Online Resource 2: Reproducibility

Manuscript: Family Balancing and Hierarchical CNNs for Arabic Handwritten Symbols: A Controlled Five-Seed Evaluation

Author: Meshaal Mouawad
Department of Electrical and Computer Engineering, Mississippi State University
Contact: mm4922@msstate.edu
Experimental-record release: `IJDAR-CNN-2026-10-28-R1`

This supplement explains the operational protocol and its evidence locations. Online Resource 3 supplies the frozen code, model states, predictions, logs, source receipts, and commands. Its `README.md` is the detailed entry point.

1 Source reconstruction and labels

MADBase supplies 60,000 original training and 10,000 test digit images. The Arabic Handwritten Characters Dataset supplies 13,440 training and 3,360 test letter images. The original source hashes and download URLs appear in `record/upstream_source_receipts.json`; CSV hashes and names appear in `record/prepared_canonical/source_and_split_manifest.json`.

The importer reads headerless CSV rows, reconstructs digit images as 28×28 arrays and letter images as 32×32 arrays, transposes letters, and retains both source polarities. It applies Pillow bilinear resizing to 64×64, converts to float32, and divides by 255. Source digit labels remain 0–9. Source letter labels 1–28 map to global labels 10–37. No training augmentation is applied.

2 Image separation and model-development partitions

BLAKE2b-128 fingerprints are calculated after geometric and polarity processing. The original test partition is retained. The pipeline removes 23 source-training rows that match 12 unique processed test fingerprints before dividing development data. Removed source indices are recorded. Identical processed images are kept together during class-stratified splitting, using a 15% validation target and random seed 2026. The original

Table 1: Recorded prepared partitions

Partition	Total	Digits	Letters
Training	62,403	50,979	11,424
Validation	11,014	8,998	2,016
Test	13,360	10,000	3,360

partition audit records zero exact processed-image overlap across the three partitions. The delivered label-only extracts preserve class order without duplicating third-party image pixels. Original prepared files do not contain source-row index vectors or writer identities; reconstruction follows the included algorithm and source hashes.

3 Training and selection

The six configurations and five seeds are listed in Online Resource 1. All use four convolutional blocks with 16, 32, 64 and 128 filters, batch normalization, pooling, dropout and a 128-dimensional global-average-pooled representation. Each run uses 20 epochs, batch size 128 and Adam learning rate 0.001. The smooth objective uses $\lambda = 0.5$ and $\tau = 0.2$.

The frozen mixed-family batch sampler uses an epoch-specific random permutation, retains the natural sample mix, and drops the incomplete last batch. It rejects a batch schedule if a full batch lacks either family; it does not oversample letters. All configurations use the same label-based sampler rule. The selected checkpoint maximizes validation worst-family macro-F1; the earliest epoch wins ties. All 30 training records must exist before the controller runs held-out evaluation.

4 Transforms and measurements

Noise is generated with NumPy seed 1234 for each condition, added at intensity standard deviation 0.05 or 0.10, and clipped to $[0, 1]$. Gaussian blur uses SciPy's `gaussian_filter` with spatial sigma 0.8 or 1.3 pixels, reflected boundaries and truncate 4. Contrast is $\text{clip}(0.5 + \alpha(x - 0.5), 0, 1)$ with $\alpha = 0.75$ or 0.50. Each condition is identical across the model comparisons.

Digit and letter macro-F1 are computed from the full 38-class confusion matrix, so predictions crossing the family boundary count as errors. The primary clean endpoint is the lower family score. Paired Student-t intervals describe variability across five matched seeds. H3 compares corrupt-minus-clean changes and weights the six conditions equally. The first-seed image bootstrap in the recorded statistical JSON is a separate diagnostic.

5 Evidence map and executable route

The files `record/runs/<configuration>_seed<seed>/training_record.json` and `test_results.json` contain each model's history, selection and evaluation metadata. Alongside them, `initial.pt`, `best.pt`, and `predictions.npz` preserve the model states and per-image outputs. Source and partition SHA-256 values are held in `record/EXPERIMENT_LOCK.json`. The `code/` directory matches its 16 recorded code hashes; `locked/Reproducibility_Code.zip` preserves the earlier code archive byte-for-byte.

Run the following from the extracted Online Resource 3 directory to verify reported metrics without image data, a GPU, or retraining:

```
python -m pip install -r requirements-verification.txt
python verification/verify_manifest.py -root .
python verification/verify_predictions.py -root .  -out verification_rerun
```

The README also supplies source-acquisition, partition-reconstruction, selected-checkpoint inference, and full training commands. Re-execution uses a new output directory so that the archived record remains unchanged.

6 Environment and verification scope

Training records report PyTorch 2.11.0+cu130 and CUDA execution. The original record does not pin every other package version. Requirements files identify imports needed for execution, not a recovered complete environment. The frozen code and file identities support inspection and saved-evidence verification; another runtime must be checked for numerical differences.

The current release independently recalculates all 210 saved-prediction rows and the primary paired estimates. The prior audit's partition checks and selected-checkpoint replay are retained under `prior_audit/`. Neither verification is a new repeat of all 30 training runs. Writer identity auditing and calibration of corruptions to a specific document collection remain outside the demonstrated experiment.