Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

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

Meshaal Mouawad

mm4922@msstate.edu

Mississippi State University

Research Article

Keywords: Arabic handwriting recognition, convolutional neural networks, hierarchical classification, class imbalance, document analysis, robustness

Posted Date: October 30th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11541042/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

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

Meshaal Mouawad[1*]

[1*]Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, Mississippi, USA.

Corresponding author(s). E-mail(s): mm4922@msstate.edu;

Abstract

A joint Arabic handwriting recognizer must distinguish digits and letters despite unequal representation in its training data. We test whether family-balanced learning and hierarchical classification improve the less accurately recognized family. To examine this question, six convolutional neural network (CNN) configurations used the same four-block feature-extractor architecture and were trained with five matched random seeds on a 38-class task assembled from MADBase digits and the Arabic Handwritten Characters Dataset. Comparisons separated equal-family loss weighting, hierarchical prediction, smooth emphasis on the higher-loss family, and parameter count. Exact-image-separated partitions contained 62,403 training, 11,014 validation, and 13,360 test images. Each selected model was evaluated on clean images and six fixed corruptions, producing 210 evaluation records. Across these configurations, overall clean-image accuracy ranged from approximately 98.2% to 98.4%. The lower of the digit and letter class-averaged F1 scores was 0.9637 for the flat baseline and 0.9658 for its family-balanced counterpart. Their paired difference was +0.0021 F1, with a descriptive five-seed interval of $[-0.0076, 0.0118]$. Hierarchical prediction produced mixed clean-image effects. Smooth worst-family training reduced mean corruption retention by 0.0317 F1 relative to balanced hierarchical training. Additive noise and midpoint-centered contrast changes caused substantially larger performance losses. Taken together, the controlled comparisons identify small, inconsistent gains from family-aware training and output organization, alongside pronounced sensitivity to the specified image transformations. Family-specific evaluation reveals these differences and directs subsequent work toward understanding the observed corruption response.

Keywords: Arabic handwriting recognition, convolutional neural networks, hierarchical classification, class imbalance, document analysis, robustness

Author ORCID: https://orcid.org/0000-0002-1152-8324

1 Introduction

A joint Arabic handwriting recognizer needs to identify letters as reliably as digits when both are

learned from unequal numbers of examples. This study examines whether changing the allocation of training loss or organizing predictions by symbol family improves the weaker family's recognition. The question arises at the isolated-symbol stage of document analysis: a cropped image containing one handwritten symbol is assigned a digit or letter identity. Convolutional neural networks (CNNs) already provide effective learned representations for this task [1, 2]. Our controlled comparison asks how that recognition quality is distributed across the two kinds of symbol.

The difference between the number of classes and the number of examples makes the problem concrete. A *class* is one symbol identity; a *family* is the group of ten digits or the group of 28 letters. The combined source training pool used here contains 60,000 digit images and 13,440 letter images, so digits contribute 81.70% of its examples. An ordinary training objective averages the error over those images and consequently gives the digit family greater aggregate weight. Overall accuracy is also weighted by the evaluation set's composition. Reporting the digit and letter outcomes separately reveals a performance disparity that a single pooled score can obscure.

Prior Arabic recognition studies have improved feature learning through network depth, residual connections, and ensembles, and have investigated both separate datasets and combined sources [2–4]. Complementary machine-learning research changes the contribution of differently represented classes or groups to training [5, 6], or organizes prediction into broad categories and their constituent classes [7]. These approaches motivate a specific unresolved comparison: with the images, feature-extractor architecture, training budget, and model-selection rule held constant, how much does each intervention change recognition of the weaker Arabic symbol family?

We investigate three prespecified hypotheses. **H1** predicts that giving equal total loss weight to digits and letters improves the lower of their class-averaged F1 scores. **H2** predicts that combining digit/letter probabilities with probabilities within each family improves recognition relative to a flat 38-class output, including a parameter-matched comparison. **H3** predicts that further emphasizing the currently higher-loss family improves retention of recognition quality under fixed image corruptions. The motivating hypothesis is that giving more weight to the higher-loss family strengthens its less well learned symbol distinctions, which may remain useful after image degradation. The first two questions concern clean recognition; the third asks whether attention to family difficulty during training also affects sensitivity to changes in the input image.

Six CNN configurations, each trained with five matched random seeds, provide the corresponding comparisons. All selected models were tested

on the same 13,360 held-out images and on six deterministic noise, blur, and contrast conditions. The clean-image differences were small and varied across seeds; all five paired clean-image intervals crossed zero. Corruption-induced losses were much larger than those clean-image differences. The study therefore contributes controlled evidence about when plausible family-aware changes yield little consistent benefit, together with a reproducible account of their behavior under specified perturbations. The following review develops the mechanisms behind these comparisons before the methods and evaluation criteria are introduced.

2 Related Work

The review connects Arabic visual recognition with two design choices used here: how training losses are weighted and how class probabilities are constructed. It then examines fixed perturbation tests and repeated training as ways to assess those choices. The progression establishes the capabilities inherited by the present experiment and the comparison it carries forward.

2.1 Learning visual distinctions in Arabic handwriting

For an isolated-symbol recognizer, the input is an image containing one symbol and the output is its identity. A convolutional neural network learns image features through filters whose responses are combined across layers. El-Sawy, Loey and El-Bakry [1] investigated this approach for the Arabic Handwritten Characters Dataset (AHCD), comprising 16,800 handwritten characters. Their CNN achieved 94.9% test accuracy. This established a directly relevant starting point: learned image features can distinguish isolated Arabic letters without requiring a separately specified feature extractor.

Subsequent work investigated how network depth and training procedures affect recognition. Mudhsh and Almodfer [3] used a very deep, VGG-style network with dropout and data augmentation for Arabic alphanumeric recognition. They evaluated the design separately on the ADBase digit dataset and the Handwritten Arabic Characters Database (HACDB), whose 66 shape classes differ from AHCD's 28 letter identities. Their study connects deeper visual representations with measures intended to control overfitting. Its experiments also show why the label inventory must accompany an accuracy figure: distinguishing character shapes and distinguishing letter identities define different classification tasks.

Palatnik de Sousa [2] examined a complementary route: combining predictions from multiple CNNs. The training procedure used adaptive gradient descent in early epochs and stochastic gradient descent later, and the study examined Monte Carlo and K-fold validation strategies. The reported test accuracies were 99.47% on MADBase digits

and 98.42% on AHCD characters. The algorithm was trained and evaluated separately on the two datasets. Thus, the study demonstrates the usefulness of an ensemble training procedure across both families, while leaving a different experimental question for a joint recognizer: how should one shared model allocate its learning effort when both families enter the same classification problem?

These studies establish feature learning, network depth and prediction aggregation as relevant design choices. They also identify the appropriate starting point for the present work: a CNN with an explicit label space and evaluation protocol. The question carried forward concerns the distribution of performance within that label space, rather than whether CNNs can recognize Arabic handwriting.

2.2 Combining datasets and interpreting recognition scores

Combining Arabic digit and letter sources has an identifiable precedent. Al-Taani and Ahmad [4] evaluated residual neural networks on MADBase, AIA9K and AHCD and also constructed a combined dataset from those sources. They reported 98.9% validation accuracy for the constructed dataset. This work brings multi-source recognition closer to the present problem than separate digit and letter evaluations. Its validation result and Palatnik de Sousa's dataset-specific test results answer different evaluation questions; their percentages do not form a common performance ranking.

Alghyaline [8] examined another source of performance variation by adapting twelve CNN architectures to small Arabic character images. The study evaluated architectures including VGG-, ResNet- and Inception-based designs on Hijja, AHCD and AIA9K, and examined data augmentation. Its comparison shows that input size, architecture adaptation and training data treatment are part of the recognition method. This motivates holding those choices constant when the aim is to isolate the effect of a loss function or output head.

Dataset composition also changes what the model must learn. Altwaijry and Al-Turaiki [9] introduced Hijja, containing 47,434 character images from 591 children aged 7–12, and trained their CNN separately on Hijja and AHCD. They reported 88% and 97% accuracy, respectively. The experiments concern different writing populations and character-form coverage. They therefore provide evidence about performance under two dataset conditions, rather than an experiment that trained on one population and transferred directly to the other. For the present comparison, the implication is to keep the source images, preprocessing and partitions identical across model variants.

A joint label space introduces an additional distinction between the number of classes and the number of examples. Our source pool contains

ten digit classes with 60,000 images and 28 letter classes with 13,440 images. Pooling all examples gives digits greater influence on average loss and overall accuracy, even though letters occupy more output classes. This arithmetic motivates family-specific evaluation. It also defines a focused question that architecture comparisons alone do not resolve: whether changing the allocation of training loss improves the less accurately recognized family under otherwise matched conditions.

2.3 Balancing frequent examples and weaker families

Balancing methods change which training errors receive attention. Buda, Maki and Mazurowski [10] systematically varied class imbalance and compared corrective strategies in CNN experiments using MNIST, CIFAR-10 and ImageNet. Their study examined oversampling, undersampling, two-phase training and decision-threshold adjustments, with oversampling performing strongly across the evaluated settings. The relevant methodological lesson is to compare imbalance treatments under controlled data conditions, because the treatment itself changes the learning problem.

Cui and colleagues [5] addressed a more specific issue: additional examples from an already frequent class need not contribute proportionally new information. Their class-balanced loss weights classes using an *effective number of samples*, a quantity that grows more slowly as the class count increases. They evaluated the approach on long-tailed CIFAR datasets and iNaturalist. This provides a principled precedent for modifying loss weights according to representation. The weighting unit in their method is an individual class. Here, equal-family weighting instead assigns half of the loss to the mean digit error and half to the mean letter error; those weights are distributed across ten and 28 classes, respectively.

Representation and difficulty also need separate treatment. A family with fewer examples is not necessarily the family with the larger current loss. Sagawa and colleagues [6] studied distributionally robust optimization over predefined groups, aiming to improve the worst group's performance. Their experiments showed that minimizing worst-group training loss can still leave poor worst-group generalization, and that regularization and early stopping materially affect the outcome. This connects group-aware objectives to model selection: increasing attention to a group during training must be assessed using held-out group performance.

The present experiment separates these mechanisms. Equal-family loss tests a fixed allocation of attention; a smooth maximum of family losses tests an allocation that increases attention to the higher-loss family. The latter is a log-sum-exp objective specified in Section 3; Sagawa and colleagues' work supplies the motivation to examine

worst-group behavior. Family macro-F1 then measures the outcome after training, with the lower of the two family scores serving as the primary endpoint. This separates the training intervention from the evidence used to judge it.

2.4 Organizing prediction through a hierarchy

A hierarchy changes how a prediction is constructed. A flat classifier assigns probabilities directly to all symbol classes. A hierarchical classifier can first estimate a broader category and then estimate identity within that category. For Arabic digits and letters, the broad category has a clear meaning: whether the image belongs to the digit family or the letter family. The practical question is whether representing this relationship explicitly improves the final 38-class decision.

Yan and colleagues [7] investigated coarse-to-fine image recognition through a hierarchical deep convolutional neural network, HD-CNN. Their design organized visually confusable classes into coarse groups, combined a coarse classifier with specialized fine classifiers, and shared low-level convolutional computation. The approach used probabilistic prediction aggregation and was evaluated on CIFAR-100 and ImageNet. It established a concrete mechanism for combining broad discrimination with specialized recognition. The grouping in the present study is instead fixed by symbol type, and the specialization occurs in output heads attached to a common compact feature extractor.

This distinction makes the hierarchy testable without requiring a large ensemble. Our family head and two conditional heads produce one distribution by multiplying each family probability by its within-family class probabilities. A hard routing decision would discard the other family's candidates; the soft construction retains both. It also exposes a possible trade-off: lowering the probability of the correct family lowers every candidate probability within that family. Whether specialization compensates for that interaction is an empirical question.

The flat and hierarchical variants therefore use the same feature-extractor architecture, training schedule and checkpoint rule. A parameter-matched flat control addresses the additional output-head parameters. These comparisons distinguish the effect of organizing the prediction from the effect of increasing parameter count, and allow hierarchy to be examined both with pooled loss and with equal-family loss.

2.5 Testing recognition when image quality changes

Recognition accuracy on the original test images and sensitivity to altered image quality are separate properties. Dodge and Karam [11] evaluated

four ImageNet-trained neural networks under distortions including blur, noise, contrast changes and compression. Their experiments found substantial sensitivity to blur and noise, with different behavior for other distortions. This established the value of examining transformations individually: an aggregate corruption score can conceal which image change disrupts a recognizer.

Hendrycks and Dietterich [12] developed standardized corruption and perturbation benchmarks to make such comparisons reproducible. ImageNet-C evaluates recognition under fixed corruption types and severities, while ImageNet-P examines prediction stability along perturbation sequences. Mu and Gilmer [13] brought a related benchmark design to handwritten digits through MNIST-C, releasing fifteen corrupted versions of MNIST. These contributions motivate holding the corrupted inputs fixed across models and reporting the transformation alongside the performance measure.

Arabic handwriting studies have also included degradation tests. Attar [14] compared a CNN with conventional classifiers on a 28-class Arabic character dataset and reported robustness experiments involving Gaussian blur, contrast variation, partial occlusion and displacement of dots. This is a direct Arabic-script precedent for investigating how image changes affect recognition. Its character-only data and evaluation design differ from the joint digit–letter experiment here. The additional question in the present study is how the same specified corruption affects each family and whether family-aware training changes the resulting loss of performance.

Transformation definitions are essential to that comparison. Additive noise changes local pixel intensities, blur redistributes them spatially, and contrast scaling depends on the intensity about which it is centered. In our protocol, contrast is centered at 0.5, so reducing contrast also raises the intensity of a black background. The six conditions therefore test particular, reproducible input changes. Reporting family scores for each condition makes it possible to distinguish a change in clean recognition from a change in sensitivity to those inputs.

2.6 From prior capabilities to a controlled comparison

The reviewed literature supplies the components of the research argument: CNNs learn Arabic symbol features; balancing changes the allocation of training loss; hierarchy organizes class probabilities; and corruption tests expose sensitivity to image changes. These capabilities motivate a comparison in which the data and computational design are shared, so that the effect of each intervention can be examined separately. Table 1 summarizes how the prior evidence informs that comparison.

Repeated training is part of the comparison rather than an additional architecture feature.

Bouthillier and colleagues [15] examined variability in machine-learning benchmarks arising from factors including sampled data, parameter initialization and hyperparameter choices. Their analysis demonstrates why a ranking based on one favorable run can misrepresent an algorithm comparison. Here, five matched random seeds provide repeated observations of each within-protocol contrast, and paired differences expose how consistently an intervention changes the primary metric.

The unresolved relationship addressed by this study is consequently specific: under the same joint Arabic-symbol task, does equal-family weighting, hierarchical factorization or smooth worst-family emphasis improve the weaker family's recognition, and does that effect persist under fixed image corruptions? The response is a six-configuration experiment with a common CNN architecture, protected test partition, validation-based checkpoint selection, family-disaggregated metrics and a parameter-count control. This design links each research question to an observable contrast. The following section defines the data transformations, model outputs, objectives and evaluation rules that make those contrasts interpretable.

3 Materials and Methods

The experiment varies two parts of a joint recognizer: how its output probabilities are organized and how training errors contribute to its objective. Every configuration receives the same prepared symbol images, extracts a 128-dimensional feature vector with the same CNN architecture, and produces probabilities for 38 labels. Training uses a development partition, checkpoint selection uses a separate validation partition, and the selected model is then evaluated on the held-out test partition. This sequence separates the learning intervention from the evidence used to assess it.

3.1 Data sources and image preparation

MADBase supplies Arabic handwritten digits and the Arabic Handwritten Characters Dataset (AHCD) supplies isolated letters [16, 17]. The public distributions contain 60,000 digit and 13,440 letter training candidates, together with 10,000 digit and 3,360 letter test images. Digits retain global labels 0–9. The 28 letter identities occupy global labels 10–37; within the letter family, subtracting ten gives indices 0–27. The shared label space enables one final prediction to be assessed across both families.

The native digit and letter images have dimensions 28×28 and 32×32, respectively. Each image is resized to 64×64 grayscale using bilinear interpolation, and its pixel values are divided by 255 to obtain intensities in $[0, 1]$. AHCD images are transposed according to the fixed orientation setting in the preprocessing record. Corner-pixel checks

Table 1 How the reviewed evidence motivates the controlled comparison. Dataset-specific accuracies are discussed in the text with their evaluation conditions; this table summarizes mechanisms rather than ranking unlike benchmarks.

Research strand and sources	What the studies establish	Question carried into this study
Arabic CNN recognition: El-Sawy et al.; Mudhsh and Almodfer; Palatnik de Sousa [1–3]	Learned features, deeper networks and ensembles support digit or character recognition.	How does one shared classifier distribute quality across both families?
Dataset and architecture conditions: Al-Taani and Ahmad; Alghyaline; Altwaijry and Al-Turaiki [4, 8, 9]	Combined-source, adapted-architecture and population-specific evaluations define different recognition settings.	What changes when only the objective or output head varies on fixed data?
Representation-aware loss: Buda et al.; Cui et al. [5, 10]	Sampling and class-based weighting alter learning under unequal representation.	Does equal aggregate weight for digits and letters improve the weaker family?
Worst-group learning: Sagawa et al. [6]	Worst-group generalization responds to both objective design and regularization.	Does smooth family-risk emphasis improve held-out family macro-F1?
Coarse-to-fine prediction: Yan et al. [7]	Broad and specialized predictions can be combined in a hierarchical visual classifier.	Does a fixed digit/letter factorization help at matched model scale?
Corruption evaluation: Dodge and Karam; Hendrycks and Dietterich; Mu and Gilmer; Attar [11–14]	Specified image transformations reveal condition-dependent recognition behavior.	Do family-aware interventions change losses under identical corruptions?
Benchmark variability: Bouthillier et al. [15]	Variation in training and evaluation affects algorithm comparisons.	Are paired differences consistent across the five training seeds?

retained the original polarity of both sources. The resulting input has one intensity channel and a common spatial shape across the two datasets.

3.2 Partition construction and exact-image separation

Partition construction keeps repeated images from crossing the model-development and evaluation boundaries. A fingerprint of each processed image is first compared across the source training and test sets. All 13,360 source test rows are retained; 23 matching digit rows are excluded from the source training pool before validation is formed. The remaining 73,417 images are divided using a class-stratified, duplicate-grouped split with a target validation fraction of 0.15 and split seed 2026. All copies of an identical processed image enter the same development subset.

Table 2 reports the resulting 62,403 training and 11,014 validation images. Their totals follow the duplicate-group assignments rather than an exact row-by-row percentage. The partition audit found no shared processed-image fingerprint between any two of the training, validation, and

test sets. This construction gives every model the same image-level separation and leaves the original test set intact.

3.3 Shared feature extractor and output organization

Each training run fits its own parameters within the common feature-extractor architecture. The encoder applies four convolutional blocks with 16, 32, 64, and 128 channels. Each block contains a 3×3 convolution with padding, a rectified linear unit (ReLU), batch normalization, 2×2 maximum pooling, and dropout with probability 0.2. ReLU retains positive activations; pooling reduces spatial size; dropout randomly suppresses activations during training. Global average pooling averages each final feature map, producing the vector $z = f_\theta(x) \in \mathbb{R}^{128}$ for input image x and encoder parameters θ. The implementation uses batch-normalization epsilon 10^{-3} and momentum 0.01.

A flat output head converts this vector directly into 38 class probabilities through one linear layer and a softmax. The scores before the softmax are called logits. The hierarchical alternative in Fig. 1 uses three linear heads: one estimates digit/letter membership, one estimates identity among the ten digits, and one estimates identity among the 28 letters. Multiplying the family probability by the corresponding conditional class probability gives the final probability of each symbol. Both families remain candidates until the largest of the 38 final probabilities is selected.

Let $g(y) = D$ for a digit label $0 \le y \le 9$ and $g(y) = L$ for a letter label $10 \le y \le 37$. The within-family indices are $\kappa_D(y) = y$ and $\kappa_L(y) = y - 10$. The two-class family softmax is q_ϕ, with parameters ϕ; the conditional softmax for family g is r_{g,ψ_g}, with parameters ψ_g. Their joint distribution is

$$\begin{aligned} p(y \mid x) &= q_\phi\big(g(y) \mid z\big) \\ &\quad \times r_{g(y),\psi_{g(y)}}\big(\kappa_{g(y)}(y) \mid z\big), \qquad (1) \\ z &= f_\theta(x). \end{aligned}$$

For example, global letter label 14 uses the letter-family probability and the conditional probability at letter index 4. This is a constructed label-mapping illustration. The probabilities over all digits sum to the digit-family probability, and those over all letters sum to the letter-family probability; the resulting 38-class distribution therefore sums to one.

The flat reference has 102,534 trainable parameters. The hierarchical model has 102,792, an increase of 258 from its family head. A parameter-matched flat control adds two learned, feature-dependent affine offsets to the ordinary flat logits, applying one offset to every digit logit and the other to every letter logit. Its offsets are initialized to zero. Each offset can be absorbed into the corresponding flat linear weights, so this control retains

Table 2 Source and model-development partitions. Exact matches to test images are removed from the source training pool before the grouped training/validation split; the original test set is retained.

Partition	Digits	Letters	Total
Source training candidates	60,000	13,440	73,440
Quarantined from source training	23	0	23
Training used	50,979	11,424	62,403
Validation used	8,998	2,016	11,014
Original held-out test	10,000	3,360	13,360

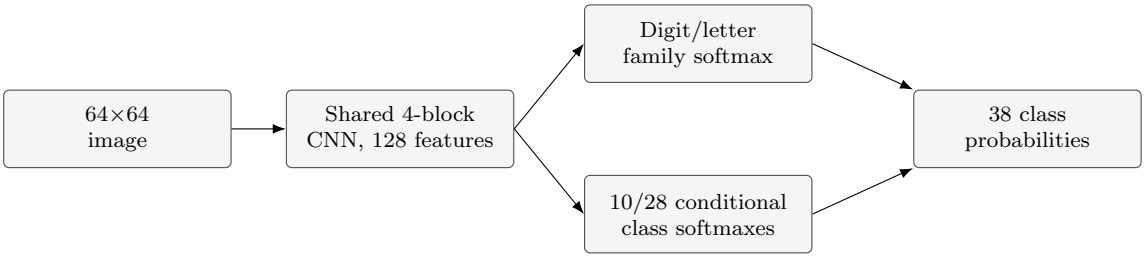


Fig. 1 Hierarchical prediction in B2, B3, and P. One image encoder supplies both the digit/letter head and the two conditional identity heads. Multiplication combines their outputs into a normalized distribution over all 38 symbols

the flat output's representational form while matching the hierarchical model's parameter count. It tests the effect of the additional parameterization alongside the output-factorization comparison.

3.4 How training attention is allocated

A loss function specifies how prediction errors influence each parameter update. For image-label pair (x_i, y_i), the negative log likelihood is $e_i = -\log p(y_i \mid x_i)$. Ordinary pooled cross entropy averages e_i over all images in the minibatch, giving each image equal weight. Equal-family training first averages within the digit and letter subsets, then assigns half of the objective to each family. If $\mathcal{B}_D$ and $\mathcal{B}_L$ denote those minibatch subsets, the two family losses and their balanced combination are

$$\ell_g = \frac{1}{|\mathcal{B}_g|} \sum_{i \in \mathcal{B}_g} e_i, \quad g \in \{D, L\},$$
$$\ell_{\text{bal}} = \tfrac{1}{2}(\ell_D + \ell_L). \tag{2}$$

This changes the aggregate attention given to each family while retaining all sampled images. It differs from giving each of the 38 individual classes an equal share of the objective.

The smooth worst-family variant shifts additional attention toward whichever family currently has the larger mean loss. A temperature $\tau > 0$ controls how closely a differentiable log-sum-exp term follows that larger loss. A mixture weight λ combines this term with equal-family training:

$$\ell_{\text{smooth}}(\tau) = \tau \log\left(\frac{e^{\ell_D/\tau} + e^{\ell_L/\tau}}{2}\right),$$
$$\tau > 0, \tag{3}$$

$$\ell_P = (1-\lambda)\ell_{\text{bal}} + \lambda\ell_{\text{smooth}}(\tau), \qquad \lambda = 0.5, \quad \tau = 0.2. \tag{4}$$

The derivative of the smooth term assigns softmax weights to the two family losses, so the higher-loss family receives more gradient weight from that term. The normalization by two makes the smooth term equal to the common family loss when both family losses are equal. These definitions distinguish fixed equal-family weighting from adaptive emphasis on current family difficulty.

3.5 Controlled configurations and checkpoint selection

The six configurations in Table 3 combine the output and loss choices so that each planned comparison changes a named mechanism. B0 is the flat, pooled-loss reference; B1 changes only its loss to equal-family weighting. B2 and B3 use the hierarchical output with pooled and equal-family losses, respectively. P adds smooth worst-family emphasis to the balanced hierarchical configuration. C is the parameter-matched flat control. These labels are used consistently in the results tables and figures.

Each configuration was trained for 20 epochs with seeds $\{7, 17, 29, 43, 61\}$, Adam optimization at learning rate 10^{-3}, and minibatches of 128 images. A seed-indexed sampler preserves the natural family mixture, requires both families in each complete minibatch, and supplies the same minibatch schedule to all six configurations. The incomplete remainder at each epoch is omitted. Training uses the prepared clean images; the noise, blur, and contrast transformations are reserved for evaluation. All 30 training records identify PyTorch version `2.11.0+cu130` and CUDA execution.

After each epoch, the model is scored on validation worst-family macro-F1, defined in Section 4.1. The checkpoint with the greatest value is retained, with the earlier epoch retained in a tie. The held-out test evaluation begins after all 30 training and validation records are complete. Thus, every reported test result corresponds to a checkpoint selected through the same development procedure. The frozen experiment record identifies the partitions, model configurations, seeds, and source-code hashes used for that procedure. The evaluation below defines how each selected model is scored and compared.

4 Evaluation Design

The evaluation asks whether a training intervention improves the weaker symbol family and whether that improvement persists after a fixed change to the image. A training run is one configuration fitted with one random seed. Its selected checkpoint yields seven evaluation records: one for clean test images and six for corrupted versions of those same images. The experiment therefore comprises 30 trained models and 210 condition-level records, with model comparisons paired by training seed.

Table 3 Six controlled configurations. CE denotes pooled cross entropy. P combines equal-family and smooth worst-family loss; C matches the hierarchical parameter count.

Arm	Output	Loss	Planned comparison
B0	Flat 38-way	Pooled CE	Reference
B1	Flat 38-way	Equal-family	B1−B0: balancing
B2	Hierarchical 2/10/28	Pooled CE	B2−B0: factorization
B3	Hierarchical 2/10/28	Equal-family	B3−B1: factorization with balancing
P	Hierarchical 2/10/28	Smooth worst-family	P−B3: risk emphasis
C	Flat + 2 group offsets	Pooled CE	B2−C: matched parameter count

4.1 Family-specific outcomes and paired comparisons

F1 combines precision and recall for an individual class. Precision measures how often predictions of that class are correct; recall measures how often its true examples are recovered. A family macro-F1 score averages the individual class F1 values within the ten digit classes or the 28 letter classes, giving each class in that family equal weight. All false positives and false negatives are counted in the full 38-label prediction space. A letter predicted as a digit therefore contributes an error to the relevant class scores on both sides of the family boundary.

The primary endpoint is $F_{\text{worst}} = \min(F_{1,D}, F_{1,L})$, where $F_{1,D}$ and $F_{1,L}$ are the digit and letter class-macro F1 scores. A positive change in this endpoint means that the lower-performing family improves. Overall accuracy, digit accuracy, letter accuracy, and 38-class macro-F1 provide complementary outcomes. The distinction between sample-weighted accuracy and class-averaged F1 is essential when the families contain different numbers of images and classes.

H1 compares B1 with B0. H2 compares B2 with B0, B3 with B1, and B2 with the parameter-matched control C. The clean P-minus-B3 comparison is a supplementary check on smooth worst-family training; H3's designated endpoint concerns corruption retention. For every contrast, a difference is computed within each of the five matched seeds before averaging. Descriptive 95% Student-t intervals use the sample standard deviation of those five differences and four degrees of freedom. The archive also contains a class-stratified paired image bootstrap for the first seed; the main comparisons use the five-seed intervals.

4.2 Fixed image transformations and retention

All models receive identical corrupted test arrays generated after pixel scaling, with corruption seed 1234. Additive Gaussian noise uses intensity standard deviations $\sigma \in \{0.05, 0.10\}$ and clips intensities to $[0, 1]$. Gaussian blur uses spatial standard deviations $\sigma \in \{0.8, 1.3\}$ pixels. Midpoint-centered contrast uses scales $\alpha \in \{0.75, 0.50\}$ through $x' = 0.5 + \alpha(x - 0.5)$, followed by clipping.

The labels and image order remain unchanged in every condition.

The contrast mapping changes the background level together with the difference between stroke and background intensities. Substituting a black pixel, $x = 0$, gives $x' = 0.125$ at $\alpha = 0.75$ and $x' = 0.25$ at $\alpha = 0.50$. These values make the tested operation explicit: the model sees a lighter background as well as a reduced intensity range. Noise and contrast severities are measured in normalized intensity units, whereas blur severity is measured in pixels.

For H3, let F_m^{clean} and F_m^{corrupt} denote worst-family macro-F1 for model m on clean images and one corrupted condition. We compare the change from clean performance for P with the corresponding change for B3:

$$\Delta_{\text{ret}} = (F_P^{\text{corrupt}} - F_P^{\text{clean}}) - (F_{B3}^{\text{corrupt}} - F_{B3}^{\text{clean}}). \tag{5}$$

A positive value means that P retains more F1 than B3 relative to its own clean baseline; a negative value favors B3. The designated summary first takes an equal-weight average of this contrast over the six corruptions within each seed and then summarizes the five seed-level values. Retention here is an absolute F1 change, with a directly defined comparison to clean performance.

4.3 Experimental record and verification

The preserved record contains all 30 training histories, initial states, selected checkpoints, and per-image prediction files. Recomputing accuracy, class-macro F1, and family-confusion counts from the predictions reproduced all 210 evaluation records. The prediction and selected-checkpoint hashes matched their saved metadata, and the processed partitions matched the frozen data hashes. Additional inference checks on samples from three saved checkpoints reproduced predictions for clean, noise, blur, and contrast inputs. The supplementary audit records these checks and the recalculated paired summaries, providing a traceable basis for the findings below.

5 Results

5.1 Clean recognition is less accurate for letters

The six configurations achieve similar overall clean-image accuracy, approximately 98.2–98.4%, and the letter family has the lower macro-F1 in every configuration (Table 4). The flat reference B0 has 0.9835 overall accuracy, 0.9901 digit macro-F1, and 0.9637 letter macro-F1. Its worst-family score is therefore determined by the letters. The family decomposition makes the clean-image disparity visible alongside the high pooled accuracy.

Table 4 Clean held-out results over five seeds, shown as mean ± sample standard deviation. F1 is calculated from the full 38-label confusion matrix.

Model	Overall accuracy	Digit macro-F1	Letter macro-F1	Worst-family F1
B0: Flat CE	0.9835 ± 0.0032	0.9901 ± 0.0024	0.9637 ± 0.0109	0.9637 ± 0.0109
B1: Flat balanced	0.9836 ± 0.0011	0.9896 ± 0.0014	0.9658 ± 0.0070	0.9658 ± 0.0070
B2: Hierarchical CE	0.9837 ± 0.0025	0.9902 ± 0.0023	0.9643 ± 0.0045	0.9643 ± 0.0045
B3: Hierarchical balanced	0.9822 ± 0.0046	0.9880 ± 0.0036	0.9653 ± 0.0085	0.9653 ± 0.0085
P: Smooth-worst	0.9835 ± 0.0026	0.9898 ± 0.0021	0.9651 ± 0.0057	0.9651 ± 0.0057
C: Matched flat	0.9840 ± 0.0029	0.9903 ± 0.0024	0.9650 ± 0.0055	0.9650 ± 0.0055

B1 has the largest mean worst-family macro-F1, 0.9658, compared with 0.9637 for B0. The parameter-matched flat model C has the largest rounded overall accuracy, 0.9840, while its worst-family macro-F1 is 0.9650. The two metrics consequently select different mean leaders. Table 4 reports sample standard deviations over five seeds; the paired contrasts below assess each intervention under matched seeds.

5.2 Balancing and hierarchy yield small, mixed clean-image differences

Equal-family weighting in the flat classifier produces a mean B1-minus-B0 change of +0.00212 worst-family F1, or +0.212 percentage points. Its descriptive 95% interval extends from −0.756 to +1.179 percentage points (Table 5). The paired seed outcomes contain both gains and losses. H1's predicted improvement is therefore favorable on average and inconsistent across the repeated training runs.

Hierarchical prediction produces a mean change of +0.00054 under pooled loss (B2 minus B0) and −0.00053 under balanced loss (B3 minus B1). Against the parameter-matched flat control, B2 minus C is −0.00077. All three H2 intervals cross zero, and their mean directions differ between comparisons. The clean P-minus-B3 diagnostic is likewise small, at −0.024 percentage points. Figure 2 places all five clean-image contrasts on the same scale so that their mean differences can be read together with their seed variability.

5.3 Corruptions cause larger losses than the clean-model differences

The fixed transformations change recognition much more than the clean-image interventions (Table 6). For B0, worst-family macro-F1 decreases from 0.9637 on clean images to 0.9058 under blur with $\sigma = 0.8$ pixels and to 0.6011 under blur with $\sigma = 1.3$ pixels. Under noise with $\sigma = 0.05$, it reaches 0.0062; midpoint-centered contrast with $\alpha = 0.75$ gives 0.0359. Several stronger noise and contrast results are near the F1 floor.

The model differences also depend on the transformation. Among the four configurations displayed in Table 6, B1 has the highest mean score

Table 5 Paired five-seed differences in clean worst-family macro-F1, expressed in percentage points. Intervals are descriptive 95% Student-t intervals across the five matched seeds. The clean smooth-worst row is supplementary to H3's corruption-retention endpoint.

Comparison	Mean difference (pp)	95% interval (pp)
H1: balancing (B1−B0)	+0.212	[-0.756, +1.179]
H2: hierarchy, pooled (B2−B0)	+0.054	[-1.370, +1.477]
H2: hierarchy, balanced (B3−B1)	-0.053	[-1.257, +1.152]
Smooth-worst: clean diagnostic (P−B3)	-0.024	[-1.418, +1.370]
Parameter-count control (B2−C)	-0.077	[-0.969, +0.815]

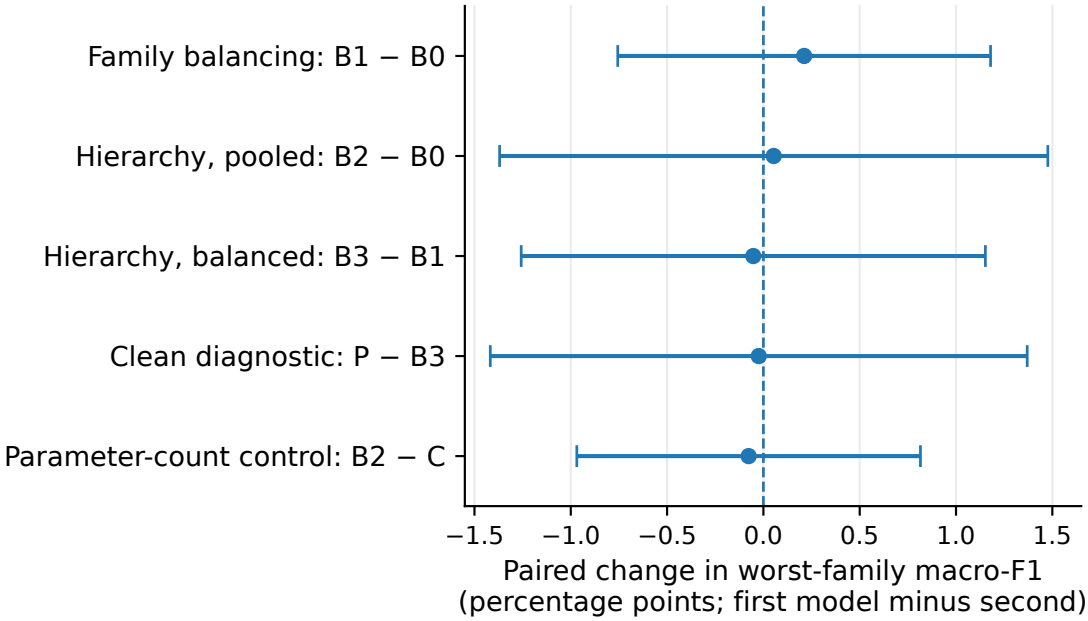


Fig. 2 Paired differences in clean worst-family macro-F1. Points are means over five matched training seeds and bars are descriptive 95% Student-t intervals, in percentage points. Each row names the subtraction order; positive values favor the first model. The clean P-minus-B3 comparison is a supplementary diagnostic; H3 uses the six-corruption retention contrast

at both blur levels, whereas B3 has the highest mean at noise $\sigma = 0.05$ and contrast $\alpha = 0.75$. Figure 3 reports all six configurations under the same seven conditions. The visual comparison separates a general drop from clean performance from the smaller differences between models within each perturbation.

For H3, the mean difference in clean-to-corrupt F1 changes, P relative to B3, is -0.03168 when averaged with equal weight over the six conditions. Its descriptive five-seed interval is $[-0.09548, +0.03212]$. The negative mean favors the balanced hierarchical control, and the interval crosses zero. The smooth worst-family objective therefore provides no consistent improvement in the designated corruption-retention endpoint.

Table 6 Worst-family macro-F1 under clean and fixed corrupted-image conditions (means over five seeds). Values near zero indicate a floor effect. All six model variants are plotted in Fig. 3; the four shown here include the main family-aware comparison.

Condition	B0	B1	B3	P
Clean	0.9637	0.9658	0.9653	0.9651
Noise $\sigma = .05$	0.0062	0.0100	0.0134	0.0008
Noise $\sigma = .10$	0.0000	0.0018	0.0000	0.0000
Blur $\sigma = .8$	0.9058	0.9347	0.8850	0.8337
Blur $\sigma = 1.3$	0.6011	0.7218	0.5629	0.4928
Contrast $\alpha = 0.75$	0.0359	0.0631	0.0836	0.0262
Contrast $\alpha = 0.50$	0.0004	0.0000	0.0009	0.0008

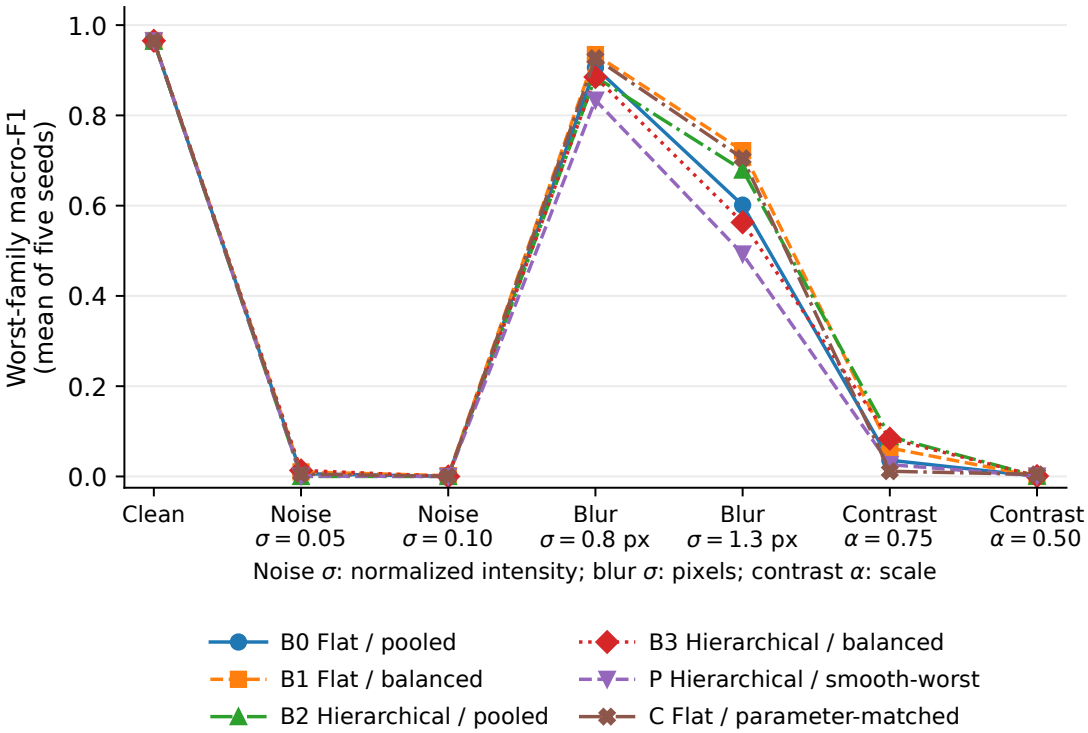


Fig. 3 Mean worst-family macro-F1 across five training seeds for each configuration and test condition. Noise labels give intensity standard deviation σ on the $[0, 1]$ scale; blur labels give spatial standard deviation σ in pixels; contrast labels give midpoint-centered scale α. Distinct markers identify the six models. Each condition uses identical transformed images across models; H3 is evaluated separately by paired retention

6 Discussion

6.1 Family-specific evaluation changes the design question

The family scores expose a recognition problem that remains after high overall accuracy is reached. Digits attain higher macro-F1 than letters in all six configurations, and the gap remains around two to three F1 percentage points. The lower letter score determines the primary endpoint, so an intervention is assessed by its effect on the less

accurately recognized family rather than its ability to improve a pooled score dominated by more numerous test images. This makes the design question explicit: which change helps the family that still makes more class-averaged errors?

Earlier separate-dataset CNN and ensemble studies establish strong Arabic recognition capability [1, 2]; the combined-source study by Al-Taani and Ahmad establishes a related joint-learning setting [4]. The present evidence adds a comparison within one fixed joint task. Its distinguishing observation is that equal-family weighting, output factorization, and additional loss emphasis do not produce a consistent gain under the shared protocol. The result refines the choice of mechanism rather than ranking the separate-dataset studies by unlike accuracy figures.

6.2 Training emphasis and hierarchical structure have different roles

Equal-family weighting increased the mean weaker-family score slightly. The consistency question is answered by the paired runs: the gain changes sign across seeds and is small relative to their variation. This complements the distinction between representation-based weighting and held-out group performance developed by Cui and colleagues and Sagawa and colleagues [5, 6]. Giving a family greater weight changes the optimization target; whether it improves recognition is an empirical question measured after checkpoint selection.

The hierarchical comparison concerns the organization of probabilities. The digit/letter factorization is meaningful, and its joint probabilities retain all 38 candidates. The clean-image contrasts show that this organization supplied no consistent advantage over the flat output, including the parameter-matched control. Yan and colleagues' HD-CNN uses coarse groups of visually confusable classes and specialized classifiers [7]; our fixed semantic families and shared compact encoder represent a different specialization choice. A useful follow-up is to test whether grouping by observed visual confusion changes the outcome under the same evaluation controls.

One possible explanation for the small clean-image changes is that the shared encoder already represents the distinctions needed by both output forms. Another is that different random training trajectories outweigh the effect of the modest intervention at this accuracy level. These explanations are hypotheses for targeted representation and training-dynamics analyses. The controlled results themselves establish the magnitude and variability of the output-head and objective changes.

6.3 The corruption response identifies the next experimental priority

The perturbation results reveal an input-sensitivity problem larger than the clean-model differences. Every configuration uses the same feature-extractor architecture, input preparation, and clean-image training procedure, and every one encounters the same transformed test images. Severe losses across these configurations therefore point to a vulnerability common to the evaluated recognition pipeline. Altering the allocation of loss or factorizing the output did not remove that vulnerability in the tested conditions.

The contrast definition helps specify what should be investigated. Its arithmetic raises the black background as it reduces the stroke-to-background range. Consequently, the measured response is to a controlled photometric shift with both effects. Prior work by Dodge and Karam, Hendrycks and Dietterich, and Attar also demonstrates the value of defining and evaluating image degradations explicitly [11, 12, 14]. Here, the next diagnostic should separate background offset from contrast scaling, inspect feature activations and batch-normalization responses, and compare each with the saved clean-image prediction. This would turn the observed failure into a test of its mechanism.

The findings also suggest a focused training comparison: retain the present architecture and evaluation protocol while adding matched noise, blur, or photometric augmentation during training. That experiment would test whether exposure to these transformations improves the same family-specific endpoint. A further evaluation with verified writer identifiers would examine handwriting-style transfer separately from the current family and dataset-source comparison. These follow-ups follow directly from the measured sensitivity and the study's generalization boundaries.

7 Conclusion

The controlled 38-class comparison shows that high overall Arabic-symbol accuracy can coexist with a persistent difference between digit and letter recognition. Across six CNN configurations and five matched seeds, the best mean worst-family macro-F1 was 0.9658 for flat equal-family training, compared with 0.9637 for the pooled flat reference. The H1 and H2 comparisons produced small, mixed effects with intervals crossing zero. The H3 corruption-retention contrast averaged -0.0317 F1 against balanced hierarchical training.

The contribution is a reproducible separation of family-loss allocation, output hierarchy, and parameter count under one evaluation design. Its main finding is that these interventions changed

clean recognition inconsistently, while the specified noise and photometric transformations caused much larger losses. Subsequent development should prioritize identifying that shared sensitivity and testing matched training-time augmentation, with family-specific outcomes retained as the decision criterion.

8 Data and Software Availability

MADBase and AHCD are available from the cited source repositories [16, 17]. Study-specific materials accompany this manuscript as three numbered online resources. Online Resource 1 (`ESM_1.pdf`) reports experimental results, and Online Resource 2 (`ESM_2.pdf`) describes reproducibility. Online Resource 3 (`ESM_3.zip`), release `IJDAR-CNN-2026-10-28-R1`, contains the frozen implementation, all 30 training and evaluation records, initial and selected checkpoints, per-image predictions for the 210 conditions, split provenance, source receipts, and audit outputs. Its `README.md` maps the six configurations and five seeds to their files and gives commands for recomputing saved-prediction metrics and reconstructing the prepared partitions from the upstream CSV distributions. The archive manifest identifies each file by its SHA-256 digest.

The prepared image arrays are reconstructed from the original public distributions using the recorded orientation, interpolation, grouped split, and source hashes. The source archives and large processed image arrays remain in the preserved full experimental archive; Online Resource 3 supplies their identities and reconstruction procedure. It includes the per-image evidence needed to recalculate the reported classification metrics without retraining.

An earlier course-project CNN report is separately archived at Zenodo [18]. Its historical 98.86% peak validation result belongs to that earlier experiment. The present study's six-configuration held-out results are documented in Online Resources 1–3.

9 Limitations

The conclusions concern one joint task, one compact feature extractor, fixed training settings, and five random seeds. The paired Student-t intervals summarize training-seed variability; they do not establish population-level superiority or equivalence. The image bootstrap provided for the first seed does not account for within-writer dependence. The verification audit recalculated saved evaluations and repeated selected checkpoint inference, rather than retraining all 30 models from initialization.

The train/validation split groups exact processed-image duplicates rather than writer identities. The original test partitions are preserved,

and writer identities were not independently audited. Digits and letters come from different source datasets, so family differences can include acquisition and population differences. Within-partition duplicates remain: 4,245 extra identical rows in training, 789 in validation, and one in testing. Near-duplicate and writer-related dependence were not assessed. The task covers isolated symbols; connected-word or line recognition requires additional evaluation.

The six synthetic corruption settings have not been calibrated to the distribution of a particular scanner or document collection. Midpoint-centered contrast changes both background intensity and stroke contrast. Saved predictions and sampled checkpoint replay confirm the severe response to these specified transforms; their causal mechanism and transfer to naturally degraded handwriting remain unresolved. Controlled background, contrast, and augmentation studies would address this boundary directly.

Statements and Declarations

Funding. No funding was received for this research.

Competing interests. The author declares no competing interests.

Author contributions. Meshaal Mouawad is the sole listed author and takes responsibility for the final manuscript, analysis and interpretation.

Ethics approval and consent. This study analyzes publicly distributed handwriting datasets and contains no new recruitment of participants.

Generative AI declaration. Wiley Scholar Gateway was used to support literature discovery, review preparation, and source-oriented synthesis. ChatGPT was used to assist with manuscript drafting and revision and other research publishing tools. The author reviewed and revised AI-assisted outputs, checked supporting sources and experimental evidence, and takes full responsibility for the accuracy, integrity, interpretation, and final content of this work.

Supplementary information

Online Resource 1. Experimental results: clean-image summaries, paired comparisons, and corruption measurements (`ESM_1.pdf`).

Online Resource 2. Reproducibility: data preparation, model configurations, execution settings, and integrity checks (`ESM_2.pdf`).

Online Resource 3. Versioned experimental record and implementation, including predictions, checkpoints, provenance, and metric-recomputation commands (`ESM_3.zip`).

References

[1] A. El-Sawy, M. Loey, and H. El-Bakry, "Arabic handwritten characters recognition using convolutional neural network," *WSEAS*

Transactions on Computer Research, vol. 5, pp. 11–19, 2017. https://www.wseas.com/journals/articles.php?id=3300.

[2] I. Palatnik de Sousa, "Convolutional ensembles for Arabic handwritten character and digit recognition," *PeerJ Computer Science*, vol. 4, e167, 2018. https://doi.org/10.7717/peerj-cs.167.

[3] M. A. Mudhsh and R. Almodfer, "Arabic Handwritten Alphanumeric Character Recognition Using Very Deep Neural Network," *Information*, vol. 8, no. 3, article 105, 2017. https://doi.org/10.3390/info8030105.

[4] A. T. Al-Taani and S. T. Ahmad, "Recognition of Arabic handwritten characters using residual neural networks," *Jordanian Journal of Computers and Information Technology*, vol. 7, no. 2, pp. 192–205, 2021. https://doi.org/10.5455/jjcit.71-1615204606.

[5] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-Balanced Loss Based on Effective Number of Samples," in *CVPR*, 2019, pp. 9268–9277. https://doi.org/10.1109/CVPR.2019.00949.

[6] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang, "Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization," *ICLR*, 2020. https://openreview.net/forum?id=ryxGuJrFvS.

[7] Z. Yan, H. Zhang, R. Piramuthu, V. Jagadeesh, D. DeCoste, W. Di, and Y. Yu, "HD-CNN: Hierarchical Deep Convolutional Neural Networks for Large Scale Visual Recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2740–2748. https://doi.org/10.1109/ICCV.2015.314.

[8] S. Alghyaline, "Optimised CNN Architectures for Handwritten Arabic Character Recognition," *Computers, Materials & Continua*, vol. 79, no. 3, pp. 4905–4924, 2024. https://doi.org/10.32604/cmc.2024.052016.

[9] N. Altwaijry and I. Al-Turaiki, "Arabic handwriting recognition system using convolutional neural network," *Neural Computing and Applications*, vol. 33, pp. 2249–2261, 2021. https://doi.org/10.1007/s00521-020-05070-8.

[10] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018. https://doi.org/10.1016/j.neunet.2018.07.011.

[11] S. Dodge and L. Karam, "Understanding How Image Quality Affects Deep Neural Networks,"

in *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, 2016, pp. 1–6. https://doi.org/10.1109/QoMEX.2016.7498955.

[12] D. Hendrycks and T. Dietterich, "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations," *ICLR*, 2019. https://arxiv.org/abs/1903.12261.

[13] N. Mu and J. Gilmer, "MNIST-C: A Robustness Benchmark for Computer Vision," dataset, version 2, Zenodo, 2019. https://doi.org/10.5281/zenodo.3239543.

[14] E. T. Attar, "Deep convolutional neural network for isolated Arabic handwritten character recognition: design, evaluation, and comparative study," *Scientific Reports*, vol. 15, article 42467, 2025. https://doi.org/10.1038/s41598-025-26658-x.

[15] X. Bouthillier et al., "Accounting for Variance in Machine Learning Benchmarks," *Proceedings of Machine Learning and Systems*, vol. 3, 2021. https://proceedings.mlsys.org/paper_files/paper/2021/hash/0184b0cd3cfb185989f858a1d9f5c1eb-Abstract.html.

[16] M. Loey, "Arabic Handwritten Digits Dataset," public distribution repository, accessed October 2026. https://github.com/mloey/Arabic-Handwritten-Digits-Dataset.

[17] M. Loey, "Arabic Handwritten Characters Dataset," public distribution repository, accessed October 2026. https://github.com/mloey/Arabic-Handwritten-Characters-Dataset.

[18] M. Mouawad, "Deep Learning Optimization Methods for Arabic Handwritten Recognition Using CNN," Zenodo, archived project report, 2026. https://doi.org/10.5281/zenodo.20745650.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Supplementary.zip