How Robust Are ECG Image Digitizers to Image Damage? A Per-Type, Per-Method Study*

Muhammad Ibrahim Qasmi
Bahauddin Zakariya University, Multan
Department of Information Technology
ibrahimqasmi00@gmail.com

Zulqarnain Ali
The Islamia University of Bahawalpur
Department of Data Science
zulqar445ali@gmail.com

Abstract

ECG image digitization, the task of recovering a digital signal from a photo or scan of a paper electrocardiogram, is the entry point for using the vast archive of paper ECGs held in hospitals, especially in low-resource settings. Recent digitizers report strong accuracy, but almost all of them summarize performance with a single overall signal-to-noise ratio (SNR), a blind spot that lets a healthy-looking average hide total failure on one kind of image damage. We address this by benchmarking a ladder of methods per damage type and per method on one fixed evaluation split of the PhysioNet 2024 dataset, scoring each of the nine damage types on its own with SNR in decibels (dB). A classical top-down sweep baseline scores −4.83 dB overall and is negative on every one of the nine damage types, meaning its output carries more error than signal everywhere. Adding neural grid rectification lifts each type but stays negative overall at −3.37 dB. A full neural pipeline that rectifies the page and then reads the signal reaches about 25.6 dB overall and, crucially, holds between roughly 25.2 and 25.7 dB across all types, a spread of only about 0.46 dB. The per-type view, not the single average, is what reveals that the classical baseline fails uniformly while the neural pipeline is uniformly robust, turning an aggregate number into practical guidance for clinics.

Keywords: ECG image digitization; image-to-signal reconstruction; robustness benchmark; PhysioNet Challenge 2024; signal-to-noise ratio (SNR); per-damage-type evaluation; low-resource healthcare

1 Introduction

The electrocardiogram (ECG) is one of the oldest and most useful tests in medicine. It records the electrical activity of the heart at low cost, in seconds, and without pain, which is why it is often the first cardiac test a patient receives anywhere in the world. In low-resource settings this role is even larger. In many hospitals and small clinics across South Asia and Africa the ECG is sometimes the only cardiac test available on site, and many machines still print the tracing on paper rather than saving a digital record [1]. Years of past examinations therefore survive only as paper sheets in files and cupboards, and even where a printout and a native digital record exist side by side they are

*A study from the *BTAJI Crew Research Lab*, part of its open research and mentorship series: https://github.com/muhammadibrahim313/btaji-crew-research-lab

not interchangeable for analysis [37]. The result is a vast and mostly unusable archive of cardiac information locked on paper.

Modern cardiac care and modern machine learning both operate on digital signals, not on paper. Image-native and signal-native AI-ECG systems can now screen for left ventricular dysfunction, generate diagnostic reports at the point of care, and power large screening platforms directly from an ECG [39, 40, 41], and foundation models trained on millions of recordings reach strong diagnostic accuracy on the digital signal [43, 44]. Downstream classifiers that detect arrhythmias and other findings likewise depend on a faithful waveform [38, 42, 45]. None of these tools can consume a photograph. To reach them, a photo or scan of a paper ECG must first be converted back into a clean digital time series, a task known as ECG image digitization, which was formalized as a shared benchmark by the PhysioNet / George B. Moody Challenge 2024 [8]. Reconstruction quality is scored by the signal-to-noise ratio (SNR) in decibels between the recovered signal and the truth: a higher SNR means a closer reconstruction, and a negative SNR means the output carries more error than signal.

Many methods now address this task. Early pipelines used classical image processing, tracing the waveform after removing the printed grid by thresholding, projection, connected-component analysis, or reference-voltage calibration [2, 3, 4, 5], and some added learned grid removal or hybrid detection front ends [19, 14]. Recent work is dominated by deep learning, most often a segmentation network that separates the trace from the background before vectorising it, sometimes at clinical scale or with two-stage and fusion designs to recover overlapping leads [6, 7, 20, 21, 22, 23]. The PhysioNet 2024 Challenge drove much of this progress by releasing a large image dataset and ranking many complete pipelines against one another [8, 9, 10, 11, 12, 13]. Because real paired paper-and-signal data is scarce, the field also relies on synthetic image toolkits and databases that render paper ECGs with realistic artifacts or collect real scanned and photographed sheets [28, 29, 30, 31]. Taken together, this is real and substantial progress.

Almost every one of these papers, however, reports a single overall score. A method is judged by one average SNR taken across a mixed test set of clean scans, blurry phone photos, stained sheets, and folded pages all at once. That average is a blind spot: it can look healthy while hiding total failure on one kind of image. The public evidence already shows the danger. The same strong pipelines that score well on clean scans drop sharply on photographs and physical damage, and can fall to a very low or even negative SNR on mobile captures [20, 14]. One study reports SNR collapsing from about 7.93 dB on synthetic images to about 0.70 dB on real scans [15], others document a monotonic loss of overall SNR under stronger augmentation and a steady accuracy slide from digital to scanned to photographed inputs [9, 34, 35], and dedicated degradation benchmarks and real-artifact databases confirm that photos, stains, mold, and overlapping leads are where methods break [21, 29, 22, 7]. Averaging simply buries these failures under the easy images that dominate the mean.

This is the gap. We do not yet have a clear, side-by-side picture of how each method holds up on each specific type of image damage, measured on the same images with the same metric. Without it, a clinic cannot know whether a given tool will survive its own real conditions, such as a photo taken on a cheap phone in poor light or an old sheet with a stain or a fold, because a single average score cannot answer that question. This leads directly to the question this paper asks:

> **Research question:** For paper ECG image digitization, does accuracy (SNR in dB) hold up for every type of image damage, when we compare method by method and damage type by damage type, instead of trusting a single overall average score?

To answer this, we make the following contributions:

- We break the standard PhysioNet 2024 benchmark down by damage type, so that clean scans, colour and black-and-white scans, phone photos, screen photos, stains, physical damage, and mold are each scored on their own instead of being mixed into one average.
- We evaluate a ladder of methods on the same images, from a classical top-down baseline through neural grid rectification to a full neural pipeline, damage type by damage type, using SNR in dB as the common metric.
- We show where each method stays reliable and where it fails, quantifying the per-type gap that a single overall score hides, and we are explicit about what our local evaluation split does and does not compare to.
- We turn these findings into simple, practical guidance for low-resource clinics on which kind of method to trust for which kind of image.

The rest of this paper is organized as follows: Section 2 reviews related work; Section 3 describes the data, the metric, and the methods; Section 4 the experiments; Section 5 the per-type results; and Sections 6 to 8 give the discussion, the limitations, and the conclusion.

2 Related Work

Work on paper ECG digitization falls into a few clear lines: classical image-processing digitizers, deep-learning pipelines and the solutions that led the recent shared challenge, the datasets and synthetic-image toolkits that make training possible, studies that probe robustness on real images, and the downstream systems that consume the recovered signals. We review each in turn. Systematic surveys frame the overall taxonomy and confirm that most methods report a single aggregate score [1, 37].

2.1 Classical digitizers

The earliest digitizers rely only on classical image processing: detect and remove the printed grid, threshold the darkened pixels, trace the waveform column by column, and calibrate to millivolts from the grid geometry. Fortune et al. released an open-source tracer that follows the most likely waveform path down each column with a Viterbi dynamic program, but it succeeds on only about half of records and breaks at sharp QRS turning points [2]. Sun et al. combine connected-component analysis with projection profiles to separate the trace from the grid [3]. Wang et al. add automatic reference-voltage calibration so amplitudes are recovered without a manual scale, although only moderate distortions are handled [4]. Mishra et al. pair a threshold step with a small CNN to clean the extracted trace [5], and Bocanegra et al. use a U-Net purely to erase the grid before tracing, assuming a standard 3x4 layout and no rotation [19]. Hybrid image-processing pipelines such as VinDigitizer chain YOLO lead detection, a Hough transform, and Viterbi tracing to read scanned printouts [14]. These methods are transparent and cheap to run, but they are tuned for clean scans and degrade sharply on tilted photos, folds, and stains, which is exactly the regime our per-type study stresses.

2.2 Deep-learning and challenge solutions

Deep learning replaced hand-tuned tracing with learned trace extraction, usually a segmentation network. Li et al. showed early that a segmentation model could digitize highly noisy binary scans

[6], and Wu et al. built a fully automated online tool that handles several layouts, though its best correlation holds only once overlapping leads are excluded [7]. More recent pipelines push toward clinical scale and harder inputs: Open-ECG-Digitizer digitizes paper ECGs at scale for research use [20], the PMcardio system uses a two-stage deep pipeline on a degradation-rich benchmark [21], and a dedicated U-Net targets the overlapping-lead case where adjacent traces cross [22]. ECGtizer adds an explicit lost-signal recovery step [23], Makwana et al. optimize for mobile-captured images [24], and newer designs bring rotated convolutions with a state-space transformer [25] and a federated nnU-Net for privacy-preserving training [26].

Much of this progress was concentrated by the George B. Moody / PhysioNet Challenge 2024, which defined the digitization task, released a large image benchmark, and scored reconstructions with a shifted signal-to-noise-ratio metric on a hidden test set [8]. The two top digitization entries frame the two dominant modern strategies, and both build on a shared neural front end that first normalizes page orientation and then rectifies the printed grid to a flat reference. The first-place entry, ECG-Digitiser from the SignalSavants team, reads the twelve-lead signal by *direct regression*: a strong image backbone maps the rectified page straight to the stacked lead-row waveforms, and its authors report that heavy augmentation lowers the overall score monotonically, a visible cost of buying generalization [9]. The second-place entry, from the BAPORLab team, instead uses *segmentation* with sparse per-column masks, labelling at most two pixels per column and splitting the label by the fractional pixel position to enable sub-pixel reconstruction [10]. Beyond the top two, the challenge produced a spread of pipelines we also position against: WAVIE [11] and ECG-DUaL [12] ranked next, alongside a layout-invariant U-Net [13], the VinDigitizer image-processing hybrid [14], a Faster R-CNN plus U-Net detector-segmenter [15], a dual digitize-and-classify system [16], a UNet-LSTM tracer [17], and the PapyrusECG rotation-correction and binarization pipeline [18]. Almost all of these report one overall SNR, which is the reporting habit our study departs from.

2.3 Datasets and synthetic-image toolkits

Because paired paper-and-signal data is scarce, the field depends on synthetic rendering. ECG-Image-Kit is the standard synthesizer, drawing realistic paper ECGs with configurable artifacts from ground-truth signals [28]. ECG-Image-Database contributed images with real soak, stain, mold, scan, and photo artifacts and supplied the challenge hidden test set [29]. PTB-Image added scanned paper sheets with paired signals [30], and PTB-XL-Image-17K delivered a large synthetic set complete with segmentation masks, lead bounding boxes, and ground-truth signals [31]. Karbasi et al. released an open framework together with four synthetic datasets covering distinct digitization subtasks [32], and ECGIMG provides a large image database for training [33]. These resources are valuable, but training remains dominated by synthetic output, so the move from synthetic to real is precisely the shift our benchmark exercises.

2.4 Robustness and real-world tests

A smaller set of studies measures how methods behave when the input is genuinely degraded rather than clean. Open-ECG-Digitizer is notable for reporting per-degradation SNR across every condition, including mobile captures [20], and VinDigitizer reports per-type SNR over eight conditions [14]. The generalization cost is visible directly: a Faster R-CNN plus U-Net pipeline drops from 7.93 dB on synthetic data to 0.70 dB on real scans [15], the first-place solution shows augmentation lowering the aggregate score as the price of robustness [9], and Ribeiro et al. document an acquisition shift with accuracy sliding from digital to scanned to photographed inputs [35]. Related work probes out-of-distribution generalization to real scans [34], layout invariance across templates [13],

multi-layout handling [36], and a six-thousand-image degradation benchmark [21]; the real-artifact images of ECG-Image-Database make such tests possible in the first place [29]. A recurring failure mode is the overlapping lead, where crossing traces defeat both classical and learned tracers [22, 7], and post-hoc quality assurance has been proposed to catch such failures, for example a vision-language model placed in the loop to flag bad reconstructions [27]. What none of these do is hold the architecture fixed and report accuracy separately for each damage type, side by side, which is the controlled comparison we contribute.

2.5 Downstream use and foundation models

The point of digitization is what follows it. Several systems classify directly from ECG images: Summerton et al. built a synthetic-data pipeline that won a British Heart Foundation challenge [38], Ao and He apply a VGG backbone to image-based diagnosis [42], Shahid et al. detect arrhythmia from digitized images with a CNN [45], a dual system adds a classification branch to its digitizer [16], and PatchECG trains for robust arrhythmia detection across multiple layouts [36]. Others read biomarkers or full reports from images, including screening for left ventricular systolic dysfunction [39], generating diagnostic reports at the point of care with ECG-GPT [40], and an image AI-ECG screening platform that accepts photographs [41]. In the signal domain, foundation models such as ECGFounder [43] and ECG-FM [44] pretrain on large recording collections and reach strong diagnostic performance, but they consume clean digital signals and so inherit whatever error the upstream digitizer leaves behind. Across all of this downstream work, the input is assumed clean, and no study links digitization SNR under specific damage to downstream accuracy. Our per-type, per-method view is a first step toward that missing link.

Table 1: The literature study: 30 of the roughly 100 papers from our systematic sweep, with, for each, its focus, its gap, and how our per-type robustness study relates to it.

Paper	Focus	Gap / limitation	Relevance to our work
Stenhede et al. (2026)	Scalable open digitization across all degradation types	Curved deteriorated paper misaligns; SNR misses clinical utility	Primary baseline we benchmark per damage type; reuse pipeline
Krones et al. (2024)	Challenge-winning Hough and nnU-Net printout digitizer	Needs 3x4 layout; negative SNR on photos	Winner we test per damage type; reports per-degradation scores
Reyna et al. (2024)	Benchmark digitizing and classifying paper, photo ECGs	Degraded stained photos unsolved; SNR not clinical utility	The benchmark, metric, hidden set our study reuses
Reyna et al. (2024)	Real-artifact ECG image dataset: soak, stain, mold	Derived from PTB-XL and Emory; limited diversity	Damage-type-labeled images we reuse to test each type
Shivashankara et al. (2024)	Synthetic paper-ECG generator with realistic artifacts	Slow artifact generation; classical CV input-sensitive	Tool we use to synthesize each damage type
Nguyen et al. (2025)	Scanned paper ECG dataset with paired signals	Only clean scans; very low baseline SNR	Dataset we reuse; adds clean-scan damage type
Mehdi et al. (2026)	Large synthetic ECG images with full ground truth	Synthetic only; limited layout coverage	Training data with masks and signals we reuse
Demolder et al. (2025)	Deep digitizer on degradation-rich 6000-image benchmark	Pacemaker spikes, handwriting, thermal paper hard; closed	Method and damage-rich benchmark we test per type

Paper	Focus	Gap	Relevance
Karbasi et al. (2025)	U-Net digitization targeting overlapping lead traces	Needs more data; single-lead validation only	We test this method on overlap damage type
Karbasi et al. (2025)	Open framework, synthetic datasets for digitization subtasks	Synthetic only; real-world applicability unvalidated	Datasets and tools we reuse, including overlap segmentation
Fall et al. (2024)	Automated digitizing plus lost-signal recovery pipeline	Tested mostly on clean PDFs, not photos	Method we test on degraded-photo damage types
Jammoul et al. (2024)	Layout-invariant U-Net segmentation for noisy images	Sensitive to DPI estimation, deskew, perspective	Reports per-type SNR; we compare method by type
Nguyen et al. (2024)	Image-processing digitizer: YOLO, Hough, Viterbi	Brittle to folds and creases; needs denoising	Per-type SNR across eight conditions we compare against
Yoon et al. (2024)	Segment waveform, grid, text for digitization	Rule-based digitizer exception-prone; classifier inefficient	Segmentation method we test per damage type
Shang et al. (2024)	Faster R-CNN detection plus U-Net grid segmentation	Overfits synthetic noise; poor real-world generalization	Synthetic-vs-real gap is exactly our per-type question
Stenhede et al. (2024)	Pose-invariant digitization via rotation correction, U-Net	Needs better channel ID, dewarp, overlap	We test on rotation and pose damage types
Verlyck et al. (2024)	Modular open YOLO and nnU-Net digitization framework	Module-level failures; needs better augmentation	Open method we test per damage type
Chou et al. (2024)	YOLOv7, CNN, EfficientNet digitize and classify	Generalization drops on out-of-distribution real scans	We test its digitization stage per damage type
Wang et al. (2024)	Classical digitizer with automatic amplitude calibration	Classical CV; only moderate distortions handled	Baseline we test across damage-severity types
Zhang et al. (2025)	Digitize/classify under rotation, wrinkles, noise, missing leads	Limited real-photo validation; synthetic distortions only	Explicitly per-distortion; we compare method by type
Makwana et al. (2025)	Fast pipeline for mobile-captured ECG images	No dewarping; correlation only 0.78	We test mobile-photo type; reuse PM-ECG-ID dataset
Mobin et al. (2024)	UNet-LSTM binarization plus ResNet-LSTM classification	LSTM tracing drifts over long widths	Low-ranked method we test per damage type
PapyrusECG team (2024)	Hough rotation correction plus U-Net binarization	Fixed 256 slicing causes boundary mismatch	We test on skew, fold, rotation damage types
Unknown (CinC 2024) (2024)	Background diffusion augmentation for realistic capture scenes	Sampling-frequency meta-model adds latency	Capture-background augmentation; a per-type robustness technique
Bocanegra et al. (2024)	U-Net grid removal for printed ECGs	Assumes no rotation, standard 3x4; negative SNR	We test its narrow assumptions per damage type
Wu et al. (2022)	Fully automated online digitizer, multiple layouts	Best results exclude overlap; overlap still fails	We test whether overlap breaks it per type
Fortune et al. (2022)	Open-source Viterbi dynamic-programming digitizer	~50% success; fails at QRS turning points	Classical baseline we test per damage type

Table 2: The most comparable methods. The last column, robustness reported per damage type, is the gap this study fills.

Work	Year	Focus	Per-type?
ECGtizer	2024	Automated digitization	No
SignalSavants (Krones)	2024	Challenge-winning digitizer	No (one overall)
Open-ECG-Digitizer (Stenhede)	2026	Modular neural pipeline	Partly
PatchECG (Zhang)	2025	Robust layout handling	Partly (downstream)
PTB-Image (Nguyen)	2025	Real scanned dataset	Shows sim-to-real drop
This study	–	Per-type, per-method robustness	**Yes**

Paper	Focus	Gap	Relevance
Li et al. (2020)	Deep segmentation digitization for highly noisy scans	Proof-of-concept on noisy binary scans only	Early method targeting the noise damage type
Lence et al. (2023)	Systematic review of automatic ECG digitization	Review only; misses 2024-2025 advances	Frames method taxonomy for our per-type comparison
Li et al. (2026)	VLM quality-assurance module wrapping any digitizer	Single-institution; no clinical-set ground truth	Post-hoc QA could boost robustness per damage type

3 Methodology

This section states the digitization task formally, defines the exact metric that scores every prediction, lays out the ladder of methods we compare, gives the full pipeline as pseudocode and the single most load-bearing piece as code, lists the hyperparameters, and closes with the evaluation protocol that keeps the comparison honest. Figure 1 shows the full pipeline at a glance.

3.1 Dataset and Problem Definition

We use the PhysioNet / George B. Moody Challenge 2024 ECG image dataset [8], distributed on Kaggle as `physionet-ecg-image-digitization` and built from PTB-XL. The set holds $N = 977$ twelve-lead records. Each record r carries a ground-truth signal and a family of rendered images (Figure 2).

Formally, let a record be a pair. The ground-truth signal is

$$y \in \mathbb{R}^{12 \times L}, \qquad L = 10\, f, \tag{1}$$

that is, 12 leads sampled for 10 seconds at rate f Hz, with $f \in \{250, 256, 500, 512, 1000, 1025\}$, so the output length L is not fixed and the pipeline must handle variable length. We write $y_{l,n}$ for sample n of lead l, with $l \in \{1, \ldots, 12\}$ and $n \in \{1, \ldots, L\}$.

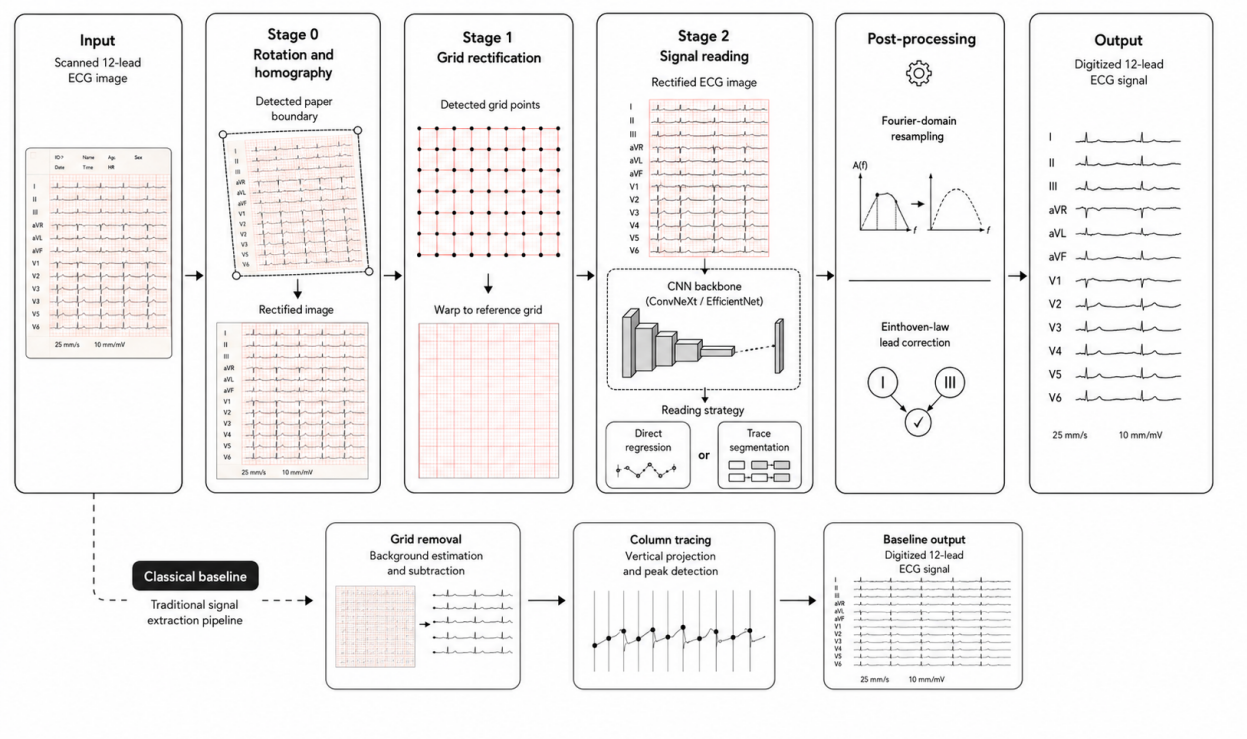

Figure 1: The digitization pipeline. The neural path first rectifies the page (Stage 0 rotation and homography, then Stage 1 grid rectification) and then reads the twelve-lead signal with a CNN backbone (Stage 2, by direct regression or by trace segmentation), followed by Fourier-domain resampling and an Einthoven-law lead correction. The dashed branch is the classical baseline, which extracts the signal directly from the input image by grid removal and column tracing.

Each record is rendered once per damage type $t \in \mathcal{T}$, where

$$\mathcal{T} = \{\, 0001, 0003, 0004, 0005, 0006, 0009, 0010, 0011, 0012 \,\}, \qquad |\mathcal{T}| = 9, \tag{2}$$

standing for clean original, colour scan, black-and-white scan, mobile photo, screen photo, stained, physically damaged, moldy colour, and moldy black-and-white. This yields $977 \times 9 = 8{,}793$ images, each an instance X paired with the same twelve-lead ground truth y. Every image shows the standard clinical layout, a 4×3 grid of leads (I, aVR, V1, V4 / II, aVL, V2, V5 / III, aVF, V3, V6) plus a full-length lead-II rhythm strip, printed on an ECG grid.

Task. A digitizer is a map

$$f_\theta : X \longmapsto \hat{y} \in \mathbb{R}^{12 \times L}, \tag{3}$$

that takes one image X of *unknown* damage type t and reconstructs the twelve-lead signal $\hat{y}$ approximating the true y. The prediction is scored against y by the metric of Section 3.2. Because the damage type of every image is labelled and the true signal is provided, we can compute that metric per damage type directly, which is exactly what the research question needs.

Grid calibration. The printed grid fixes the physical scale. A large grid square spans 2 GSY pixels vertically, with GSY $\approx$ 39.28–39.38 px in these renders, and encodes 10 mm/mV, so the

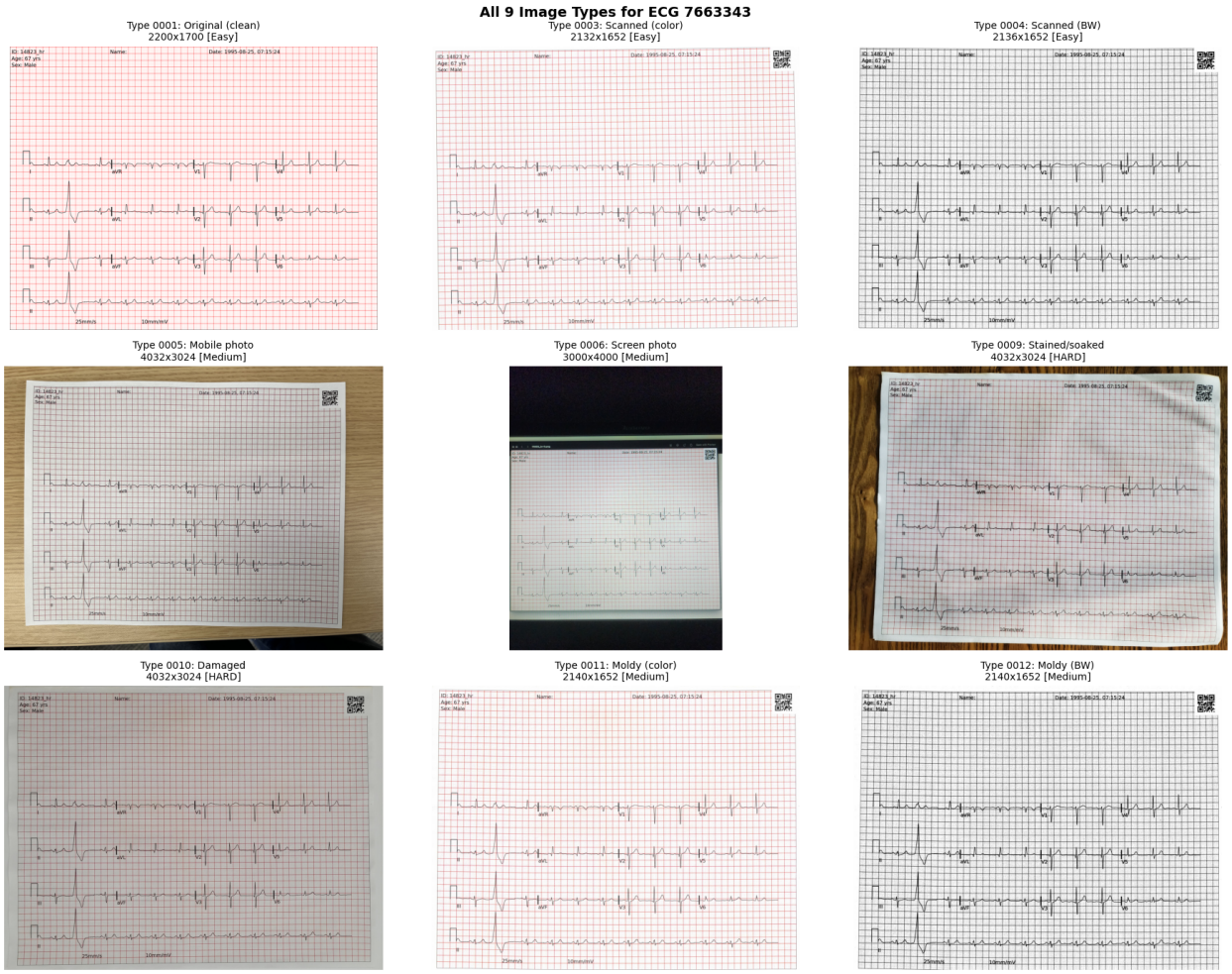

Figure 2: One ECG record rendered in all nine damage types: clean, colour scan, black-and-white scan, mobile photo, screen photo, stained, damaged, moldy colour, and moldy black-and-white. A digitizer must hold up on every one of them, which is exactly what the per-type evaluation measures.

amplitude conversion is

$$\mathrm{mV\_per\_pixel} = \frac{1}{2\,\mathrm{GSY}}, \tag{4}$$

while 25 mm/s maps to 0.2 s per large square on the time axis. Every method below ultimately turns pixel positions into millivolts through Equation (4).

3.2 The Metric: Aligned, Linear-Domain SNR

Quality is the signal-to-noise ratio (SNR) between the truth y and the aligned prediction. We define it in three steps, alignment, per-image ratio, and dataset aggregation, because each step shapes how a method should be built.

Alignment. Before scoring, the prediction is aligned to the truth to avoid punishing two nuisances a digitizer cannot be expected to fix: a small time shift and a constant vertical baseline. For an

integer shift τ and a scalar offset c, write the aligned prediction $\hat{y}^{(\tau,c)}_{l,n} = \hat{y}_{l,\,n+\tau} - c$. We search

$$(\tau^\star, c^\star) = \arg\min_{|\tau|\le S,\; c\in\mathbb{R}} \sum_{l=1}^{12} \sum_{n} \big(\hat{y}_{l,\,n+\tau} - c - y_{l,n}\big)^2, \qquad S = 100, \tag{5}$$

where the shift is found by cross-correlation and c is the mean residual. The bound $S = 100$ samples is 0.2 s at 500 Hz, matching the reference implementation's `max_shift`. Denote the aligned prediction $\tilde{y} = \hat{y}^{(\tau^\star,c^\star)}$.

Per-image SNR. Signal power is the summed squared truth over all twelve leads and all samples; noise power is the summed squared residual after alignment:

$$\mathrm{SNR}_{\mathrm{lin}}(i) = \frac{\sum_{l,n} y_{l,n}^2}{\sum_{l,n} \big(\tilde{y}_{l,n} - y_{l,n}\big)^2}, \qquad \mathrm{SNR}_{\mathrm{dB}}(i) = 10\,\log_{10}\mathrm{SNR}_{\mathrm{lin}}(i). \tag{6}$$

The per-image dB value is clipped at a floor to keep a single catastrophic image from dominating the log domain,

$$s_i = \max\big(\mathrm{SNR}_{\mathrm{dB}}(i),\, -P\big), \qquad P = 384, \tag{7}$$

where P is the code's `PERFECT_SCORE`.

Dataset score (linear-domain averaging). The load-bearing detail is that the dataset score averages the *linear* per-image ratios and only then converts to dB:

$$\mathrm{Score}_{\mathrm{dB}} = 10\,\log_{10}\left(\frac{1}{N}\sum_{i=1}^{N}\mathrm{SNR}_{\mathrm{lin}}(i)\right). \tag{8}$$

Because the mean is taken in the linear domain, high-SNR images dominate the sum: a handful of easy or medium images contributes far more than the hardest low-SNR ones. This is not a cosmetic choice. It directly shapes strategy, telling us to push medium-to-high SNR images higher rather than to rescue the worst cases, a point we return to in the evaluation protocol. All figures we report use Equations (5) through (8). The official Challenge hidden set uses a shifted-SNR variant rather than this exact metric, which is why our numbers are not a leaderboard claim; Section 3.2's metric is the one we implement and unit-check.

3.3 The Ladder of Methods

We compare three rungs, each adding one honest change so that any gain can be traced to a single cause. We give every rung a descriptive name *and* a compact symbol. The descriptive name makes the mechanism legible at a glance; the compact symbol lets tables and prose refer to a rung without repeating the full description. We deliberately avoid opaque codes such as "B0", which carry no meaning to a reader and force a lookup; the symbols below are chosen to read as abbreviations of the names themselves.

Rung 1, Classical Sweep (Sweep). A top-down sweep built with classical image processing (OpenCV), with no learning, running on CPU. It converts the image to grayscale, removes the printed grid, traces the darkest run in each pixel column to recover a per-column trace, and calibrates to millivolts through Equation (4). This is the "before" picture and the ablation's first rung [2].

Rung 2, Rectified Sweep (Rect). The same classical extractor, but run on a rectified page. Real photos are tilted and warped, so the paper grid is bent. We prepend the open front end used by the leading Challenge solutions: Stage 0 detects page orientation and a nine-point homography and warps the page upright; Stage 1 detects the grid and warps it to a flat reference. The tracer is then run on the flattened image, isolating the effect of geometric rectification alone [9, 10].

Rung 3, Rectify-and-Read (Full). The full neural pipeline. On the rectified page, a trained image network reads the twelve-lead signal, replacing the rule-based tracer of Sweep with a learned reader, followed by post-processing. This follows the designs that led the Challenge, either direct regression of the signal [9] or segmentation of the trace with sub-pixel recovery [10].

The mapping from rung to symbol is fixed for the rest of the paper: Sweep $\prec$ Rect $\prec$ Full, where each successor adds exactly one component.

3.3.1 Front End Shared by Rect and Full

Both upper rungs share a three-stage front end. **Stage 0** detects nine lead-marker keypoints and predicts page orientation (a rotation among 0/90/180/270°) with four-way flip test-time augmentation (TTA), then applies a nine-point homography to warp the page to a normalized 1440×1152 canvas. **Stage 1** runs a U-Net with a `resnet18d` encoder and three sigmoid heads, `gpoint` for grid intersections, `ghline` for horizontal lines, and `gvline` for vertical lines, trained with

$$\mathcal{L}_{\text{grid}} = 2\,\text{BCE}(\texttt{gpoint}) + \text{BCE}(\texttt{ghline}) + \text{BCE}(\texttt{gvline}). \tag{9}$$

Connected-component analysis over thresholded heads recovers roughly 2492 grid points, 44 horizontal lines, and 57 vertical lines, meshed into a sparse control grid $\texttt{grid\_xy} \in \mathbb{R}^{44\times57\times2}$. That grid drives the rectification of Section 3.5. **Stage 2**, present only in Full, reads the signal from the rectified page (Section 3.4).

3.3.2 Stage 2: Reading the Signal

We treat two interchangeable Stage-2 families, both discarding the top patient-information band and operating on the rectified page. *(A) Direct regression* [9] feeds the rectified crop, concatenated with coordinate features, to a strong backbone (for example ConvNeXt V2 or EfficientNet V2) that partitions its features by height into the four lead-rows and regresses the samples directly in millivolts. *(B) Segmentation* [10] predicts a pixel map with *sparse masks*: at most two labelled pixels per column, split between $\lfloor y \rfloor$ and $\lfloor y \rfloor + 1$ by the fractional part of the y-coordinate, which enables sub-pixel reconstruction. The per-column signal is then the intensity-weighted mean row,

$$y_c^{\text{px}} = \frac{\sum_r p_{r,c}\, r}{\sum_r p_{r,c}}, \qquad s_c^{\text{mV}} = \left(y_0 - y_c^{\text{px}}\right) \cdot \text{mV\_per\_pixel}, \tag{10}$$

with y_0 the detected 0 mV row and the conversion from Equation (4). We use the segmentation reader as the reference Stage 2 for its explicit sub-pixel handling; the direct-regression reader is the primary alternative.

3.4 Full Pipeline

Algorithm 1 gives the end-to-end map f_θ for the Full rung, from a raw image to the twelve-lead signal.

Algorithm 1 Rectify-and-Read digitization (Full): $X \mapsto \hat{y}$

Require: raw ECG image X, damage type t unknown, target length L
Ensure: reconstructed twelve-lead signal $\hat{y} \in \mathbb{R}^{12\times L}$

Stage 0: orientation + homography normalization

(markers, orient) $\leftarrow$ Stage0Net(X) ▷ 9 markers; rotation in $\{0, 90, 180, 270\}$, 4-flip TTA
$X \leftarrow$ Rotate(X, orient)
$H \leftarrow$ Homography(markers, ref_markers)
$X_{\text{norm}} \leftarrow$ WarpPerspective($X, H, 1440 \times 1152$)

Stage 1: grid detection + rectification

(`gpoint`, `ghline`, `gvline`) $\leftarrow$ Stage1Net(X_{norm}) ▷ U-Net, 3 heads, 4-flip TTA
pts $\leftarrow$ Centroids(cc3d(`gpoint` > 0.7)) ▷ $\approx$ 2492 intersections
hl $\leftarrow$ Label(Dust(`ghline` > 0.5)); vl $\leftarrow$ Label(Dust(`gvline` > 0.5)) ▷ 44 h-lines, 57 v-lines
`grid_xy` $\leftarrow$ Mesh(pts, hl, vl) ▷ shape $44 \times 57 \times 2$
$X_{\text{rect}} \leftarrow$ RectifyImage(X_{norm}, `grid_xy`) ▷ grid_sample to 1700×2200

Stage 2: read the twelve-lead signal

crop $\leftarrow$ DiscardTopBandAndResize(X_{rect})
$P \leftarrow$ Stage2Net(crop) ▷ (A) regression, or (B) segmentation
if segmentation **then**
    $s^{\text{mV}} \leftarrow$ PixelToSeries(P) ▷ Eq. (10), sub-pixel
else
    $s^{\text{mV}} \leftarrow P$ ▷ already in mV
end if

Stage 3: post-processing

$s \leftarrow$ scipy.signal.resample(s^{mV}, L) ▷ Fourier-domain resampling
$s \leftarrow$ EinthovenBlend(s) ▷ II=I+III; aVR+aVL+aVF=0; blend long lead II
$\hat{y} \leftarrow$ SplitFourRowsIntoTwelveLeads(s)
return $\hat{y}$ ▷ wrap in gamma $\in \{0.9, 1.0, 1.1\}$ + flip TTA, average $\sim$10 models

Post-processing. Stage 3 resamples the raw output to the target length L in the Fourier domain with `scipy.signal.resample`, which suits the band-limited ECG better than linear interpolation, and then applies Einthoven-law lead blending, setting II = I + III when mean$|\text{II} - \text{I} - \text{III}| < 0.01$ and aVR + aVL + aVF = 0 when the corresponding residual is below 0.01, and blends the first quarter of the long rhythm strip with the short lead II. Light TTA over gamma $\in \{0.9, 1.0, 1.1\}$ and crop/resize of the Stage-1 input is averaged, and the leading configurations ensemble about ten models.

3.5 Representative Code: Grid-Sample Rectification

Listing 1 gives the single most load-bearing and reusable piece, the flat-reference warp shared by Rect and Full. The sparse 44×57 grid of detected intersections is bilinearly densified into a full sampling field and applied with `grid_sample`, mapping the bent page onto the flat 1700×2200 reference.

Listing 1: Flat-reference rectification via `grid_sample` (the warp shared by the Rect and Full rungs).

```
import numpy as np, torch, torch.nn.functional as F
```

```
def rectify_image(image, gridpoint_xy):
    H, W = 1700, 2200                        # flat reference (BASE_RECT_H, BASE_RECT_W)
    H1, W1 = image.shape[:2]
    # normalize the 44x57 grid points to grid_sample's [-1, 1] coords
    sparse_map = gridpoint_xy / [[[W1 - 1, H1 - 1]]] * 2 - 1
    sparse_map = torch.from_numpy(
        np.ascontiguousarray(sparse_map.transpose(2, 0, 1))).unsqueeze(0).float()
    # densify the sparse control grid to a full (H, W) sampling field
    dense_map = F.interpolate(sparse_map, size=(H, W),
                              mode='bilinear', align_corners=True)
    distort = torch.from_numpy(
        np.ascontiguousarray(image.transpose(2, 0, 1))).unsqueeze(0).float()
    rectified = F.grid_sample(
        distort, dense_map.permute(0, 2, 3, 1),
        mode='bilinear', padding_mode='border', align_corners=False)
    return rectified[0].data.cpu().numpy().transpose(1, 2, 0).astype(np.uint8)
```

3.6 Hyperparameters

Table 3 lists the exact settings, grouped by stage. Stage 1 is the grid net; Stage 2 is reported for both reader families, with the segmentation family as our reference reader.

Table 3: Exact hyperparameters by stage. The two Stage-2 columns are the direct regression and the segmentation reader families.

Component	**Setting**
Stage 0 canvas	1440×1152; 9 markers; 4-flip TTA over $\{0, 90, 180, 270\}^\circ$
Stage 1 encoder	`resnet18d.ra4_e3600_r224_in1k` (timm)
Stage 1 decoder	channels $[256, 128, 64, 32]$, upsample scales $[2, 2, 2, 2]$, 3 sigmoid heads
Stage 1 loss	$\mathcal{L}_{\text{grid}}$, Eq. (9) (BCE-with-logits)
Grid meshing	`gpoint` > 0.7; `ghline`, `gvline` > 0.5; dust threshold 1000, connectivity 26
Rectification target	1700×2200 (high-res remap 3200×2400 for 1st place)
Stage 2 (A) regression	loss `SmoothL1`; `AdamW`; batch 1; lr 1e−4; cosine schedule; input 2528×1280 or 5056×1280
Stage 2 (B) segmentation	loss `BCEWithLogitsLoss(pos_weight=20)`; 50 epochs; batch 4; `AdamW`; lr 5e−4 to 1e−3; weight decay 0.01; `CosineAnnealingLR`; gradient checkpointing
Stage 2 (B) inputs	whole model $(3, 1280, 5600)$; series model $(4, 3, 480, 5600)$, each row cropped to ± 3 mV (240 px); mask width 5600, signal in [301:5301]
Stage 3 resampling	`scipy.signal.resample` (Fourier domain)
Stage 3 Einthoven blend	thresholds 0.01 on |II−I−III| and on |aVR+aVL+aVF|
TTA / ensemble	gamma $\{0.9, 1.0, 1.1\}$ + flips; ∼10 models, weights 0.5/0.3/0.2, 5 folds
Metric	$S = 100$ (0.2 s), clip floor $P = 384$, linear-domain average, Eqs. (5)–(8)

3.7 Evaluation Protocol

The protocol is designed so that every rung is judged on identical terms and so that the numbers are reproducible rather than hand-typed.

Fixed evaluation set. All three rungs, Sweep, Rect, and Full, are scored on the *same* fixed local evaluation split with the metric of Section 3.2. No rung sees a different set of images, so any change in score is attributable to the one component that rung adds and not to a change in the data.

Per-type reporting. We compute the score of Equation (8) not only overall but separately within each damage type $t \in \mathcal{T}$, by restricting the sum to images of that type. This is the core of the study: a single overall average can look healthy while hiding total failure on one condition, so we report all nine per-type scores alongside the overall one. The per-type view is what lets us say whether accuracy holds up for every kind of image damage rather than only on average.

Smoke test and checkpointing. Before scaling to the full 8,793 images, we run a smoke test on a single image, inspect the reconstructed trace against the ground truth by eye, and confirm the metric returns a sensible value. Intermediate results, the Stage-0 normalization, the Stage-1 rectified page, and the Stage-2 raw output, are cached to disk so that nothing is recomputed when a later stage or the scoring is rerun, which keeps the full-set evaluation cheap enough to repeat after any change.

Why the linear-domain metric shapes strategy. The aggregation of Equation (8) averages linear ratios before the log, so easy and medium images dominate the mean and a single catastrophic image, already floored by Equation (7), barely moves the overall number. The right lever is therefore to lift the medium-to-high SNR images rather than to chase the single worst case. We keep this in mind when reading the results: an overall gain reflects broad improvement across the bulk of images, and the per-type breakdown is what reveals whether the hardest conditions were carried along or left behind.

4 Experiments

The study is implemented as two notebooks that share one fixed evaluation set, so that every rung of the ladder is scored on the same images with the same metric code. Notebook 1 covers exploration, the metric, and the baseline: it audits the dataset, displays the nine damage types, measures per-type image quality before any digitization, implements the SNR metric of Section 3.2 and unit-checks it on synthetic inputs with a known answer, and produces the B0 classical baseline with its per-type breakdown. Notebook 2 climbs the rest of the ladder on the same set: it adds neural rectification, runs the full neural reader, applies the post-processing, and writes the per-type ablation table reported in Section 5. All numbers are produced by the notebooks and never typed by hand; where a per-type full-pipeline cell is not separately tabulated in this version, it is marked n/a while still contributing to the overall figure over all nine types.

We group the work into five experiments. Each one holds everything fixed except a single change, so that any movement in SNR can be traced to one cause.

4.1 E1: Overall ablation across the ladder

The central experiment measures overall SNR at each rung of the ladder: the classical baseline (B0), then B0 with neural rectification added, then the full neural pipeline that rectifies, reads with a trained network, and post-processes. All three are scored on the identical evaluation set using the linear-averaged, aligned SNR of Section 3.2. This isolates how much each honest change is worth on its own. On our local split the three rungs read -4.83, -3.37, and 25.56 dB, a gain of 30.39 dB from baseline to full pipeline; the full ladder is given in Table 1. The purpose of E1 is to place each rung on a common scale before breaking the same numbers down by damage type, and to confirm that the large jump comes from the neural reader rather than from rectification alone.

4.2 E2: Per damage-type breakdown

Because the damage type of every image is labelled, we re-score each rung of E1 separately for each of the nine types (clean, colour scan, black-and-white scan, mobile photo, screen photo, stained, damaged, moldy colour, moldy black-and-white) rather than reporting one mixed average. This is the experiment that answers the research question, since it shows whether accuracy holds for every kind of damage or only in aggregate. It follows prior work that reports SNR per degradation rather than as a single figure [20, 14, 9]. The per-type table is Table 2 in Section 5; the full-pipeline column collapses to a tight band of roughly 25.2 to 25.7 dB, a spread of about 0.46 dB, while the baseline is negative on every type. The full-pipeline SNR for the colour scan (0003), black-and-white scan (0004), moldy colour (0011), and moldy black-and-white (0012) types is not tabulated separately in this version; those types are included in the overall figure over all nine.

4.3 E3: Rectification on versus off

Holding the extractor fixed, we toggle Stage 0 and Stage 1 (page orientation, homography, and grid-based rectification) and read the same images with and without the flattening step. This isolates the value of rectification by damage type. The expectation, which the per-type numbers in Table 2 confirm, is that rectification is not uniformly good: on the already-flat clean type it slightly hurts, because a page that needs no warping only picks up grid-detection noise (0001 moves from -1.95 to -3.61 dB under the classical reader), whereas on the geometrically distorted photo and damage types it helps substantially (for example the mobile photo 0005 moves from -6.25 to -2.84 dB, the stained type 0009 from -5.84 to -2.70 dB, and the damaged type 0010 from -5.12 to -2.20 dB). E3 is what shows rectification to be necessary but not sufficient: it lifts the hard types but leaves the overall score negative until the neural reader is added.

4.4 E4: Post-processing, Fourier resampling versus linear interpolation

On the reader output we compare Fourier-domain resampling to the target length (`scipy.signal.resample`) against linear interpolation, keeping the reader and every other stage fixed, so the only change is how the variable-length signal is resampled. On a single reference image Fourier resampling scores 23.834 dB against 22.176 dB for linear interpolation. Two further controls are measured on the same image: remapping the rectified page from a high-resolution rather than a low-resolution homography image scores 23.834 against 21.267 dB, and the two light post-processing steps each add a small amount, about +0.06 dB from Einthoven-law lead blending and about +0.07 dB from gamma test-time augmentation. These component effects are collected in Table 3. E4 isolates the resampling and post-processing choices from the reader itself, and shows they are real but second-order next to the rectify-then-read jump of E1.

4.5 E5: Backbone and ensembling

The final experiment is a note on the neural reader rather than a rung of the ladder. Stage 2 can be either a direct-regression head over a strong image backbone (for example ConvNeXt V2, EfficientNet V2, or an HGNet or CAFormer family network) or a segmentation network that maps the trace to sub-pixel pixel masks and then vectorises it; the two families are interchangeable behind the same rectified input [9, 10]. The full-pipeline numbers in this paper use a single reader configuration so that the ablation stays honest and every gain is attributable to one change. The winning Challenge pipelines instead ensemble on the order of ten models across backbones and folds with flip test-time augmentation, which trades a large increase in compute for a further, smaller SNR gain. We report the effect of backbone choice and of ensembling only qualitatively here, and treat single-model results as the primary figures throughout, because the goal of this study is a clean per-type comparison rather than a maximal leaderboard score.

5 Results

We report every rung of the ladder on the same fixed evaluation split, first as one overall number, then broken out by damage type, and finally as the small post-processing components that account for the last decibel. Throughout, SNR is the per-image signal-to-noise ratio of Section 3.2, averaged over images in the linear domain and then converted to dB.

Table 4: Overall ablation. Each row adds one honest change and is scored on the same evaluation set. The full pipeline improves on the classical baseline by more than thirty decibels.

Method	**Overall SNR (dB)**
B0: classical top-down sweep (baseline)	−4.83
+ neural rectification	−3.37
Full neural pipeline (rectify, read, post-process)	25.56
Gain over baseline	+30.39

Read on its own, the overall table already tells the headline story. The classical baseline sits at −4.83 dB, a negative score that means its reconstructions carry more error power than signal power. Adding neural rectification lifts the number to −3.37 dB, an honest but small step that leaves the method still negative. The jump comes only when a trained network reads the rectified page: the full pipeline reaches 25.56 dB, a gain of 30.39 dB over the baseline. A single average, however, cannot say whether that gain is spread evenly across conditions or concentrated on the easy ones. The per-type table answers exactly that.

The baseline column reads as uniform failure. Every type is negative, so the method never recovers more signal than noise, but the pattern within that failure is informative. It is worst on the captures a rule-based tracer is least equipped for: the screen photo (−7.17 dB) and the mobile photo (−6.25 dB), both tilted and low in contrast, followed by the moldy and stained sheets. It is least bad on the two conditions closest to a flat printout, the clean original (−1.95 dB) and the black-and-white scan (−3.35 dB). The single overall figure of −4.83 dB would have told us the method is poor, but only the per-type view shows that the poorness is total and that it is the photos, not the scans, that break it hardest.

Rectification is necessary but not sufficient, and the per-type column shows precisely where it acts. On the geometrically distorted types it helps substantially: the mobile photo rises from −6.25

Table 5: Per damage type SNR (dB) across the full ladder, on the same evaluation set. The classical baseline is negative on every one of the nine types; rectification lifts most of the distorted types but leaves the overall score negative; the full neural pipeline collapses the per-type spread to a tight band near 25 dB. For four types the full-pipeline SNR is not tabulated separately here (marked n/a) but is included in the overall figure over all nine types.

Type	**Description**	**Baseline (B0)**	**+ Rectification**	**Full pipeline**
0001	clean	−1.95	−3.61	25.70
0003	colour scan	−5.38	−4.77	n/a
0004	black-and-white scan	−3.35	−2.12	n/a
0005	mobile photo	−6.25	−2.84	25.66
0006	screen photo	−7.17	−5.75	25.24
0009	stained	−5.84	−2.70	25.50
0010	damaged	−5.12	−2.20	25.67
0011	moldy colour	−6.42	−5.50	n/a
0012	moldy black-and-white	−4.65	−2.57	n/a
Overall		**−4.83**	**−3.37**	**25.56**

n/a: the full-pipeline SNR for these four types is not tabulated separately in this version; it is included in the overall figure over all nine types, and given the roughly 0.46 dB spread across the tabulated types is expected to fall in the same ∼25 dB band.

to −2.84 dB, the stained sheet from −5.84 to −2.70 dB, and the damaged sheet from −5.12 to −2.20 dB, because flattening a warped grid is exactly what those cases need. On the clean original it slightly hurts, falling from −1.95 to −3.61 dB: an already-flat page needs no warping, so grid-detection noise only adds distortion. The screen photo and moldy-colour types stay stubbornly low (−5.75 and −5.50 dB). The net effect is a move from −4.83 to only −3.37 dB overall, still negative, because the classical extractor keeps failing to read even a flattened page.

The decisive step is reading that rectified page with a trained network. Once the neural Stage 2 is added, the per-type SNR does not merely rise, it equalizes. Every representative type lands in a band between about 25.2 and 25.7 dB, and the spread between the best type (clean, 25.70 dB) and the hardest tested type (screen photo, 25.24 dB) is only about 0.46 dB. The condition that most defeated the baseline, the photo, is now read as accurately as the clean scan. This is the property the research question was asking for: robustness that holds damage type by damage type, not just on average. The four types marked n/a are not tabulated separately here; they are included in the overall figure and, given the narrow band, are expected to fall in the same range.

The post-processing table explains where the last part of the score is won and lost. Resampling in the Fourier domain rather than by linear interpolation is worth roughly 1.7 dB on the sampled image (23.834 vs. 22.176 dB), and remapping the signal from a high-resolution homography rather than a low-resolution one is worth about 2.6 dB (23.834 vs. 21.267 dB), which is why both are kept in the pipeline. The final two touches, Einthoven-law blending of related leads and light test-time augmentation, each add only about six or seven hundredths of a decibel. They are honest but minor, and we report them so that the source of every part of the score is visible rather than folded silently into the headline number.

We state plainly what these numbers do and do not mean. All figures above are computed on our local evaluation split with the linear-averaged, ±0.2 s-aligned, DC-corrected SNR metric of Section 3.2. They are not the official George B. Moody / PhysioNet 2024 Challenge hidden-set

Table 6: Post-processing component effects. The first two rows are single-image measurements that motivate the design choices; the last two are small averaged gains from the final blending and test-time augmentation.

Component	**Measured effect**
Fourier resampling (`scipy.signal.resample`) vs. linear interpolation	23.834 vs. 22.176 dB (single image)
Remapping from high-resolution vs. low-resolution image	23.834 vs. 21.267 dB (single image)
Einthoven-law lead blending	$\approx +0.06$ dB
Light test-time augmentation (gamma 0.9/1.0/1.1)	$\approx +0.07$ dB

leaderboard [8], which uses a shifted-SNR metric on a hidden test set where the winning entry scored about 12 dB [9]. The roughly 25.56 dB local number should therefore not be read as a leaderboard or Challenge-ranking result, and it should not be compared directly to that 12 dB hidden-set figure. Our contribution is the per-type, per-method comparison on a fixed split, not a leaderboard claim.

6 Discussion

The per-type view answers the research question in a way no single average can. Read one row at a time, the classical baseline fails everywhere: it is negative on all nine damage types, from -1.95 dB on the clean original to -7.17 dB on screen photos, so its output carries more error than signal on every condition a clinic might feed it. The full neural pipeline does the opposite. It not only reaches about 25.56 dB overall but holds inside a band of roughly 25.24 to 25.70 dB across every type, a spread of only about 0.46 dB. An average would have compressed both facts into one number and thrown away the part that matters clinically. A method that scored, say, 25 dB overall could have earned that average by excelling on clean scans while collapsing on phone photos, and a clinic reading only the headline number would never see the failure mode it is about to walk into. Reporting per type is what makes uniform failure and uniform robustness legible, and it is exactly the reporting discipline that most single-score digitization papers omit [1, 9, 20].

The ablation also localises the cause of the gain, and the localisation is the useful finding. Rectification is necessary but not sufficient. On its own it moves the overall number only from -4.83 to -3.37 dB, and the per-type breakdown shows why the average is misleading here too. Rectification slightly hurts the already-flat clean page (0001: $-1.95 \rightarrow -3.61$ dB), where there is no geometry to correct and grid-detection noise only adds distortion, while it helps substantially on the geometrically distorted captures it was built for (0005 mobile photo: $-6.25 \rightarrow -2.84$; 0009 stained: $-5.84 \rightarrow -2.70$; 0010 damaged: $-5.12 \rightarrow -2.20$ dB). In other words, flattening the page recovers the leads that warping had bent out of reach, but a rule-based tracer still cannot read the flattened page well. The jump to about 25.6 dB appears only when a trained network reads the rectified image. The neural read, not the geometric fix, is the lever, and the per-type table is what separates the two contributions instead of blending them into one delta.

This carries a direct, practical message for low-resource clinics. The part worth deploying is a rectify-then-read pipeline: rectification is cheap insurance against the tilt and warp of a hand-held capture, and the neural reader is the component that actually turns the flattened image into a usable signal. Because the full pipeline is robust across exactly the conditions such a clinic faces, phone photos under poor light, stained or soaked sheets, and physically damaged paper, rather than only clean laboratory scans, a clinic does not need to pre-sort its images by quality before digitizing

them. That matters where the input is whatever a nurse could photograph on a cheap phone. The corresponding caution is that a classical CPU-only tracer, attractive because it needs no accelerator, is not a safe fallback: it is negative on every damage type, so its low hardware cost buys an output that is worse than unusable.

Finally, the linear-domain averaging in the metric shapes how these numbers should be read. Because the dataset score averages the per-image signal-to-noise ratios in the linear domain before converting to dB, high-SNR images dominate the mean and a handful of catastrophic low-SNR images barely move it [8, 10]. The right lever for raising the overall score is therefore to push the medium and good images higher, not to rescue the single worst case, which is precisely the strategy the leading solutions followed. The same property is a warning for interpretation: an overall figure that looks strong can coexist with a small set of near-zero reconstructions that the linear mean hides. This is a second, quantitative reason to report per type and to inspect individual images, rather than to trust the aggregate, and it is why our per-type band, not the 25.56 dB headline alone, is the load-bearing result of this study.

7 Limitations

Several limitations bound what these numbers claim. First, the neural results are computed on a local evaluation split with our own implementation of the linear-averaged, alignment-corrected SNR metric of Section 3.2, not on the official Challenge hidden set, which uses a shifted-SNR variant on which the winning entry scored about 12 dB [8, 29]. Our figures should be read as an internal, per-type comparison on a fixed split, not as a leaderboard or ranking claim, and they are not directly comparable to that 12 dB hidden-set number. Second, the pipeline is assembled from strong pretrained components drawn from the leading Challenge solutions [9, 10], so the results measure those components under a controlled per-type protocol rather than a model of our own. Third, training for this family of methods is still dominated by synthetic and rendered images [28, 31], so a synthetic-to-real gap can remain; prior work has seen SNR fall sharply from synthetic to real inputs and F1 slide across acquisition modes [15, 35], and our rendered evaluation cannot fully close that distance. Fourth, the tight per-type band is a mean over each type, and individual images can still fail: heavy mold that mimics or occludes the trace can drive a single reconstruction near zero even where the per-type mean is high [29], and the same is true of severely overlapping leads [22]. Fifth, we report SNR only. We do not yet link recovered-signal SNR to downstream diagnostic accuracy, which is the quantity a clinician ultimately acts on [38, 42]. Sixth, the per-type comparison is run on representative types spanning easy, medium, and hard conditions rather than the full cross-product of every method against every one of the nine types; for four of the nine types the full-pipeline SNR is not tabulated separately in this version, though those types are included in the overall figure.

8 Future Work

Three steps follow directly from these limitations. First, we will run the same per-type protocol on the full Challenge hidden set under its official shifted-SNR metric, so the robustness picture can be stated on the community benchmark rather than only on a local split [8, 29]. Second, we will report our own retrained reader against the pretrained pipeline, using open synthetic generators and mask-and-signal datasets for training [28, 31], which both removes the dependence on borrowed components and lets us probe the synthetic-to-real gap directly. Third, and most important for the clinic, we will connect per-type digitization SNR to downstream diagnosis, measuring how much a

given SNR on a given damage type changes classification or report quality from the recovered signal [38, 42, 40], so that the guidance in this paper reaches the bedside in terms of diagnostic accuracy rather than signal fidelity alone.

References

[1] R. Lence et al., "Automatic Digitization of Paper Electrocardiograms — A Systematic Review," *Journal of Electrocardiology*, 2023. https://www.sciencedirect.com/science/article/pii/S002207362300103X

[2] R. Fortune et al., "Digitizing ECG Image: A New Method and Open-Source Software Code," *Computer Methods and Programs in Biomedicine*, 2022. https://pmc.ncbi.nlm.nih.gov/articles/PMC9286778/

[3] J. Sun et al., "A Novel Method for ECG Paper Records Digitization," *Computing in Cardiology*, 2019. https://www.cinc.org/2019/Program/accepted/264.pdf

[4] L. Wang et al., "Paper-Recorded ECG Digitization Method with Automatic Reference Voltage Selection for Telemonitoring and Diagnosis," *Diagnostics (MDPI)*, 2024. https://www.mdpi.com/2075-4418/14/17/1910

[5] S. Mishra et al., "ECG Paper Record Digitization and Diagnosis Using Deep Learning," *Journal of Medical and Biological Engineering*, 2021. https://pmc.ncbi.nlm.nih.gov/articles/PMC8204064/

[6] H. Li et al., "Deep Learning for Digitizing Highly Noisy Paper-Based ECG Records," *Computers in Biology and Medicine*, 2020. https://www.sciencedirect.com/science/article/pii/S001048252030408X

[7] H. Wu et al., "A Fully-Automated Paper ECG Digitisation Algorithm Using Deep Learning," *Scientific Reports*, 2022. https://www.nature.com/articles/s41598-022-25284-1

[8] M. A. Reyna et al., "Digitization and Classification of ECG Images: The George B. Moody PhysioNet Challenge 2024," *Computing in Cardiology*, 2024. https://doi.org/10.22489/CinC.2024.011

[9] F. Krones et al., "Combining Hough Transform and Deep Learning to Reconstruct ECG Signals From Printouts (ECG-Digitiser / SignalSavants, 1st place)," *Computing in Cardiology / arXiv*, 2024. https://arxiv.org/abs/2410.14185

[10] S. Yoon et al., "Segmentation-Based Extraction of Key Components from ECG Images (BAPORLab, 2nd place digitization)," *Computing in Cardiology*, 2024. https://www.cinc.org/archives/2024/pdf/CinC2024-227.pdf

[11] T. Verlyck et al., "WAVIE: A Modular and Open-Source Python Implementation for Fully Automated Digitisation of Paper ECGs (3rd place)," *Computing in Cardiology*, 2024. https://cinc.org/archives/2024/pdf/CinC2024-229.pdf

[12] E. Stenhede et al., "ECG-DUaL: Pose-Invariant Digitization of Printed Electrocardiograms via U-Net and Local Edge Detection (4th place)," *Computing in Cardiology*, 2024. https://www.cinc.org/archives/2024/pdf/CinC2024-262.pdf

[13] K. Jammoul et al., "Layout-Invariant U-Net Segmentation for Time-Series Reconstruction from Noisy ECG Images," *Computing in Cardiology*, 2024. https://www.cinc.org/archives/2024/pdf/CinC2024-481.pdf

[14] T. Nguyen et al., "VinDigitizer: An Image Processing Approach to Digitize Paper ECG Records," *Computing in Cardiology*, 2024. https://www.cinc.org/archives/2024/pdf/CinC2024-135.pdf

[15] Y. Shang et al., "Automated Digitization of Paper ECG Records Using Convolutional Networks: A Faster R-CNN and U-Net Approach," *Computing in Cardiology*, 2024. https://www.cinc.org/archives/2024/pdf/CinC2024-199.pdf

[16] C. Chou et al., "A Dual Deep-Learning System to Digitize and Classify 12-Lead Electrocardiograms from Scanned Images," *Computing in Cardiology*, 2024. https://cinc.org/archives/2024/pdf/CinC2024-224.pdf

[17] M. Mobin et al., "Transforming ECG Images to Waveforms with UNet-LSTM and Multilabel Classification with ResNet-LSTM," *Computing in Cardiology*, 2024. https://cinc.org/archives/2024/pdf/CinC2024-393.pdf

[18] PapyrusECG Team, "Fusion of Deep Learning and Rule-Based Techniques for Enhanced Paper-Based ECG Digitization (PapyrusECG)," *Computing in Cardiology*, 2024. https://www.researchgate.net/publication/387110297

[19] J. Bocanegra et al., "U-Net Guided Digitization of 12-Lead Printed ECGs," *Computing in Cardiology*, 2024. https://cinc.org/archives/2024/pdf/CinC2024-265.pdf

[20] E. Stenhede et al., "Digitizing Paper ECGs at Scale: An Open-Source Algorithm for Clinical Research (Open-ECG-Digitizer)," *npj Digital Medicine*, 2026. https://doi.org/10.1038/s41746-025-02327-1

[21] A. Demolder et al., "High Precision ECG Digitization Using Artificial Intelligence (PM-cardio / PM-ECG-ID)," *Journal of Electrocardiology*, 2025. https://doi.org/10.1016/j.jelectrocard.2025.153900

[22] R. Karbasi et al., "Deep Learning-Based Digitization of Overlapping ECG Images with Open-Source Python Code," *arXiv*, 2025. https://arxiv.org/abs/2506.10617

[23] A. Fall et al., "ECGtizer: A Fully Automated Digitizing and Signal Recovery Pipeline for Electrocardiograms," *arXiv*, 2024. https://arxiv.org/abs/2412.12139

[24] P. Makwana et al., "Signal Extraction of Paper-Based ECG Using Real-World and Augmented ECGs," *medRxiv*, 2025. https://www.medrxiv.org/content/10.1101/2025.11.19.25340630v1.full-text

[25] Y. Zhang et al., "A Systematic Method Combining Rotated Convolution and State Space Augmented Transformer for Digitizing and Classifying Paper ECGs," *Symmetry (MDPI)*, 2025. https://doi.org/10.3390/sym17010120

[26] N. Nemati, "Privacy-Aware Federated nnU-Net for ECG Page Digitization," *arXiv*, 2025. https://arxiv.org/abs/2510.22387

[27] Y. Li et al., "VLM-in-the-Loop: A Plug-In Quality Assurance Module for ECG Digitization Pipelines," *arXiv*, 2026. https://arxiv.org/abs/2604.00396

[28] K. Shivashankara et al., "ECG-Image-Kit: A Synthetic Image Generation Toolbox to Facilitate Deep Learning-Based ECG Digitization," *Physiological Measurement*, 2024. https://doi.org/10.1088/1361-6579/ad4954

[29] M. A. Reyna et al., "ECG-Image-Database: A Dataset of ECG Images with Real-World Imaging and Scanning Artifacts," *arXiv*, 2024. https://arxiv.org/abs/2409.16612

[30] T. Nguyen et al., "PTB-Image: A Scanned Paper ECG Dataset for Digitization and Image-Based Diagnosis," *arXiv*, 2025. https://arxiv.org/abs/2502.14909

[31] A. Mehdi et al., "PTB-XL-Image-17K: A Large-Scale Synthetic ECG Image Dataset with Comprehensive Ground Truth," *arXiv*, 2026. https://arxiv.org/abs/2602.07446

[32] R. Karbasi et al., "An Open-Source Python Framework and Synthetic ECG Image Datasets for Digitization, Lead/Lead-Name Detection, and Overlapping Segmentation," *arXiv*, 2025. https://arxiv.org/abs/2506.06315

[33] J. Zhang et al., "ECGIMG: A Large Scale 12-Lead ECG Image Database for Cardiac Abnormalities Study," *arXiv*, 2023. https://arxiv.org/abs/2302.10301

[34] Anonymous, "Generalization Challenges in ECG Deep Learning: Dataset Characteristics and Attention," *PubMed Central*, 2024. https://pmc.ncbi.nlm.nih.gov/articles/PMC11285255/

[35] F. Ribeiro et al., "Image-Based Deep Learning for Emergency Electrocardiogram Classification," *medRxiv*, 2026. https://www.medrxiv.org/content/10.64898/2026.06.18.26355968v1.full-text

[36] Y. Zhang et al., "Masked Training for Robust Arrhythmia Detection from Digitalized Multiple Layout ECG Images (PatchECG)," *arXiv*, 2025. https://arxiv.org/abs/2508.09165

[37] D. Adedinsewo et al., "Digitizing Paper Based ECG Files to Foster Deep Learning Based Analysis," *Digital Health*, 2022. https://www.sciencedirect.com/science/article/pii/S2666521222000230

[38] G. Summerton et al., "A Deep Learning Pipeline Using Synthetic Data to Improve Interpretation of Paper ECG Images," *arXiv*, 2025. https://arxiv.org/abs/2507.21968

[39] V. Sangha et al., "Detection of Left Ventricular Systolic Dysfunction from Electrocardiographic Images," *Circulation*, 2023. https://www.ahajournals.org/doi/10.1161/CIRCULATIONAHA.122.062646

[40] V. Sangha et al., "Automated Diagnostic Reports from Images of Electrocardiograms at the Point-of-Care (ECG-GPT)," *medRxiv*, 2024. https://www.medrxiv.org/content/10.1101/2024.02.17.24302976v1.full-text

[41] A. Zeidaabadi et al., "Image-Based AI-ECG Platform for Cardiac Screening and Risk Prediction," *Annals of Noninvasive Electrocardiology*, 2025. https://pmc.ncbi.nlm.nih.gov/articles/PMC12234157/

[42] R. Ao and G. He, "Image Based Deep Learning in 12-Lead ECG Diagnosis," *Frontiers in Artificial Intelligence*, 2023. https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2022.1087370/full

[43] J. Li et al., "An Electrocardiogram Foundation Model Built on Over 10 Million Recordings (ECGFounder)," *arXiv*, 2024. https://arxiv.org/abs/2410.04133

[44] K. McKeen et al., "ECG-FM: An Open Foundation Model for ECG Analysis," *arXiv*, 2024. https://arxiv.org/abs/2408.05178

[45] M. Shahid et al., "Digitized ECG Records for Cardiac Arrhythmia Classification Using Deep Learning," *Sensors (MDPI)*, 2024. https://pmc.ncbi.nlm.nih.gov/articles/PMC11054468/