Online Urdu Text-Line Recognition by Bridging Stroke Dynamics and Offline Representations

Ali Hussain[1 [0009−0009−9464−2947]], Rafay Ahmad[1], Momina Moetesum[1,2 [0000−0001−9465−4678]], Adnan Ul-Hasan[1,2], and Faisal Shafait[1,2 [0000−0002−0922−0566]]

[1] School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[2] Deep Learning Laboratory, National Center of Artificial Intelligence (NCAI), National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
{ahussain.bscs23seecs, momina.moetesum, faisal.shafait}@seecs.edu.pk

Abstract. Robust recognition of Urdu has been challenging due to its cursive structure and extensive ligature variations. Modern pen devices capture rich spatio-temporal signals; however, state-of-the-art Urdu research predominantly focuses on offline images. Online-first approaches based on transformer and vision-language architectures are impractical in low-resource settings, as the limited availability of labeled online Urdu data leads to overfitting. Moreover, existing online Urdu handwriting research has largely been restricted to ligature and word-level recognition, leaving line-level modeling underexplored. To address this gap, we present two contributions: (i) We collect a new dataset Online Urdu Handwriting Dataset-Lines (OUHD-L) comprising 2,403 online Urdu handwritten text lines using the Wacom IoT paper device, which records fine-grained pen trajectories. (ii) We encode online information in an image representation to leverage pre-trained offline Urdu text-line recognizers. On the proposed dataset, incorporating online-derived feature images reduces character error rate from 4.21% using ink-only input to 3.66% when online information is fused. These results demonstrate that converting online dynamics into image space effectively bridges the data gap while allowing the reuse of mature offline architectures. This strategy provides a feasible and scalable alternative to explicit online sequence modeling in low-resource cursive script recognition.

Keywords: Online handwriting recognition · Online urdu handwriting dataset · Stroke dynamics · Multimodal fusion · Low-resource HTR

1 Introduction

Urdu is the national language of Pakistan, spoken by over 230 million native speakers worldwide [8]. Pakistan's public institutions, legal courts, land reg-

Code: https://github.com/dll-ncai/Online-Urdu-HWR Dataset (OUHD-L): https://doi.org/10.5281/zenodo.20642162

istries, and government archives conduct the overwhelming majority of their documentation in handwritten Urdu, yet these records remain largely undigitised. Nastaliq is among the most complex scripts in active use: a right-to-left cursive system with a diagonally descending baseline, heavily overlapping glyphs, and over 24,000 context-sensitive ligature forms [16,27], as shown in Fig. 1. Individual characters cannot be reliably segmented in isolation, visual appearance changes dramatically with positional context, and the right-to-left direction poses additional challenges for sequence models trained on left-to-right scripts.

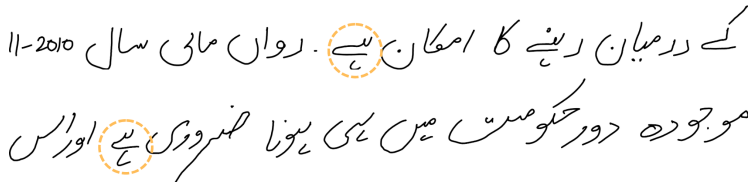

Fig. 1. A wide variety of Urdu ligatures, with further variation introduced by individual writing style. The annotated 'hai' shows ligature variation between two different writers.

Offline Urdu Handwritten Text Recognition (HTR) has received substantial attention in recent years. Convolutional Neural Network (CNN) - Recurrent Neural Network (RNN) pipelines [27,21] achieve competitive Character Error Rate (CER) on unconstrained offline benchmarks, and more recent unified transformer-based architectures [16] have demonstrated strong generalisation across printed and handwritten Urdu text, setting the current state of the art on NUST-UHWR at 6.2% CER. Despite this progress, all such systems operate exclusively on static images and therefore have no access to the online stroke dynamics that digital pen devices naturally record.

Online signals such as pen pressure, tilt, velocity, and stroke ordering carry discriminative cues that resolve visual ambiguities in cursive scripts [20], including those inherent to Urdu. Fig. 2 shows two visually similar ligatures that can be reliably distinguished by their characteristic pressure profiles. Modern IoT pen devices, such as the Wacom platform used here, record these spatio-temporal signals at sub-millimetre resolution and high temporal fidelity, yet the Urdu research community has made limited use of such data. Existing online HTR work for Urdu has focused exclusively on ligature- and word-level recognition; to our knowledge, no publicly available online Urdu text-line dataset exists, and no line-level online recogniser has been reported in the literature.

Urdu represents a low-resource setting for online handwriting: collecting large quantities of stroke-level annotated text lines is expensive, and no sizeable online Urdu corpus currently exists. This creates a fundamental mismatch between data availability and model complexity. Training an online-native transformer from scratch on a small corpus leads to severe overfitting, making the sophisticated architectures that have succeeded for well-resourced scripts, such as large

sequence-to-sequence models [26,13] and vision-language architectures [18], directly inapplicable here. At the same time, the offline community has spent years building well-regularised models on the largest available Urdu transcription corpora; these models carry rich, language-aware representations of Nastaliq that are directly transferable to the online domain and should not be abandoned in favour of training from scratch. Zero-shot transfer of such a well-regularised offline model to rendered online ink already achieves 10.65% CER on our dataset (Table. 3), confirming that the offline feature space generalises to the online domain, yet leaves substantial room for improvement through the temporal signals available at collection time.

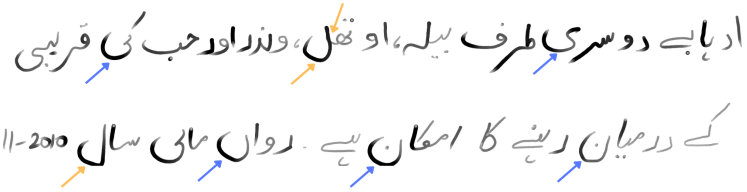

Fig. 2. Pressure difference observed in two different characters in a specific ligature setting. The 'noon gunah' and 'ye' exhibit very low pressure at the start of the curvature that progressively increases, whereas 'laam' maintains low pressure along its shaft but sustains strong pressure throughout its curvature.

As observed in Fig. 1 and Fig. 2, the challenge is not building online knowledge from scratch but injecting stroke dynamics into the offline model's input space to reuse its pre-trained representations. We encode pressure, velocity, tilt, and stroke order as spatially-aligned feature image channels, processed by independent CNN copies of [16] and fused via a learned gate, while the Transformer and its language model pre-training remain unchanged.

We address this by encoding pen dynamics as auxiliary feature images injected into the pre-trained offline model's input space, preserving its language-aware representations without architectural change. This paper makes two fundamental contributions to Urdu handwriting recognition:

- We introduce OUHD-L, the first publicly available online Urdu text-line dataset, comprising 2,403 handwritten lines from 311 writers collected via a Wacom IoT paper device at 500 Hz. Each line includes full stroke-trajectory metadata including pressure, tilt, pen height, tip thickness, and timestamp with writer-disjoint train/validation/test splits.
- We propose a representation that encodes online stroke dynamics (pressure, velocity, tilt, direction, curvature, and stroke order) as spatially-aligned feature image channels. These channels are fused with a pre-trained offline CNN-Transformer recogniser via a learned dynamic gate, enabling direct integration of temporal pen signals without any architectural modification. On OUHD-L, this reduces CER from 4.21% (ink-only fine-tuning) to 3.66%

(full fusion), a statistically significant improvement confirmed across five independent runs.

2 Related Work

We discuss earlier work in Urdu Printed and Handwritten text recognition in offline and online settings and explore gaps in online recognition.

2.1 Urdu Text Recognition

Maqsood et al. [16] addressed the label-scarce Urdu setting with a lightweight CNN-Transformer encoder paired with a decoder pre-trained on the Urdu News Corpus 1M [9], achieving state-of-the-art CERs across UPTI [19], URTI [23], and MMU-OCR-21, making it the only architecture designed specifically for the label-scarce Urdu setting and the backbone we adopt in this work.

For unconstrained offline handwriting, Ul-Hasan et al. [24] established the utility of Bidirectional LSTM with Connectionist Temporal Classification (CTC) for right-to-left cursive scripts. Zia et al. [27] introduced the NUST-UHWR benchmark alongside a convolutional-recursive architecture with an n-gram language model, reporting 5.82% CER. Riaz et al. [21] improved this to 5.31% by reframing the task as sequence-to-sequence learning with a Conv-Transformer encoder-decoder. Maqsood et al. [16] subsequently achieved 6.2% CER with their unified architecture. On the dataset side, PUCIT-OHUL [17] provides 7,309 annotated offline lines from 100 writers, reflecting the relative maturity of the offline data ecosystem. Collectively, the offline community has built well-regularised models that carry the richest Urdu-specific representations available. Critically, however, all of these systems operate on static images and have no mechanism to incorporate the temporal dynamics that a digital pen naturally captures: pressure profiles that distinguish visually similar ligatures, velocity patterns characteristic of Nastaliq's diagonal baseline, and stroke-order information that resolves spatial overlaps inherent to the script.

2.2 Online Handwriting Recognition

Online HTR systems address this gap by exploiting temporal pen-trajectory data, namely (x, y, t, p) signals, alongside or instead of rendered ink. Classical HMM-based methods gave way to deep learning: Carbune et al. [5] demonstrated strong CER on Latin and East Asian scripts using raw coordinate sequences. Multimodal fusion has since emerged as the most productive direction. Xu et al. [26] combine a stroke-sequence encoder and image encoder via Transformer cross-attention for state-of-the-art performance on IAMOn-DB, Lodh et al. [13] jointly embed image patches and stroke tokens in a shared attention space, and InkSight [18] encodes trajectories as coordinate token sequences for a Vision-Language Model (VLM), achieving zero-shot generalisation across scripts and devices. These systems confirm that temporal dynamics carry discriminative

information absent from static images, yet they are built on large, well-curated corpora for Latin, Chinese, Vietnamese, and Japanese scripts. IAMOn-DB alone contains over 13,000 annotated online lines; no comparable online Urdu corpus exists at the line level, making these training regimes directly inapplicable.

Online Urdu HTR remains nascent and almost entirely sub-line in scope. Early RNN and CNN-RNN systems addressed ligature- and character-level recognition [2]; neither scales to the text-line level, and no line-level online system has been reported. A 2025 survey explicitly characterises online line-level recognition as an open problem, attributing the gap to Urdu's complexity and the scarcity of trajectory-annotated data [3]. The contrast with offline resources is stark: NUST-UHWR [27], PUCIT-OHUL [17], and UNHD [1] together represent the most substantial offline Urdu collections available, yet no equivalent online corpus exists at the line level. Training an online-native model from scratch is not a viable option precisely because the data required to do so does not exist; and it is this same absence of large online corpora that makes the richly pre-trained offline models the most reliable knowledge source available. Among the available offline resources, PUCIT-OHUL provides either document-level images without line segmentation or isolated character images, with no line-level annotations or official splits. UNHD provides line-level images but similarly lacks official train/val/test splits, making reproducible pre-training infeasible and results incomparable with the established baseline. Arabic-script online HTR datasets such as ADAB and KHATT cover Naskh and related styles but do not include Nastaliq, the dominant script for Urdu. Nastaliq differs from Naskh in ways that matter for online recognition: its baseline descends diagonally from right to left rather than running horizontally, ligature overlaps are denser, and stroke trajectories follow a distinctive descending arc that produces unique velocity and curvature profiles absent from Naskh writing. Methods trained on ADAB or KHATT do not transfer directly to Nastaliq, and OUHD-L is therefore not a subset of the Arabic online HTR problem but an independently motivated dataset contribution. Our method therefore builds directly on the state-of-the-art offline backbone of [16], trained on NUST-UHWR, which is the only available resource combining line-level annotations, official splits, and sufficient scale for reliable pre-training, ensuring that observed gains are attributable solely to online feature encoding rather than to additional training data.

2.3 Encoding Online Signals as Image Features

The two preceding observations point to a third strategy: projecting online signals into the image space that offline models already operate in, injecting temporal dynamics into a mature system without replacing it. This has precedent in adjacent domains: Liu et al. [12] encode stroke velocity and curvature as image channels for sketch classification, showing temporal derivatives carry shape information absent from raw ink; Kunhoth et al. [10] use pressure and tilt maps as auxiliary CNN channels for dysgraphia diagnosis, confirming these signals improve accuracy independently of the ink image. For text recognition the approach remains largely unexplored. Stroke order is particularly relevant for Nastaliq,

where spatially overlapping strokes follow a defined production order that disambiguates visually similar ligature classes — information static offline systems cannot represent. To our knowledge, this is the first work to systematically en-

Table 1. OUHD-L statistics collected on Wacom 10.3-inch IoT Paper surface (297×182 mm, 1872×1404 coordinate space, 226 DPI).

Property	**Value**
Total text lines	2,403
Number of writers	311
Unique ligatures	5,829
Avg. line length (chars)	34.93
Avg. line length (words)	9.66
Train / Val / Test (lines)	1,911 / 239 / 253
Train / Val / Test (writers)	191 / 26 / 23
Sampling rate	500 Hz
Pen channels	x, y, pressure, tilt (α, β), height, thickness, eraser/pen, timestamp

code multi-channel stroke dynamics as spatially-aligned feature images for cursive text-line recognition in a low-resource right-to-left script. The availability of channels varies by device: IAMOn-DB records (x, y, t) and pen-up/down events but not pressure or tilt, as it was collected on older Wacom tablets without force or orientation sensors; ADAB similarly provides coordinates and timing only. OUHD-L, collected on the Wacom IoT Paper at 500 Hz, additionally provides pressure, tilt (α, β), pen height, and tip thickness, enabling the full 11-channel encoding evaluated in this work. The encoding principle is script-agnostic: velocity, direction, and curvature channels can be computed from any device that records (x, y, t), enabling partial-channel application to IAMOn-DB or ADAB without pressure or tilt.

3 Online Urdu Handwriting Dataset - Lines

To the best of our knowledge, no publicly available online Urdu text-line dataset exists prior to this work. We address this gap by collecting the first such resource using a Wacom IoT pen device, designed to capture not only the rendered ink but the full spatio-temporal trajectory of each writing event. Table. 1 summarises the dataset statistics.

3.1 Collection Setup

Ground-truth transcriptions were sourced from NUST-UHWR and filtered to ensure balanced ligature coverage and a length of 8–10 words per line, yielding approximately 2,500 candidate prompts. Each writer was asked to copy one

prompt at a time in their natural handwriting style, with no constraints on speed, slant, or ligature formation. Although prompts originate from NUST-UHWR, leakage is minimal for five independent reasons: writers are entirely disjoint; the modality is different (online trajectories vs. scanned offline images); word-level omissions during writing mean refined ground-truth strings rarely match any complete NUST-UHWR sentence; there is no image-level overlap; and the language model decoder is frozen during training on OUHD-L. Critically, the differential gain between the ink-only and full fusion conditions cannot be attributed to text leakage, since both are trained and evaluated on identical splits against the same frozen backbone. Writers occasionally omitted words during transcription; after collection, each ground-truth string was updated to reflect only the words actually written, reducing the usable set to the final 2,403 lines in Table 1. Splits are fully writer-disjoint: the 191 train, 26 validation, and 23 test writers are entirely non-overlapping, ensuring no writer seen during training or validation appears in the test set.

Algorithm 1 Spatial Rasterisation of a Single Feature Channel

Input: Pen-down points $\{(x_i, y_i, f_i)\}$; width range $[w_{\min}, w_{\max}]$
Output: Rasterised canvas C

$C \leftarrow \mathbf{0}^{(\max(y)+1)\times(\max(x)+1)}$ ▷ black canvas
for each consecutive pen-down pair $(i,\ i{+}1)$ **do**
  $\hat{f} \leftarrow \textsc{Normalise}(f_i)$ ▷ to $[0, 1]$
  $\tau \leftarrow w_{\min} + \hat{f}\,(w_{\max} - w_{\min})$ ▷ linear map to $[w_{\min}, w_{\max}]$
  $\textsc{cv2.line}\Big(C,\ (x_i, y_i),\ (x_{i+1}, y_{i+1}),\ \mathit{color}{=}\hat{f},\ \mathit{thickness}{=}\tau\Big)$ ▷ later strokes overwrite
end for
$C \leftarrow 1 - C$ ▷ invert: dark ink on white
return C

3.2 Preprocessing

Each recorded sequence is sorted by timestamp to ensure strict temporal ordering. Eraser events are purged to prevent spurious trajectories from corrupting the rendered images. Stroke segmentation is performed implicitly via the pressure signal: only consecutive point pairs with strictly positive pressure readings are treated as active pen-down segments, eliminating the need for an explicit pen-up/pen-down event parser.

4 Methodology

Our approach is driven by a simple low-resource constraint: the available online Urdu data is too limited to train a dedicated online sequence model from scratch,

yet the offline model of [16] already carries mature, language-aware representations of Urdu built from far larger offline corpora. The design goal is therefore to reuse those offline representations while incorporating the additional discriminative information that online signals provide. We achieve this by converting stroke dynamics into spatially-aligned feature images, channels that exist in the same domain as the model's existing input, so that the pre-trained CNN and Transformer can be retained without architectural change. The pipeline has three stages: (1) stroke rendering, in which the raw online trajectory is converted into a multi-channel feature image; (2) feature image fusion, in which online-derived channels are fused with the ink image using a learned gate; and (3) sequence recognition, in which the pre-trained offline Urdu HTR model decodes the fused representation. Fig. 3 illustrates the full pipeline.

Table 2. Online feature image channels and their physical interpretation.

Channel	Feature	Pixel intensity encodes
I_{ink}	Ink	Constant (binary presence)
$I_{\Delta x}$	Displacement-x	Normalised Δx_i between consecutive points
$I_{\Delta y}$	Displacement-y	Normalised Δy_i between consecutive points
I_{α}	Tilt-x	Stylus tilt along x-axis
I_{β}	Tilt-y	Stylus tilt along y-axis
I_{p}	Pressure	Normalised pen pressure p_i
I_{v}	Velocity	Normalised instantaneous speed
I_{a}	Acceleration	Normalised $\lvert\Delta v/\Delta t\rvert$
I_{sin}	Direction-sin	$\sin(\theta_i)$; periodic-stable vertical component
I_{cos}	Direction-cos	$\cos(\theta_i)$; periodic-stable horizontal component
I_{κ}	Curvature	Rate of directional change per arc length
$I_{\hat{t}}$	Global order	Normalised temporal index $\hat{t}_i$

4.1 Online Feature Image Generation

Given a text-line trajectory consisting of N sampled points $\{(x_i, y_i, p_i, \alpha_i, \beta_i)\}_{i=1}^{N}$, where p_i is pen pressure and (α_i, β_i) are the x- and y-tilt angles, we derive a rich set of temporal and kinematic feature channels and render each as a spatially registered grayscale image. Velocity, acceleration, and curvature are established discriminative features in online HTR [20,5]; pressure and tilt have been validated as auxiliary CNN channels for stylus-based recognition [10]; directional encoding follows the formulation of [12].

Temporal Normalisation. All timestamps are shifted so that the sequence begins at $t = 0$ by subtracting the minimum observed time. Time differences $\Delta t_i = t_i - t_{i-1}$ are computed between consecutive samples.

Kinematic Feature Extraction. Let $(\Delta x_i,\ \Delta y_i) = (x_{i+1} - x_i,\ y_{i+1} - y_i)$ denote the first-order spatial differences between consecutive pen-down points, and let

$\Delta t_i = t_{i+1} - t_i$ be the corresponding time interval. Velocity components are computed as

$$v_i^x = \frac{\Delta x_i}{\Delta t_i}, \qquad v_i^y = \frac{\Delta y_i}{\Delta t_i}, \tag{1}$$

with instantaneous speed defined as $\text{speed}_i = \sqrt{(v_i^x)^2 + (v_i^y)^2}$.

The direction of motion is encoded as

$$\theta_i = \arctan 2(\Delta y_i,\ \Delta x_i), \tag{2}$$

with $\sin\theta_i$ and $\cos\theta_i$ retained separately as periodicity-stable representations of heading. Acceleration components $\Delta v_i^x = v_{i+1}^x - v_i^x$ and $\Delta v_i^y = v_{i+1}^y - v_i^y$ are computed as first differences of the velocity series, yielding a scalar acceleration magnitude

$$a_i = \sqrt{(\Delta v_i^x)^2 + (\Delta v_i^y)^2}. \tag{3}$$

Finally, curvature is defined as the rate of directional change per unit arc length,

$$\kappa_i = \frac{\Delta\theta_i}{\Delta s_i}, \qquad \Delta s_i = \sqrt{(\Delta x_i)^2 + (\Delta y_i)^2}, \tag{4}$$

where near-zero arc lengths are clamped to 10^{-6} to avoid singularities.

Global Temporal Encoding. A normalised global time coordinate $\hat{t}_i = t_i / t_{\max}$ encodes each point's position within the full writing sequence on a $[0, 1]$ scale, preserving the sequential order in which regions of the text were written.

Feature Normalisation. All normalisation is computed on pen-down samples only; pen-up samples (pressure = 0) are set to zero. Pressure is clipped to the $[5, 95]$ percentile range, mapped to $[0, 1]$, and power-transformed with exponent γ to amplify mid-range variation; line width is derived as $w = w_{\min} + s \cdot (w_{\max} - w_{\min})$ with $w_{\min} = 1.5$ and $w_{\max} = 7.0$ pixels. Tilt and kinematic channels are z-score standardised, compressed via $s = (\tanh(\alpha z) + 1)/2$, and remapped to $[0, 1]$. Rendered images are tightly cropped using an intensity threshold of 250.

Spatial Rasterisation. Each normalised feature channel is rendered onto a canvas of size $(\max(y_i)+1) \times (\max(x_i)+1)$ pixels, determined by the bounding box of the pen-down trajectory. Consecutive pen-down points are connected, with pixel intensity and stroke thickness both modulated by the normalised feature value at point i; overlapping segments are resolved by overwriting (Algorithm. 1).

The final input to the recognizer is formed by concatenating all channels in Table. 2 along the depth axis:

$$\mathbf{I} = \left[I_{\text{ink}};\ I_{\Delta x};\ I_{\Delta y};\ I_{\alpha};\ I_{\beta};\ I_{\sin};\ I_{\cos};\ I_{\kappa};\ I_{v};\ I_{a};\ I_{\hat{t}};\ I_{p}\right] \in \mathbb{R}^{H \times W \times 12}. \tag{5}$$

These channels in $\mathbf{I}$(Eq. 5) constrain the feature space towards physically plausible stroke dynamics.

4.2 Model Architecture

We build on the unified CNN-Transformer architecture of [16] as our offline backbone. The full model consists of four sequentially connected components: a fused CNN encoder that ingests the multi-channel image $\mathbf{I}$ and produces a sequence of 256-dimensional feature vectors; a Transformer encoder that contextualises those features; a CTC projection head for sequence-level supervision; and a pre-trained Transformer decoder that generates the final transcription autoregressively.

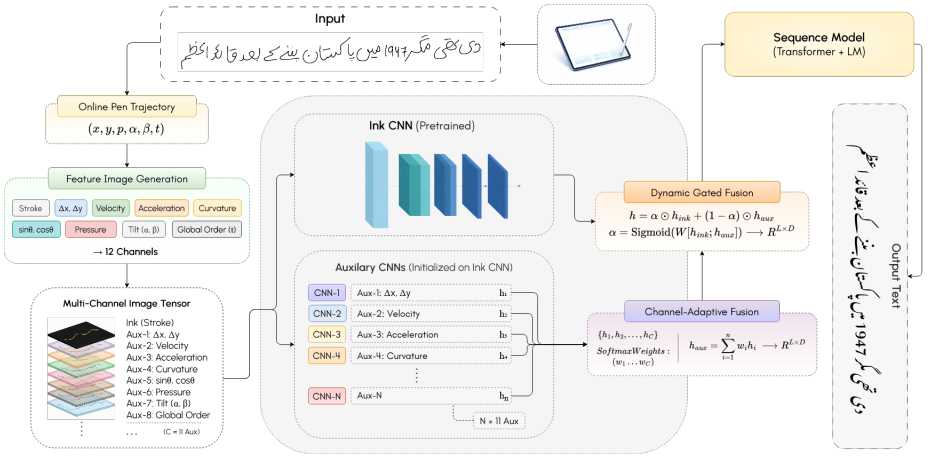

Fig. 3. Overview of the proposed pipeline. The raw pen trajectory $(x, y, p, \alpha, \beta, t)$ is encoded into 12 spatially-aligned feature image channels, each processed by a dedicated CNN branch with shared pre-trained weights. Per-channel embeddings are fused via channel-adaptive and dynamic gated fusion before decoding by the pre-trained Transformer encoder-decoder.

CNN Backbone The convolutional backbone of [16] comprises five progressively deeper blocks of 3×3 convolutions, batch normalisation, GELU activations, max-pooling, and dropout. Given an input of size $H \times W \times 1$, the blocks reduce the spatial resolution to $1 \times (W/4)$ and expand the channel dimension to 256, yielding a feature sequence:

$$\mathbf{F} = \text{CNN}(I) \in \mathbb{R}^{L \times 256}, \quad L = W/4. \tag{6}$$

Fused CNN Encoder To incorporate the $C = 11$ auxiliary channels alongside the ink image, each modality is processed through a dedicated pathway before merging at the feature-sequence level.

Ink branch. The ink channel $I_{\text{ink}} \in \mathbb{R}^{H \times W \times 1}$ is processed by the pre-trained backbone:

$$\mathbf{h}_{\text{orig}} = \text{CNN}^{(\text{pre})}(I_{\text{ink}}) \in \mathbb{R}^{L \times 256}. \tag{7}$$

Auxiliary branch. Each auxiliary channel I_c is processed by an independent copy of the CNN initialised from the same pre-trained weights:

$$\mathbf{f}_c = \mathrm{CNN}^{(c)}(I_c) \in \mathbb{R}^{L\times 256}, \qquad c = 1, \ldots, C. \tag{8}$$

Separate networks per channel allow each modality to specialise its own low-level features while retaining the inductive biases of the pre-trained weights.

Channel-adaptive fusion. The C branch outputs are aggregated via a learned per-feature weighting. A trainable matrix $\mathbf{W} \in \mathbb{R}^{C\times 256}$ is normalised over the channel dimension:

$$\boldsymbol{\alpha} = \mathrm{softmax}(\mathbf{W},\ \mathrm{dim}{=}C) \in \mathbb{R}^{C\times 256}, \tag{9}$$

and the auxiliary representation is formed as:

$$\mathbf{h}_{\mathrm{aux}} = \sum_{c=1}^{C} \alpha_c \odot \mathbf{f}_c \in \mathbb{R}^{L\times 256}. \tag{10}$$

This allows the model to up-weight informative channels such as velocity or curvature and suppress noisier ones in a feature-selective manner.

Dynamic Adaptive Fusion The ink and auxiliary sequences are merged by a dynamic fusion module that computes a content-dependent gate at every sequence position and feature dimension. Both inputs are first independently layer-normalised:

$$\tilde{\mathbf{h}}_{\mathrm{orig}} = \mathrm{LN}(\mathbf{h}_{\mathrm{orig}}), \qquad \tilde{\mathbf{h}}_{\mathrm{aux}} = \mathrm{LN}(\mathbf{h}_{\mathrm{aux}}). \tag{11}$$

The normalised sequences are concatenated and passed through a linear layer with sigmoid activation to produce a per-position gate:

$$\boldsymbol{\alpha} = \sigma\Big(\mathbf{W}_g\big[\tilde{\mathbf{h}}_{\mathrm{orig}} \parallel \tilde{\mathbf{h}}_{\mathrm{aux}}\big] + \mathbf{b}_g\Big) \in (0,1)^{L\times 256}, \tag{12}$$

where $\mathbf{W}_g \in \mathbb{R}^{512\times 256}$. The fused representation is then a convex combination:

$$\mathbf{h} = \boldsymbol{\alpha} \odot \tilde{\mathbf{h}}_{\mathrm{orig}} + (1 - \boldsymbol{\alpha}) \odot \tilde{\mathbf{h}}_{\mathrm{aux}} \in \mathbb{R}^{L\times 256}. \tag{13}$$

Unlike a scalar gate, the per-position formulation allows the model to rely on the ink representation for well-formed strokes and shift towards auxiliary kinematic features at structurally ambiguous regions such as ligature joins and pen-lifting boundaries.

Transformer Encoder, Decoder, and CTC Head The fused sequence $\mathbf{h}$ is passed to the Transformer encoder of [16], producing context-enriched representations $\mathbf{E} \in \mathbb{R}^{L\times 256}$. A linear CTC head maps $\mathbf{E}$ to log-probabilities over the character vocabulary used in training, while the pre-trained Transformer decoder attends to $\mathbf{E}$ and generates tokens autoregressively via teacher forcing during training and beam search at inference. All downstream Transformer components are unchanged from [16], preserving pre-trained language modelling behaviour throughout fine-tuning.

5 Experimental Setup

We report Character Error Rate (CER) and compare against the SOTA baselines. The experiments were carried out in specific settings to ensure reproducibility. Experiments were built upon [16] training and inference settings with any changes described explicitly.

5.1 Baselines

1. Offline backbone, zero-shot: the pre-trained offline recogniser [16] applied directly to rendered ink from our online dataset without any fine-tuning (CER = 10.65%). This baseline quantifies the transferability of the offline feature space to the online domain.
2. Offline backbone, fine-tuned (ink-only): the same model fine-tuned on I_{ink} from our dataset, without any online feature channels (CER = 4.2%). This is the primary baseline against which our method is evaluated.

Table 3. Channel contribution results on the OUHD-L test set (seed 42). Each row is a separately trained model with a single auxiliary channel added to the ink-only baseline (ref. 4.20%). ↓ indicates CER improvement.

Channel	**CER (%)**	Δ
Zero-shot [16]	10.65	–
Ink-only (reference)	4.20	–
$+I_{\cos}$ (dir.-cos)	3.71	−0.49 ↓
$+I_{v}$ (velocity)	3.80	−0.40 ↓
$+I_{\Delta y}$ (disp.-y)	3.86	−0.34 ↓
$+I_{\kappa}$ (curvature)	3.88	−0.32 ↓
$+I_{p}$ (pressure)	3.89	−0.30 ↓
$+I_{\sin}$ (dir.-sin)	3.94	−0.26 ↓
$+I_{a}$ (acceleration)	4.03	−0.17 ↓
$+I_{\Delta x}$ (disp.-x)	4.05	−0.15 ↓
$+I_{\alpha}$ (tilt-x)	4.09	−0.11 ↓
$+I_{\dot{t}}$ (g. order)	4.09	−0.11 ↓
$+I_{\beta}$ (tilt-y)	4.12	−0.08 ↓
Full fusion	**3.67**	**-0.53** ↓

5.2 Implementation Details

All experiments are implemented in PyTorch and run on a single NVIDIA RTX 4090. All experiments use a fixed random seed of 42, applied to Python's `random` module, NumPy, and PyTorch (including all CUDA operations). DataLoader workers are individually seeded via a worker initialisation function to ensure deterministic data ordering across runs. To assess reproducibility, five independent runs were conducted by varying the data-loading seed, augmentation

sequence, per-channel dropout mask, and weight-decay initialisation. A fixed seed of 42 was used within each run to ensure intra-run determinism. Input images are resized to $1{,}600 \times 64$ pixels with aspect ratio preserved, consistent with the backbone pre-training protocol [16]. For inference, the Transformer decoder uses beam search with a beam width of 4.

5.3 Training Protocol

Initialisation. All components (CNN encoder, Transformer encoder and decoder, and CTC head) are initialised from weights pre-trained on offline NUST-UHWR [16]. The Transformer encoder and decoder remain frozen to preserve offline representations as a stable anchor, while the original CNN and auxiliary branches are trainable. This allows adaptation to I_{ink} and integration of online auxiliary signals.

Loss Function. Training uses a joint CTC and cross-entropy objective:

$$\mathcal{L} = \lambda \mathcal{L}_{\text{CTC}} + (1 - \lambda)\mathcal{L}_{\text{CE}}, \quad \lambda = 0.8. \tag{14}$$

Optimiser and Early Stopping. Optimisation uses AdamW [14] with learning rate 3×10^{-4}, batch size 8, and mixed-precision training (`torch.amp`). The best checkpoint (validation loss or CER) is retained, and training stops after 10 epochs without improvement. Inference uses beam search decoding as in [16].

Data Augmentation. I_{ink} undergoes TACo augmentation [6] ($p = 0.9$; vertical tiles $[10, 100]$ px, horizontal $[10, 50]$ px). Auxiliary channels receive no geometric augmentation. During training, per-channel dropout ($p_{\text{ch}} = 0.25$) and group dropout ($p_{\text{grp}} = 0.15$) are applied in a mutually exclusive manner, with group dropout evaluated first.

6 Results and Analysis

We evaluate the proposed system on OUHD-L and report results across three configurations: zero-shot transfer, ink-only fine-tuning, and full multi-channel fusion.

6.1 Main Results

Table 3 presents recognition results on the OUHD-L across all evaluated configurations. Zero-shot transfer of the pre-trained offline backbone [16] to rendered online ink yields 10.65% CER, confirming that the offline feature space generalises partially to the online domain but suffers a substantial penalty from the domain shift between scanned offline images and digitally rendered pen trajectories. Fine-tuning the same model on ink-only images reduces CER to 4.21% (mean over five independent runs; $\pm 0.09\%$), a relative reduction of approximately

60%, demonstrating that domain adaptation alone, without any online-specific information, accounts for the majority of the recoverable error.

The full fusion model, incorporating all 11 auxiliary feature channels, achieves 3.66% CER (mean over five independent runs; ±0.04%), a further relative gain of 13% over the ink-only baseline. While the absolute margin is modest, it is statistically significant across all five runs (paired t-test: $t(4) = 14.45$, $p < 0.001$), and is attributable solely to the online dynamics encoded in the auxiliary channels, as the underlying architecture and training procedure are otherwise identical.

6.2 Channel Contribution Analysis

Table 3 reports the standalone contribution of each channel when added individually to the ink-only baseline. Every channel reduces CER, confirming broad complementarity across kinematic, directional, spatial, and temporal signal types.

Direction-cosine ($I_{\cos}$, −0.49 pp) is the most informative single channel, followed by velocity (I_v, −0.40 pp) and displacement-y ($I_{\Delta y}$, −0.34 pp). This ordering reflects Urdu's discriminative structure: heading angle at ligature junctions is the primary discriminative cue, with pen speed reinforcing it at those same boundaries. The angular channels tilt-x (I_α, −0.11 pp) and tilt-y (I_β, −0.08 pp) contribute less in isolation, consistent with raw stylus angles being less diagnostic than derived directional and kinematic signals.

The full fusion (3.67%, seed 42) outperforms every single-channel configuration, confirming that the auxiliary channels are collectively informative beyond any individual signal. The best single channel ($I_{\cos}$, 3.71%) already recovers the majority of the gain over the ink-only baseline; full fusion adds a further modest margin (−0.04 pp), indicating that the remaining channels resolve complementary error classes that do not fully overlap.

6.3 Discussion

The results confirm the central hypothesis: online stroke dynamics carry discriminative information beyond the static ink image, and encoding them as spatially-aligned feature channels is effective in a low-resource cursive script setting. Every channel reduces CER individually, confirming broad complementarity across kinematic, directional, spatial, and temporal signal types.

Direction-cosine ($I_{\cos}$) leads because heading angle at ligature junctions is the primary discriminative cue in Nastaliq; velocity (I_v) reinforces it by capturing pen speed transitions at those same boundaries. Tilt-x (I_α) and tilt-y (I_β) contribute less in isolation, consistent with raw stylus angles being less distinctive than derived directional and kinematic features. Full fusion achieves a further margin over the best single channel, indicating that complementary error classes are resolved across channel types and do not fully overlap.

6.4 Complementary Investigation: VLM Benchmarks

To test whether model scale alone suffices for low-resource cursive script recognition, we benchmark general-purpose and specialised VLMs on OUHD-L (Ta-

Table 4. VLM benchmark results on the OUHD-L test set. Cleaned CER is computed after discarding output tokens outside the Urdu Unicode block (U+0600-U+06FF), which several models produce when defaulting to Arabic or outputting mixed-script responses.

Model	CER (%)
Ours (full fusion)	**3.66**
Maqsood et al. [16] (ink-only)	4.21
Qwen3-VL-Instruct-4B (QLoRA) [7]	7.28
Qwen3-VL-Instruct-2B (QLoRA) [7]	7.69
Qwen3-VL-Instruct-2B (Base) [4]	27.02
Qwen3-VL-Instruct-4B (Base) [4]	27.95
Camel OCR [22]	36.74
DeepSeek-OCR [25]	38.04
Nanonets-OCR2-3B [15]	46.58
Dots-OCR [11]	68.80

ble 4). All VLMs received identical rendered-ink images (the same input as the ink-only baseline) and a fixed prompt: *"Perform Urdu OCR on this image. Output the text in Nastaliq order."* No per-model prompt engineering was applied, ensuring a fair and reproducible comparison. No VLM matches our pipeline: zero-shot models fail substantially (Qwen3-VL-Instruct-4B Base, 27.95%), and QLoRA fine-tuning closes only part of the gap (Qwen3-VL-Instruct-4B, 7.28%). The binding constraint in low-resource cursive script recognition is not model capacity but the availability of script-specific representations and production-level dynamics absent from general visual training data.

7 Conclusion

We presented the first line-level online Urdu HTR system and dataset, built on a principled response to a fundamental constraint: in low-resource cursive scripts, reusing mature offline representations is more statistically efficient than training from scratch. Encoding pen dynamics as spatially-aligned feature images performs representation alignment, projecting temporal signals into the spatial domain the offline backbone already occupies. Zero-shot transfer, ink-only fine-tuning, and full fusion yield 10.65%, 4.21%, and 3.66% CER respectively, confirming that online dynamics recover production variables absent from the static ink image without any architectural change. A VLM benchmark further confirms that representation alignment outperforms scaling in this data regime. Future work will target learned dynamic-to-image encodings that maximise per-channel information gain, and cross-device robustness where pressure and tilt are noisy or unavailable. A natural extension is to validate the encoding principle on well-resourced online datasets for other scripts: IAMOn-DB (English) and ADAB (Arabic Naskh) provide online trajectories from which velocity, direction, and curvature channels can be computed without pressure or tilt, enabling a partial-channel ablation that tests script-independent generality.

Acknowledgements

This work is supported by *Wacom Co., Ltd. (Japan)*. We thank Takahiro Yamamoto, Toshihiko Horie, and Masamitsu Ito for their support and guidance.

References

1. Ahmed, S.B.: UNHD: Urdu nastaliq handwritten dataset. Kaggle (2023), https://www.kaggle.com/datasets/drsaadbinahmed/unhd-dataset
2. Ahmed, S.B., Naz, S., Swati, S., Razzak, M.I.: Handwritten Urdu Character Recognition Using a One-Dimensional BLSTM Classifier. Neural Computing and Applications **31**(4), 1143–1151 (2019). https://doi.org/10.1007/s00521-017-3146-x
3. Anjum, T., Azhar, A.: A Survey on Urdu Handwritten Text Recognition: State of the Art, Challenges, and Future Directions. Journal of Computing and Artificial Intelligence **3**(1), 33–48 (2025). https://doi.org/10.69591/jcai.3.1.2
4. Bai, S., Cai, Y., et al.: Qwen3-VL Technical Report (2025), https://arxiv.org/abs/2511.21631
5. Carbune, V., Gonnet, P., Deselaers, T., Rowley, H.A., Daryin, A., Calvo, M., Wang, L.L., Keysers, D., Feuz, S., Gervais, P.: Fast Multi-language LSTM-Based Online Handwriting Recognition. International Journal on Document Analysis and Recognition (IJDAR) **23**(2), 89–102 (2020). https://doi.org/10.1007/s10032-020-00350-4
6. Chaudhary, K., Bali, R.: Easter2.0: Improving convolutional models for handwritten text recognition (2022), https://arxiv.org/abs/2205.14879
7. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: QLoRA: Efficient Finetuning of Quantized LLMs (2023), https://arxiv.org/abs/2305.14314
8. Eberhard, D.M., Simons, G.F., Fennig, C.D. (eds.): Ethnologue: Languages of the World. SIL International, Dallas, TX, USA, 26th edn. (2023), https://www.ethnologue.com/
9. Hussain, K., Mughal, N., Ali, I., Hassan, S., Daudpota, S.M.: Urdu News Dataset 1M. Mendeley Data, v3 (2021). https://doi.org/10.17632/834vsxnb99.3
10. Kunhoth, J., Al-Maadeed, S., Kunhoth, S., Bouraoui, S., Microchiche, M.: CNN Feature and Classifier Fusion on Novel Transformed Image Dataset for Dysgraphia Diagnosis in Children. Expert Systems with Applications **231**, 120820 (2023). https://doi.org/10.1016/j.eswa.2023.120740
11. Li, Y., Yang, G., Liu, H., Wang, B., Zhang, C.: dots.ocr: Multilingual Document Layout Parsing in a Single Vision-Language Model. arXiv:2512.02498 (2025). https://doi.org/10.48550/arXiv.2512.02498
12. Liu, M., Jin, L., Xie, Z.: PS-LSTM: Capturing Essential Sequential Online Information with Path Signature and LSTM for Writer Identification. In: 14th IAPR International Conference on Document Analysis and Recognition (ICDAR 2017). vol. 1, pp. 664–669. IEEE, Kyoto, Japan (2017). https://doi.org/10.1109/ICDAR.2017.114
13. Lodh, A., et al.: A Transformer Based Handwriting Recognition System Jointly Using Online and Offline Features. In: 8th Asian Conference on Pattern Recognition (ACPR 2025). LNCS, vol. 16174, pp. 243–257. Springer (2026). https://doi.org/10.48550/arXiv.2506.20255
14. Loshchilov, I., Hutter, F.: Decoupled Weight Decay Regularization. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=Bkg6RiCqY7

15. Mandal, S., Talewar, A., Thakuria, S., Ahuja, P., Juvatkar, P.: Nanonets-OCR2: A model for transforming documents into structured markdown with intelligent content recognition and semantic tagging (2025), https://huggingface.co/nanonets/Nanonets-OCR2-3B
16. Maqsood, A., Riaz, N., Ul-Hasan, A., Shafait, F.: A Unified Architecture for Urdu Printed and Handwritten Text Recognition. In: Document Analysis and Recognition - ICDAR 2023. pp. 116–130. Springer Nature Switzerland (2023)
17. Missen, M.S., Manzoor, A., Husnain, M.: Urdu Handwritten Text Dataset. Zenodo (2023). https://doi.org/10.5281/zenodo.7711168
18. Mitrevski, B., Rak, A., Schnitzler, J., Li, C., Maksai, A., Berent, J., Musat, C.: InkSight: Offline-to-Online Handwriting Conversion by Teaching Vision-Language Models to Read and Write (2025), https://arxiv.org/abs/2402.05804
19. Naeem, M.F., ul Sehr Zia, N., Awan, A.A., Shafait, F., Ul-Hasan, A.: Impact of ligature coverage on training practical Urdu OCR systems. In: 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 131–136 (2017). https://doi.org/10.1109/ICDAR.2017.30
20. Plamondon, R., Srihari, S.: Online and off-line handwriting recognition: a comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(1), 63–84 (2000). https://doi.org/10.1109/34.824821
21. Riaz, N., Arbab, H., Maqsood, A., Nasir, K.B., Ul-Hasan, A., Shafait, F.: Conv-Transformer Architecture for Unconstrained Off-Line Urdu Handwriting Recognition. International Journal on Document Analysis and Recognition (IJDAR) **25**(4), 373–384 (2022). https://doi.org/10.1007/s10032-022-00416-5
22. Sakthi, P.: Camel-Doc-OCR-080125 (Revision c212648). Hugging Face model card (2025). https://doi.org/10.57967/hf/6489
23. Shafait, F., Hasan, A., Keysers, D., Breuel, T.: Layout analysis of Urdu document images. In: IEEE International Multitopic Conference (INMIC). pp. 293 – 298 (01 2007). https://doi.org/10.1109/INMIC.2006.358180
24. Ul-Hasan, A., Ahmed, S.B., Rashid, F., Shafait, F., Breuel, T.M.: Offline Printed Urdu Nastaleeq Script Recognition with Bidirectional LSTM Networks. In: 2013 12th International Conference on Document Analysis and Recognition. pp. 1061–1065 (2013). https://doi.org/10.1109/ICDAR.2013.212
25. Wei, H., Sun, Y., Li, Y.: DeepSeek-OCR: Contexts Optical Compression (2025), https://arxiv.org/abs/2510.18234
26. Xu, Z., Chen, Z., Wu, Y., Li, H., Lv, W., Jin, L., Wang, Q.: A Multi-Scale Bimodal Fusion Network for Robust and Accurate Online Handwriting Recognition. In: ICASSP. pp. 6460–6464 (2024). https://doi.org/10.1109/ICASSP48485.2024.10446390
27. Zia, N., Naeem, M., Raza, S., Khan, M., Ul-Hasan, A., Shafait, F.: A Convolutional Recursive Deep Architecture for Unconstrained Urdu Handwriting Recognition. Neural Computing and Applications **34**(2), 1635–1648 (2022). https://doi.org/10.1007/s00521-021-06498-2