG: Joint Photographic Experts Group; P95: 95th percentile; PSNR: Peak signal-to-noise ratio; SPRR: Structure-prior residual refinement; SSIM: Structural similarity index measure.

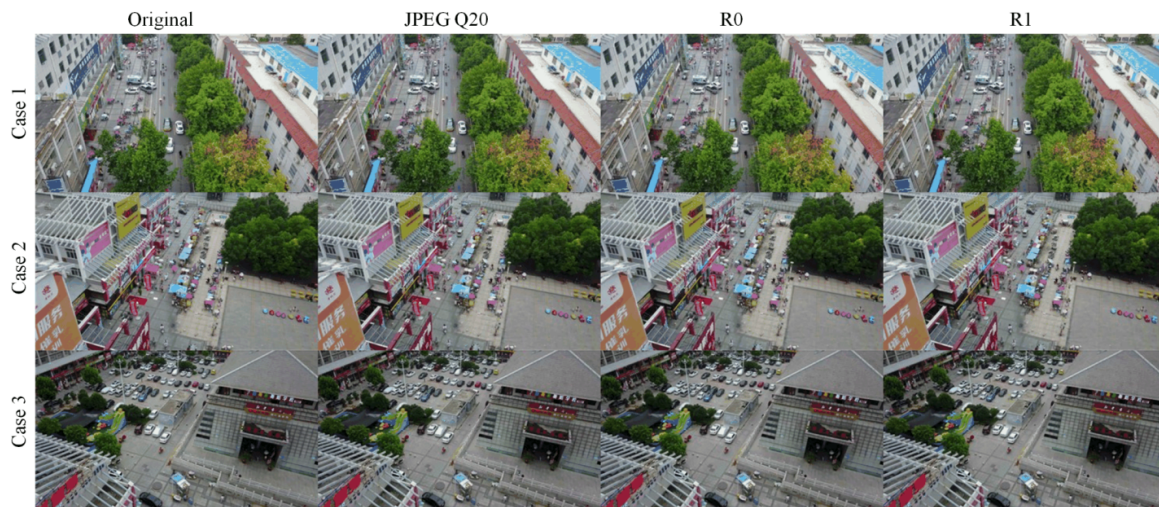


Figure 5. Representative qualitative restoration comparisons at JPEG Q20. Columns show the reference image, JPEG-decoded image, BRR (R0), and SPRR (R1). The examples were generated with the seed 42 checkpoints and are included for illustration only; quantitative conclusions are based on the validation-set results. Images are reproduced from the VisDrone2019-DET validation split for research illustration.

Abbreviations: ARCNN: Artifact reduction convolutional neural network; BRR: Bounded residual reconstruction; JPEG: Joint Photographic Experts Group; SPRR: Structure-prior residual refinement.

and local structures in selected scenes. The detection examples show representative cases in which recovered local content coincides with retained or newly detected compact objects. These panels are illustrative only; all quantitative comparisons and inferential claims are based on the validation-set metrics, size-bin statistics, and bootstrap results reported above.

4. Discussion

4.1. Compatibility-oriented receiver-side workflow design

The central contribution is not a new image codec and not a claim that JCRR lowers bitrate. JCRR inserts a lightweight refinement stage after standard JPEG decoding

Table 5. Additional detector robustness check using YOLO26n

Quality	Method	Bpp	Recall	mAP50	mAP50-95	Δ mAP50 vs JPEG (pp)
Reference	Source-reference	—	0.3201	0.3197	0.1811	—
Q15	JPEG	0.3190	0.2951	0.2874	0.1623	—
Q15	WebP-nearby	0.3017	0.3031	0.2966	0.1667	—
Q15	ARCNN	0.3190	0.3006 ± 0.0039	0.2971 ± 0.0003	0.1684 ± 0.0003	+0.96
Q15	BRR (R0)	0.3190	0.3018 ± 0.0011	0.2983 ± 0.0006	0.1693 ± 0.0002	+1.09
Q15	SPRR (R1)	0.3190	0.2991 ± 0.0015	0.2985 ± 0.0004	0.1693 ± 0.0002	+1.10
Q20	JPEG	0.4085	0.2926	0.2926	0.1647	—
Q20	WebP-nearby	0.3919	0.2998	0.3021	0.1705	—
Q20	ARCNN	0.4085	0.3052 ± 0.0012	0.3033 ± 0.0007	0.1717 ± 0.0002	+1.08
Q20	BRR (R0)	0.4085	0.3077 ± 0.0018	0.3041 ± 0.0004	0.1721 ± 0.0002	+1.15
Q20	SPRR (R1)	0.4085	0.3081 ± 0.0009	0.3048 ± 0.0002	0.1725 ± 0.0001	+1.22
Q25	JPEG	0.5192	0.2996	0.2975	0.1674	—
Q25	WebP-nearby	0.4924	0.3084	0.3068	0.1738	—
Q25	ARCNN	0.5192	0.3080 ± 0.0013	0.3098 ± 0.0003	0.1757 ± 0.0002	+1.23
Q25	BRR (R0)	0.5192	0.3087 ± 0.0009	0.3092 ± 0.0003	0.1758 ± 0.0002	+1.17
Q25	SPRR (R1)	0.5192	0.3096 ± 0.0014	0.3098 ± 0.0003	0.1759 ± 0.0000	+1.23

Notes: The second detector was a separately trained YOLO26n checkpoint. JPEG, WebP, and source-reference values are fixed results. ARCNN, BRR, and SPRR values are mean $\pm$ SD across the three restoration seeds; the restoration checkpoints were not retrained for this analysis. Δ mAP50 is computed against the JPEG result at the same JPEG quality and transmitted JPEG Bpp. WebP is included as a nearby operating-point comparison and is not used for Δ mAP50 because it is not exactly bitrate matched to JPEG.

Abbreviations: ARCNN: Artifact reduction convolutional neural network; Bpp: Bits per pixel; BRR: Bounded residual reconstruction; JPEG: Joint Photographic Experts Group; mAP: Mean average precision; pp: Percentage points; SPRR: Structure-prior residual refinement.

while retaining the existing JPEG sender, syntax, and payload. The global detection improvements in Table 2 are therefore best interpreted as recovery of detection-relevant information at a fixed transmitted JPEG payload. This configuration is relevant when UAV transmitters rely on fixed camera pipelines, legacy encoders, established protocols, or compute-constrained front-end hardware.

This design rationale also explains why the method uses a small bounded residual network and an input-level Sobel prior rather than a larger learned codec or detector-guided

optimization loop. The residual bound limits the extent of receiver-side modification, the Sobel prior supplies a simple structure cue without changing the transmitted file, and detector freezing separates refinement training from detector adaptation. The resulting design is deliberately constrained: it seeks a practical receiver-side intervention under compatibility and payload constraints, not a full replacement of the imaging and communication stack.

This system configuration provides a distinct design trade-off from learned codecs and sender-side

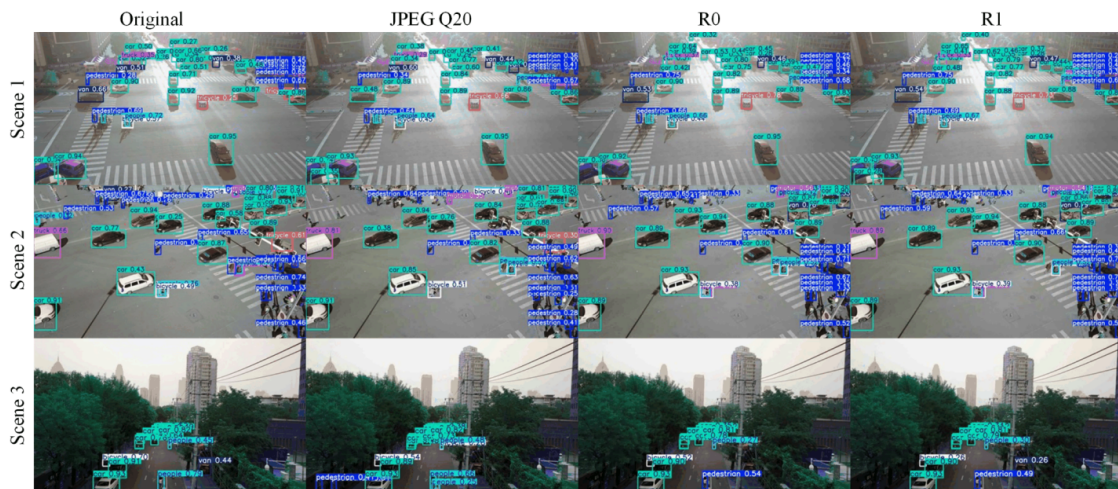


Figure 6. Representative detection-preservation examples at JPEG Q20. Columns show the reference image, JPEG-decoded image, BRR (R0), and SPRR (R1) using the seed 42 checkpoints. The panels are illustrative only and do not substitute for the quantitative results. Images are reproduced from the VisDrone2019-DET validation split for research illustration.

Abbreviations: BRR: Bounded residual reconstruction; JPEG: Joint Photographic Experts Group; SPRR: Structure-prior residual refinement.

preprocessing methods.¹²⁻⁴⁷ Those approaches may be preferable when the entire coding system can be redesigned. JCRR instead targets the case in which JPEG is already selected as the transport format and the receiver is the available intervention point. The contribution is consequently a compatibility-constrained reconfiguration of an existing imaging-and-communication workflow, rather than a replacement of the workflow itself.

4.2. Interpretation of BRR, SPRR, and ARCNN

The three-seed experiment refines the interpretation of the two JCRR variants. BRR and SPRR consistently increased mAP50 relative to JPEG, and the seed-to-seed standard deviations were small. Their global mAP50 values were closely matched: at Q20, the difference between the two means was only 0.0012 percentage points. Thus, the data do not support a claim that SPRR globally dominates BRR.

The more specific evidence concerns strict-small matched-target recall. SPRR had a small positive mean advantage over BRR at every quality and improved strict-small recall relative to JPEG in all three seed-specific bootstrap analyses. This pattern is compatible with the intended role of the normalized edge prior: it may preserve local structures useful for compact aerial-object matching. Nevertheless, the effect is modest and limited to the evaluated architecture, detector, dataset, and Q20 training protocol.

ARCNN provides an important external control. Its three-seed results show that a conventional post-decoding

restoration baseline can increase global mAP50 over JPEG under matched data. However, it did not match BRR or SPRR in the selected Q20 reference-image fidelity measures and its detection gains were smaller. The result is not an exhaustive comparison against every artifact-removal network. It establishes a transparent, externally sourced receiver-side baseline while maintaining the same images, training split, detector, and evaluation settings.

The added YOLO26n robustness check further clarifies the detector dependence of the findings. When the downstream detector was changed from YOLO26s to YOLO26n without retraining the receiver-side refinement models, ARCNN, BRR, and SPRR still improved mAP50 over JPEG at all three JPEG qualities. This supports the view that the receiver-side gains are not tied to a single YOLO26s checkpoint. At the same time, the experiment remains limited to the same dataset family and YOLO-style detectors; it should be interpreted as additional detector-level robustness evidence rather than universal detector generalization.

4.3. WebP and the scope of codec comparisons

WebP remained competitive at nearby operating points, especially in the low-rate condition. This result is important because it prevents an overly broad interpretation of JCRR. The selected WebP files used lower Bpp than the corresponding JPEG points, so the comparison cannot establish a strict codec ranking. When a transmitter-side format can be freely selected, WebP may be a strong

conventional alternative. JCRR addresses a different constraint: preserving JPEG at the sender while improving receiver-side machine analysis.

Likewise, JCRR is complementary to coding-for-machines and learned compression research. Such methods can optimize learned representations, entropy models, bit allocation, or task-aware objectives. JCRR does not optimize JPEG bit allocation, create a learned bitstream, or use detector gradients. Its demonstrated value is confined to a conventional JPEG link in which compatibility takes priority over redesign.

4.4. Limitations

This study has several limitations. First, the formal results use one VisDrone validation split. The added YOLO26n experiment partially addresses detector-level robustness, but it does not establish cross-dataset generality or universal cross-detector behavior. Second, the three training seeds quantify a limited source of variability, but they do not replace broader robustness analysis across data splits, training schedules, architectures, or detectors. Third, ARCNN is one external baseline with a specified architecture and holdout-selected training protocol; it is not a survey of all JPEG artifact-removal methods.

Fourth, the strict large-object bin contains only 35 annotations, preventing stable inference about large-object behavior. Fifth, PSNR and SSIM were measured only at Q20 on the reconstruction holdout. They do not replace perceptual studies, subjective assessment, or cross-quality reconstruction analysis. Sixth, forward timing is limited to FP32 receiver-side restoration model forwarding with preloaded GPU tensors and excludes JPEG decoding, data transfer, detector inference, communication, power, and embedded-device behavior; the reported FPS therefore should not be interpreted as end-to-end UAV-system throughput. Finally, this study is image-based and does not evaluate video temporal consistency, packet loss, network variation, or flight-system deployment.

4.5. Future directions

Future work should test additional aerial datasets and more diverse detector families, compare against additional modern post-decoding restoration baselines, and evaluate receiver-side refinement at carefully rate-matched codec operating points. Hardware-aware measurement on embedded receivers, including full decoding-refinement-detection pipelines, and temporally consistent refinement for UAV video are also needed before claims about flight-system deployment or end-to-end real-time UAV operation can be made. In addition, data-permission procedures for qualitative examples should be considered

whenever benchmark imagery is reproduced in open-access publications.

5. Conclusion

This study evaluated whether a conventional JPEG-based UAV image link can preserve downstream object detection more effectively through a receiver-side refinement stage. JCRR applies BRR or SPRR after standard JPEG decoding and before a fixed downstream detector, without changing the transmitter, JPEG syntax, or transmitted JPEG Bpp.

Across three training seeds and Q15, Q20, and Q25 JPEG operating points, both BRR and SPRR improved mAP50 over JPEG at the same transmitted payload and exceeded the implemented ARCNN baseline in global detection recovery. BRR and SPRR had closely matched global accuracy, whereas SPRR showed a small, consistent advantage in strict-small matched-target recall. Both variants improved the selected Q20 reconstruction metrics relative to JPEG while retaining bounded FP32 forward-only receiver-side costs of approximately 62 FPS on the tested RTX 4090 D platform. The additional YOLO26n robustness check showed that the positive mAP50 gains over JPEG were not restricted to the primary YOLO26s checkpoint, although cross-dataset and broader detector-family validation remain future work.

WebP remained competitive at nearby operating points; therefore, JCRR is not presented as a universal codec replacement or a bitrate-reduction method. Its contribution is a compatibility-oriented receiver-side enhancement path for UAV systems that must retain an existing JPEG sender and payload; full real-time UAV deployment requires separate end-to-end system evaluation.

Acknowledgments

None.

Funding

This work was supported by the Program for Young Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region of China (NJYT23060) and the Hohhot Basic Research and Applied Basic Research Program (Grant No. 2024-rule-base-29).

Conflict of interest

The authors declare that they have no competing interests.

Author contributions

Conceptualization: Hong Zhang, Le Wang

Data curation: Le Wang

Formal analysis: Le Wang

Funding acquisition: Hong Zhang

Investigation: Le Wang

Methodology: Le Wang, Hong Zhang.

Software: Le Wang

Supervision: Hong Zhang

Validation: Le Wang

Visualization: Le Wang

Writing—original draft: Le Wang

Writing—review & editing: Hong Zhang, Le Wang

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data

The VisDrone2019-DET dataset is available from its original provider under the stated terms of use. The supplementary archive accompanying this article includes training and evaluation scripts, deterministic split information, JPEG/WebP encoding settings, aggregated metrics, reproduction metadata, and checkpoint identifiers. No VisDrone image files, crops, JPEG/WebP derivatives, restored image variants, or detection visualizations are redistributed.

Further disclosure

During preparation of the manuscript, the authors used generative AI-assisted tools for language refinement and editorial organization under human supervision. The authors reviewed and verified all text, references, experimental outputs, figures, and interpretations and take full responsibility for the final manuscript. Generative AI tools are not listed as authors or cited as authors.

References

1. Jia C, Ye F, Sun H, Ma S, Gao W. Learning to compress unmanned aerial vehicle (UAV) captured video: benchmark and analysis. In: Proceedings of the 2023 Data Compression Conference (DCC). Snowbird, UT, USA; 2023. doi: 10.1109/DCC55655.2023.00060
2. Varga LA, Koch S, Zell A. Comprehensive analysis of the object detection pipeline on UAVs. *Remote Sens.* 2022;14(21):5508. doi: 10.3390/rs14215508
3. Zhu P, Wen L, Du D, Bian X, Hu Q, Ling H. VisDrone-DET2018: the Vision Meets Drones Object Detection in Image Challenge. In: Leal-Taixe L, Roth S, eds. Computer Vision - ECCV 2018 Workshops. Lecture Notes in Computer Science. Vol 11131. Cham: Springer; 2019:205-220. doi: 10.1007/978-3-030-11021-5_27
4. Ding J, Xue N, Xia G-S, Bai X, Yang W, Yang MY, et al. Object detection in aerial images: a large-scale benchmark and challenges. *IEEE Trans Pattern Anal Mach Intell.* 2022;44(11):7778-7796. doi: 10.1109/TPAMI.2021.3117983
5. Varga LA, Kiefer B, Messmer M, Zell A. SeaDronesSee: a maritime benchmark for detecting humans in open water. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. Waikoloa, HI, USA; 2022:2260-2270. doi: 10.1109/WACV51458.2022.00374
6. Wu J, Wu C, Lin Y, et al. Semantic segmentation-based semantic communication system for image transmission. *Digit Commun Netw.* 2024;10:519-527. doi: 10.1016/j.dcan.2023.02.006
7. Wallace GK. The JPEG still picture compression standard. *Commun ACM.* 1991;34(4):30-44. doi: 10.1145/103085.103089
8. Jocher G, Qiu J, Liu M, Lyu S, Akyon FC, Kalfaoglu ME. Ultralytics YOLO26: Unified Real-Time End-to-End Vision Models. *arXiv*. Preprint posted online 2026. doi: 10.48550/arXiv.2606.03748
9. Buckler M, Jayasuriya S, Sampson A. Reconfiguring the imaging pipeline for computer vision. In: Proceedings of the IEEE International Conference on Computer Vision. Venice, Italy; 2017:975-984. doi: 10.1109/ICCV.2017.111
10. Blasinski H, Farrell JE, Lian T, Liu Z, Wandell BA. Optimizing image acquisition systems for autonomous driving. *Electron Imaging.* 2018;2018(15):161-1-161-7. doi: 10.2352/ISSN.2470-1173.2018.05.PMII-161
11. Liu Z, Lian T, Farrell JE, Wandell BA. Soft prototyping camera designs for car detection based on a convolutional neural network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul, Korea; 2019:2383-2392. doi: 10.1109/ICCVW.2019.00292
12. Lu G, Ge X, Zhong T, Hu Q, Geng J. Preprocessing enhanced image compression for machine vision. *IEEE Trans Circuits Syst Video Technol.* 2024;34(12):13556-13568. doi: 10.1109/tcsvt.2024.3441049
13. Guleryuz OG, Chou PA, Hoppe H, et al. Sandwiched image compression: wrapping neural networks around a standard codec. In: 2021 IEEE International Conference on Image Processing. Anchorage, AK, USA; 2021:3757-3761. doi: 10.1109/ICIP42928.2021.9506256
14. Son H, Kim T, Lee H, Lee S. Enhanced standard compatible

- image compression framework based on auxiliary codec networks. *IEEE Trans Image Process.* 2021;31:664-677.
doi: 10.1109/TIP.2021.3134473
15. Talebi H, Kelly D, Luo X, *et al.* Better compression with deep pre-editing. *IEEE Trans Image Process.* 2021;30:6673-6685.
doi: 10.1109/TIP.2021.3096085
 16. Chadha A, Andreopoulos Y. Deep perceptual preprocessing for video coding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville, TN, USA; 2021:14852-14861.
doi: 10.1109/CVPR46437.2021.01461
 17. Xiang G, Jia H, Liu J, Cai B, Li Y, Xie X. Adaptive perceptual preprocessing for video coding. In: 2016 IEEE International Symposium on Circuits and Systems. Montreal, QC, Canada; 2016:2535-2538.
doi: 10.1109/ISCAS.2016.7539109
 18. Vidal E, Sturm N, Guillemot C, Corlay P, Coudoux F-X. New adaptive filters as perceptual preprocessing for rate-quality performance optimization of video coding. *Signal Process Image Commun.* 2017;52:124-137.
doi: 10.1016/j.image.2016.12.003
 19. Campos J, Meierhans S, Djelouah A, Schroers C. Content Adaptive Optimization for Neural Image Compression. *arXiv.* Preprint posted online 2019.
doi: 10.48550/arXiv.1906.01223
 20. Tsubota K, Akutsu H, Aizawa K. Universal deep image compression via content-adaptive optimization with adapters. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. Waikoloa, HI, USA; 2023:2529-2538.
doi: 10.1109/WACV56688.2023.00256
 21. Shen S, Yue H, Yang J. Dec-Adapter: exploring efficient decoder-side adapter for bridging screen content and natural image compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Paris, France; 2023:12887-12896.
doi: 10.1109/ICCV51070.2023.01184
 22. Ballé J, Laparra V, Simoncelli EP. End-to-end Optimized Image Compression. *arXiv.* Preprint posted online 2016.
doi: 10.48550/arXiv.1611.01704
 23. Ballé J, Minnen D, Singh S, Hwang SJ, Johnston N. Variational image compression with a scale hyperprior. *arXiv.* Preprint posted online 2018.
doi: 10.48550/arXiv.1802.01436
 24. Minnen D, Ballé J, Toderici G. Joint Autoregressive and Hierarchical Priors for Learned Image Compression. *arXiv.* Preprint posted online 2018.
doi: 10.48550/arXiv.1809.02736
 25. Cheng Z, Sun H, Takeuchi M, Katto J. Learned image compression with discretized Gaussian mixture likelihoods and attention modules. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, WA, USA; 2020:7939-7948.
doi: 10.1109/CVPR42600.2020.00796
 26. Lee J, Cho S, Beack SK. Context-adaptive Entropy Model for End-to-end Optimized Image Compression. *arXiv.* Preprint posted online 2018.
doi: 10.48550/arXiv.1809.10452
 27. Zou R, Song C, Zhang Z. The devil is in the details: window-based attention for image compression. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* New Orleans, LA, USA; 2022:17492-17501.
doi: 10.1109/CVPR52688.2022.01697
 28. Bjontegaard G. Calculation of average PSNR differences between RD curves. *VCEG-M33.* 2001. <https://api.semanticscholar.org/CorpusID:61598325>
 29. Li H, Li S, Ding S, *et al.* Image compression for machine and human vision with spatial-frequency adaptation. In: *Computer Vision – ECCV 2024 (Lecture Notes in Computer Science).* Switzerland: Springer Nature; 2024:382-399.
doi: 10.1007/978-3-031-72983-6_22
 30. Hu Y, Yang S, Yang W, Duan L-Y, Liu J. Towards coding for human and machine vision: a scalable image coding approach. In: 2020 IEEE International Conference on Multimedia and Expo. London, UK; 2020:1-6.
doi: 10.1109/ICME46284.2020.9102750
 31. Choi H, Bajić IV. Scalable image coding for humans and machines. *IEEE Trans Image Process.* 2022;31:2739-2754.
doi: 10.1109/TIP.2022.3160602
 32. Liu L, Hu Z, Chen Z, Xu D. ICMH-Net: neural image compression towards both machine vision and human vision. In: Proceedings of the 31st ACM International Conference on Multimedia. New York, NY, USA; 2023:8047-8056.
doi: 10.1145/3581783.3612041
 33. Bai Y, Yang X, Liu X, *et al.* Towards end-to-end image compression and analysis with transformers. In: Proceedings of the AAAI Conference on Artificial Intelligence. Singapore; 2022;36(1):104-112.
doi: 10.1609/aaai.v36i1.19884
 34. Chen Y-H, Weng Y-C, Kao C-H, Chien C, Chiu W-C, Peng W-H. TransTIC: transferring transformer-based image compression from human perception to machine perception. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Paris, France; 2023:23297-23307.
doi: 10.1109/ICCV51070.2023.02129

35. Fischer K, Brand F, Kaup A. Boosting neural image compression for machines using latent space masking. *IEEE Trans Circuits Syst Video Technol.* 2022.
doi: 10.1109/TCSVT.2022.3195322
36. Feng R, Jin X, Guo Z, *et al.* Image coding for machines with omnipotent feature learning. In: *Computer Vision – ECCV*. Cham: Springer; 2022:510-528.
doi: 10.1007/978-3-031-19836-6_29
37. Feng R, Gao Y, Jin X, Feng R, Chen Z. Semantically structured image compression via irregular group-based decoupling. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Paris, France; 2023:17237-17247.
doi: 10.1109/ICCV51070.2023.01581
38. Feng R, Liu J, Jin X, Pan X, Sun H, Chen Z. Prompt-ICM: a unified framework towards image coding for machines with task-driven prompts. *arXiv*. Preprint posted online 2023.
doi: 10.48550/arXiv.2305.02578
39. Liu J, Jin X, Feng R, Chen Z, Zeng W. Composable image coding for machine via task-oriented internal adaptor and external prior. In: *2023 IEEE International Conference on Visual Communications and Image Processing*. Jeju, Republic of Korea; 2023:1-5.
doi: 10.1109/VCIP59821.2023.10402659
40. Liu J, Sun H, Katto J. Improving multiple machine vision tasks in the compressed domain. In: *2022 26th International Conference on Pattern Recognition*. Montreal, QC, Canada; 2022:331-337.
doi: 10.1109/ICPR56361.2022.9956532
41. Li X, Shi J, Chen Z. Task-driven semantic coding via reinforcement learning. *IEEE Trans Image Process.* 2021;30:6307-6320.
doi: 10.1109/TIP.2021.3091909
42. Yan N, Gao C, Liu D, Li H, Li L, Wu F. SSSIC: semantics-to-signal scalable image coding with learned structural representations. *IEEE Trans Image Process.* 2021;30:8939-8954.
doi: 10.1109/TIP.2021.3121131
43. Liu K, Liu D, Li L, Yan N, Li H. Semantics-to-signal scalable image compression with learned revertible representations. *Int J Comput Vis.* 2021;129(9):2605-2621.
doi: 10.1007/s11263-021-01491-7
44. Duan L, Liu J, Yang W, Huang T, Gao W. Video coding for machines: a paradigm of collaborative compression and intelligent analytics. *IEEE Trans Image Process.* 2020;29:8680-8695.
doi: 10.1109/TIP.2020.3016485
45. Yang W, Huang H, Hu Y, Duan L-Y, Liu J. Video coding for machines: compact visual representation compression for intelligent collaborative analytics. *IEEE Trans Pattern Anal Mach Intell.* 2024;46(7):5174-5191.
doi: 10.1109/TPAMI.2024.3367293
46. Wang S, Wang S, Yang W, *et al.* Towards analysis-friendly face representation with scalable feature and texture compression. *IEEE Trans Multimed.* 2021;23:2957-2971.
doi: 10.1109/TMM.2021.3094300
47. Codevilla F, Simard JG, Goroshin R, Pal C. Learned image compression for machine perception. *arXiv*. Preprint posted online 2021.
doi: 10.48550/arXiv.2111.02249
48. Charbonnier P, Blanc-Feraud L, Aubert G, Barlaud M. Two deterministic half-quadratic regularization algorithms for computed imaging. In: *Proceedings of the IEEE International Conference on Image Processing*. Austin, TX, USA; 1994:168-172.
doi: 10.1109/ICIP.1994.413553
49. Loshchilov I, Hutter F. Decoupled Weight Decay Regularization. *arXiv*. Preprint posted online 2017.
doi: 10.48550/arXiv.1711.05101
50. Dong C, Deng Y, Loy CC, Tang X. Compression artifacts reduction by a deep convolutional network. In: *Proceedings of the IEEE International Conference on Computer Vision*. Santiago, Chile; 2015:576-584.
doi: 10.1109/ICCV.2015.73