**AI3011 | Spring 2026**

Machine Learning & Pattern Recognition

# Deciphering Visually Similar Indian Scripts.


**By**

**Anil Aleti · Chandrakant Singh · Yukteswar Mantha**

Student ID: U20240150, U20240157, U20240189

# Overview

# Background & Problem



## Why script identification is the bottleneck

Indian OCR runs as a 3 stage pipeline. A wrong script prediction routes a word to the wrong recogniser due to which the error propagates through every downstream stage, which makes **Stage 2 the single highest leverage step** in the system.

STAGE 1
**Scene Text Detection**
Locate and crop all text regions in the scene.

→

STAGE 2 · THIS WORK
**Script Identification**
Determine the language of each cropped word.

→

STAGE 3
**Word Recognition**
Read text with the correct script-specific recogniser.

→

OUTPUT
**End-to-End OCR**
Multilingual scene text → machine-readable text.

## The Oracle Experiment: BSTD, De et al. 2025

**36%** CURRENT PIPELINE → **71%** WITH ORACLE SCRIPT

**+ 35 percentage points** the largest single step gain available in any Indian OCR system, and ~30 pp above Google OCR.

## Why India makes this hard

- India has **22 official languages** and **13+ distinct scripts**.
- Public signage freely mixes them because one board often carries Hindi, English, and a regional script simultaneously.
- BSTD's best published model (ViT, ImageNet-21k) reached only **80.5 %** on 12-script ID and flagged Assamese/Bengali and Hindi/Marathi as systematic failures.

# Literature Review

## 1. Dataset

| DATASET | IMAGES | WORDS | LANGS | DET | SI | REC |
|---|---|---|---|---|---|---|
| **ICDAR 2015** | 1,500 | 6,545 | 1 | ✓ | ✗ | ✓ |
| **IIIT-TL-STR** | 440 | 3,035 | 2 | ✗ | ✗ | ✓ |
| **IIIT-ILST** | — | 3,168 | 3 | ✗ | ✗ | ✓ |
| **MLT-17 / MLT-19** | 18–20K | 96–191K | 9–10 | ✓ | ✓ | ✓ |
| **IndicSTR12 : Lunia et al. 2023** | — | 27,000 | 12 | ✗ | ✗ | ✓ |
| **BSTD — De et al. 2025** | 6,582 | 1,06,478 | 12 | ✓ | ✓ | ✓ |

## 2. Model contributions

| WORK | APPROACH | KEY LIMITATION FOR OUR PROBLEM |
|---|---|---|
| **BSTD ViT baseline: De et al. 2025** | ViT-B/16 (ImageNet-21k), fine-tuned | 80.5 % but Asm→Ben 39 % err, H→M 20 % error, not investigated |
| **LaSA-Net: Vijayan et al. 2024** | Language-aware spatial attention | Assumes script is known and skips identification entirely |
| **CRNN / AlexNet baselines** | CNN-RNN hybrid; ImageNet CNN | 67–76 % on 3-way; no 12-class study |
| **IIIT-ILST: Mathew et al. 2017** | Traditional OCR on 3-script dataset | Only Dev, Tel, Mal; pre-deep-learning era |

## 3. What is open

No prior work systematically asks **why** a model fails on specific confusion pairs.

Is it the architecture? The pretraining? Or something about the scripts themselves that no architecture can overcome?

THE GAP WE FILL

We treat confusion pairs not as noise to be minimised, but as *signals to be understood:* diagnostic instruments that reveal where each architecture's inductive bias breaks down.

→ Our foundation is the BSTD dataset. The next slide unpacks what it contains and what we actually used.

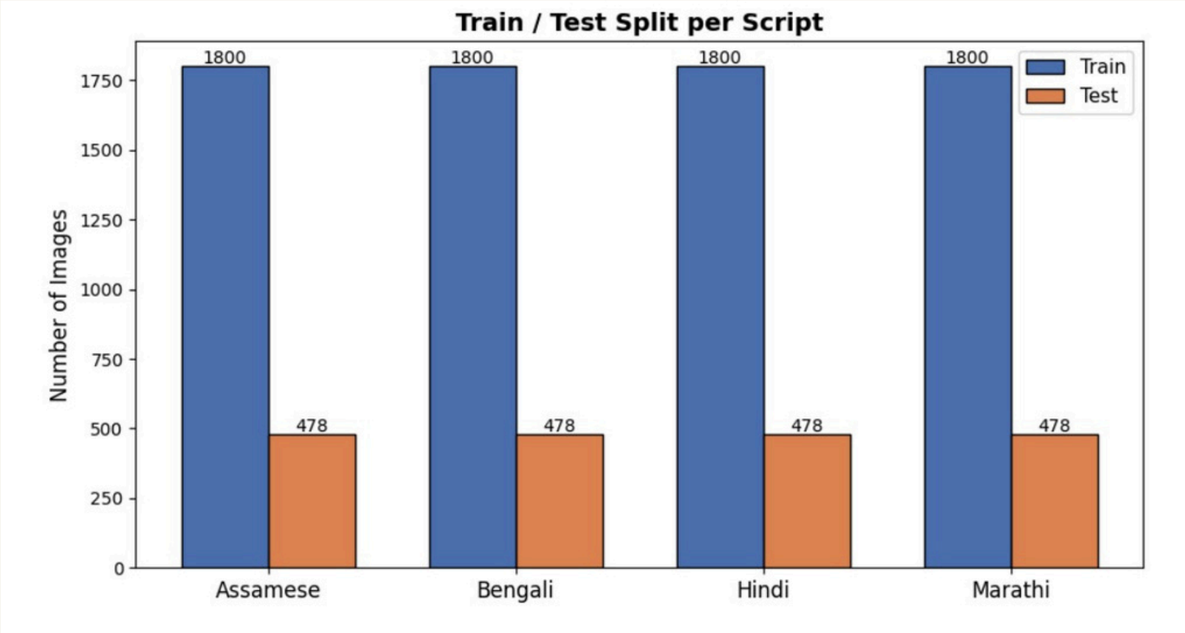
# BSTD Dataset

github.com/Bhashini-IITJ/BharatSceneTextDataset

De et al. 2025 · IIT Jodhpur / Bhashini-MeitY

## What BSTD contains

- **6,582 real world scene images** of Indian streets, signboards, railway stations, temples and hoardings.
- **1,06,478 word-level annotations** across 11 Indian languages + English, polygon-level boxes, manually annotated over 10 months.
- Supports 4 tasks: Detection · Script ID · Cropped Word Recognition · End-to-End OCR.




## The subset we used for Script Identification only

| TRAIN · PER LANGUAGE | TRAIN · 12-CLASS TOTAL |
|---|---|
| 1,800 | 21,600 |
| **TEST · PER LANGUAGE** | **TEST · 12-CLASS TOTAL** |
| 478 | 5,736 |

The subset is **perfectly balanced:** Macro F1 equals accuracy, and per-class recall reads model bias at face value.

## Preprocessing applied

| | |
|---|---|
| PHASE 1 | 224 × 224 · ImageNet norm · Flip · Rotation · ColorJitter |
| PHASE 2 | 256 × 256 · RandomPerspective · RandomAffine · RandomErasing · Label smoothing ε = 0.1 |
| CLIP | CLIP-specific mean/std · Same augmentation suite |

# Why It Is Hard

# Phase 1 - Benchmarking

| ARCHITECTURE | PARAMS | 4-CLASS ACC. | F1 | KEY BEHAVIOUR |
|---|---|---|---|---|
| **EfficientNet-B0** | 5.3 M | **61.0** | 0.61 | Worst Marathi recall (0.51); compound scaling backfires on stroke level task. |
| **ViT-B/16** · ImageNet-1k | 86 M | **~62.0** | 0.64 | Peaks at epoch 3, then collapses: catastrophic forgetting with insufficient data. |
| **MobileNet-V3** | 2.5 M | **65.7** | 0.65 | Stable; SE blocks provide marginal gain; lightweight advantage noted. |
| **ResNet-50** | 25.6 M | **64.0** | 0.63 | Overfits after epoch 30: 25 M params, only 7,200 images. |
| **ResNet-18** | 11.2 M | **67.68** | 0.67 | Best overall; residual connections + right capacity/data ratio. |

**Hindi**

**71.0 %**

**Assamese** WORST

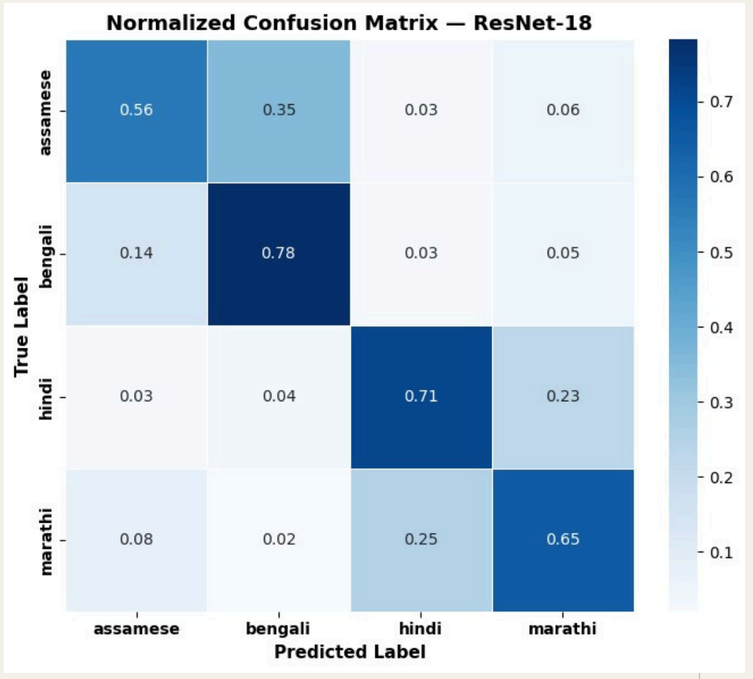
**56.0 %**

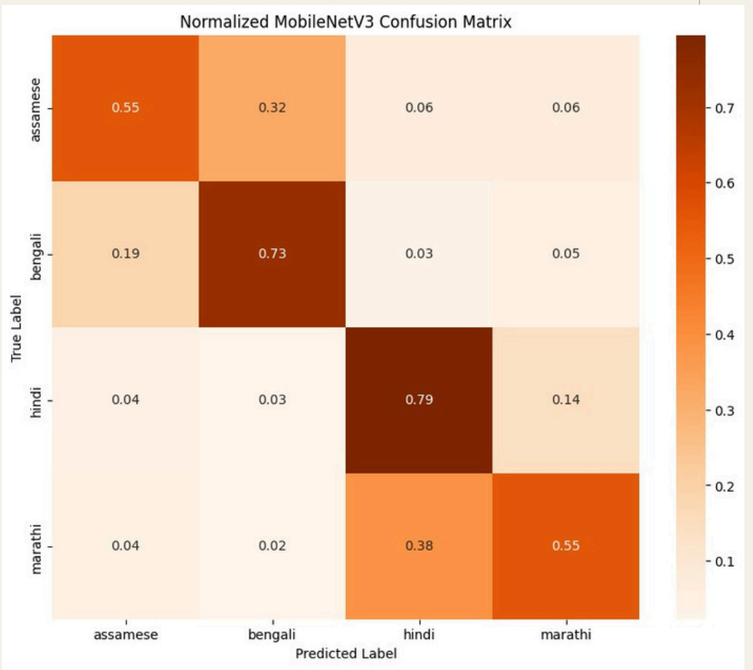
**Marathi**

**65.0 %**

**Bengali**

**78.0 %**



# Phase 2 - Training

01 · RESOLUTION

### 224 → 256 + crop

Upsizing before random crop preserves fine stroke detail that 224 px compresses away.

02 · AUGMENTATION

### Perspective + Affine + Erasing

Simulates real scene text: oblique angles, partial occlusion, varied capture distances.

03 · LOSS

### Label smoothing ε = 0.1

Hindi ↔️ Marathi are visually indistinguishable. Label smoothing acknowledges irreducible ambiguity at the loss level.

04 · HEAD

### MLP classifier head

Dropout → Linear(512→256) → BN → ReLU → Dropout → Linear(256→4). Forces diverse features.

05 · OPTIMIZER

### AdamW + weight decay

Decouples L2 from adaptive LR. Adam's weight decay interacts incorrectly with adaptive rates; AdamW fixes it.

06 · SCHEDULE

### Cosine LR + 5-ep warmup

Warmup prevents early updates from destroying pretrained features. Converged at epoch 29/50.

07 · STOPPING

### Early stopping patience = 10

Stops when validation accuracy plateaus for 10 epochs. The widening train - validation gap observed in Phase 1 would have driven test accuracy lower without it. The same process is applied identically to MobileNet-V3.

RESULT · 4-CLASS

**+ 6.33 pp**

67.68 % → 74.01 % · ResNet-18

RESNET-18 · PHASE 1

**67.68 %**

4-class baseline

RESNET-18 · PHASE 2

**74.01 %**

+ 6.33 pp vs. baseline

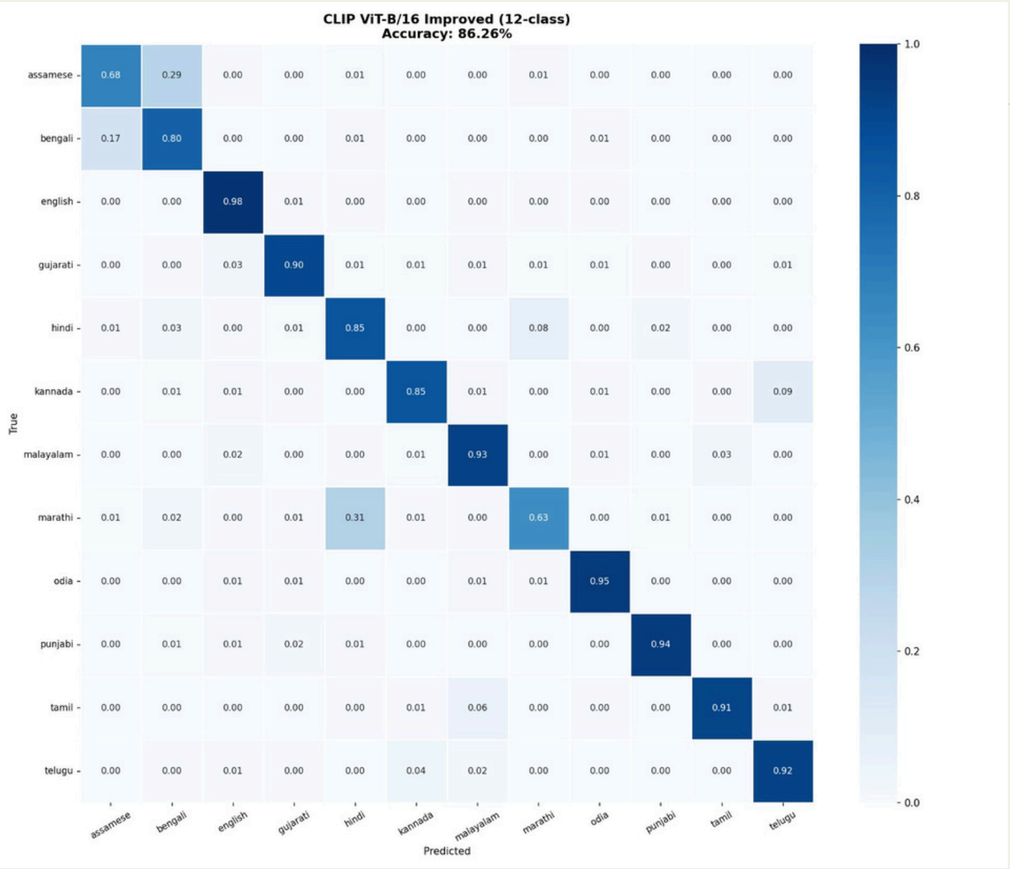
MOBILENET · PHASE 2

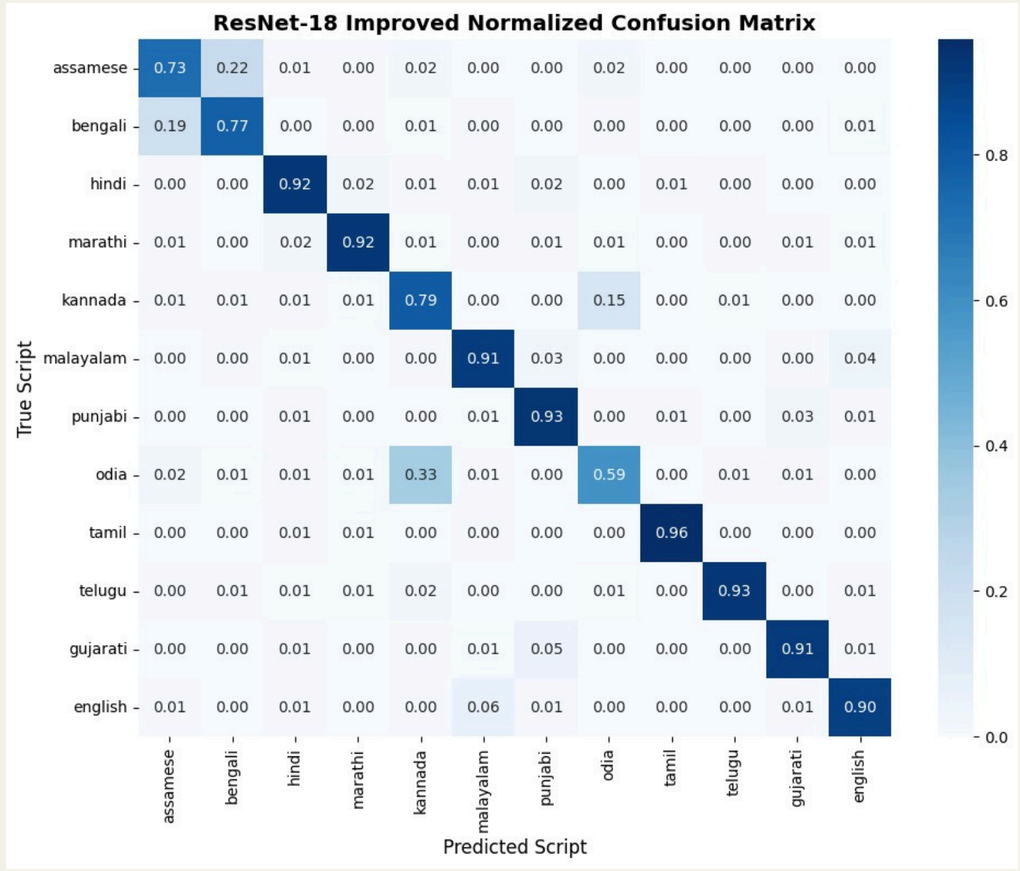
**68.57 %**

11 × fewer params than ResNet

# 12 Languages & CLIP

| MODEL | PARAMS | 12-CLS ACC. | F1 | STRONGEST | WEAKEST |
|---|---|---|---|---|---|
| **MobileNet-V3 (improved)** | 2.5 M | **82.30** | 0.82 | Tamil 0.93 | Odia 0.58 |
| **ResNet-18 (improved)** | 11.2 M | **85.41** | 0.85 | Tamil 0.96 | Odia 0.59 |
| **CLIP ViT-B/16 (fine-tuned)** | 86 M | **86.50** | 0.87 | Odia 0.95 | Asm 0.68 |




## Why CLIP succeeds where ImageNet-ViT failed

In Phase 1, ViT fine tuned from ImageNet-1k collapsed to ~62 % on 4-class. Here, CLIP's ViT reaches **86.5 % on the harder 12-class task**.

> CLIP was trained on **400 M image text pairs** from the web including multilingual Indian signs, social media posts, and documents in Indian scripts. Its visual encoder developed script-relevant features *as a by product* of aligning images with text. ImageNet cannot provide this.

# The Core Tradeoff

The same architecture that solves one pair actively breaks another. The most important result of Phase 2 is not the accuracy number, it is this pattern.

| CONFUSION PAIR | BSTD VIT<br>80.5 % overall | RESNET-18<br>85.4 % overall | MOBILENET<br>82.3 % overall | CLIP<br>86.5 % overall |
|---|---|---|---|---|
| **Assamese ↔ Bengali**<br>ASM RECALL · ASM→BEN ERR | 56 % / 39 % err | 73 % / 22 % err | 63 % / 32 % err | 77 % / 22 % err |
| **Hindi ↔ Marathi**<br>BOTH RECALLS · H→M · M→H | 64 % 70 % 20 % 20 % | 92 % 92 %<br>2 % 2 %<br>SOLVED | 88 % 90 %<br>5 % 0 %<br>NEAR-SOLVED | 81 % 72 %<br>12 % 22 %<br>REGRESSED |
| **Odia ↔ Kannada**<br>ODIA RECALL · ODIA→KAN ERR | 93 % / 1 % err | 59 % / 33 % err<br>BROKEN | 58 % / 32 % err<br>BROKEN | 96 % / 1 % err<br>SOLVED |

## Assamese-Bengali: Never Solved

Even the best model (CLIP, 77 %) misclassifies 22 % of Assamese. The discriminating visual features are too rare and subtle. A true ceiling problem.

## Hindi-Marathi: solved by CNNs, broken by CLIP

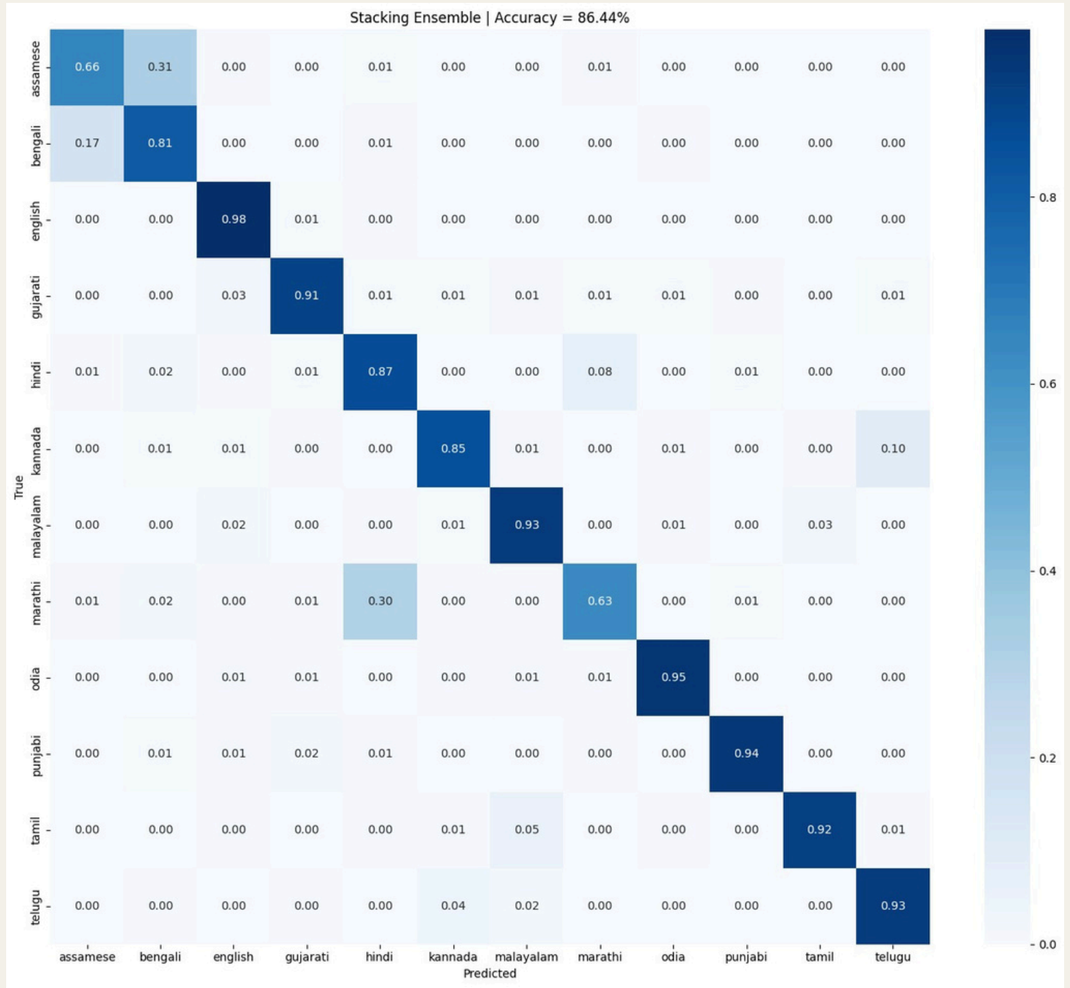
CLIP's pretraining taught it that "Devanagari text" is one semantic concept. It cannot cleanly separate Hindi from Marathi — it never had a signal that distinguished them linguistically. ResNet has no such prior; it found purely visual stroke differences.

## Odia-Kannada: solved by CLIP, broken by CNNs

Odia's identifying feature is a global arc spanning the full character. CNN 3 × 3 filters cannot detect a structure that wide in one pass. CLIP's global self-attention patches the whole character simultaneously.

# Phase 3 - Ensemble Techniques

| Step | What we did | Accuracy | Key numbers | Insight | Limitation |
|---|---|---|---|---|---|
| **E1** Weighted average | Single weight w swept from 0→1. Final = w×ResNet + (1−w)×CLIP. Also fixed class-order mismatch between models. | **86.24%** w=0 best (CLIP only) | Odia: 95% · Hindi: 85% Marathi: 63% At w=0.5 → collapses to 60% At w=1.0 → collapses to 10% | Adding any ResNet weight hurts. Odia recall drops as ResNet weight rises; Marathi barely improves. The tradeoff is structural — one weight cannot serve all classes. | No improvement A single global weight cannot simultaneously fix Marathi and preserve Odia. |
| **E2** Class-aware weights | Different blend per class column. Hindi & Marathi cols → 80% ResNet + 20% CLIP. Odia, Kannada, Assamese, Bengali cols → 20% ResNet + 80% CLIP. All other cols → 50/50. Re-normalize after. | **82.67%** ↓ worse than CLIP alone | Hindi: 91% (improved) Marathi: 16% (collapsed) Odia: 96% (preserved) Assamese: 67% | Boosting ResNet on Hindi column makes Hindi win over Marathi for almost every Devanagari sample. Marathi recall collapses from 63% to 16%. | Regression −3.57% Columns are interdependent. Scaling one column distorts all argmax decisions after re-normalization. |
| **E3** Stacking (meta-learner) | Concatenate ResNet probs (12-dim) + CLIP probs (12-dim) = 24-dim vector. Train Logistic Regression on train set. LR predicts final class at test time. | **86.44%** Best ensemble overall | Hindi: 87% · Marathi: 63% Odia: 95% Macro F1: 0.86 +0.20% over CLIP alone | LR can learn non-linear fusion. Best result technically. But learned to mostly trust CLIP — Marathi recall unchanged at 63%. | Within noise +0.20% ≈ 11 samples out of 5,736. Hard H/M samples are ambiguous to both models — no clean signal for LR to learn. |



# Phase 3 - Ensemble Techniques

| | | | | | |
|---|---|---|---|---|---|
| **E4**<br>Pair-aware routing | If CLIP's top pred is Hindi AND P(Marathi) > HM_thresh → route to ResNet for H/M decision. Same for Assamese/Bengali pair. Grid search over both thresholds. HM_routed & AB_routed count how many samples were handed off. | **86.21%**<br>Best threshold: HM=0.10, AB=0.20 | HM routed: 7 / 5,736<br>AB routed: 36 / 5,736<br>Marathi: 63% (unchanged)<br>Hindi: 85% (unchanged) | Routing on uncertainty only works if the model is uncertain when it's wrong. CLIP is confidently wrong on H/M — P(Marathi) stays low even when it misclassifies Marathi, so the routing condition almost never fires. | 7 routes fired<br>CLIP's H/M errors are high-confidence errors, not low-confidence ones. Uncertainty-based routing cannot catch them. |
| **E5**<br>HM specialist override | Whenever CLIP predicts Hindi or Marathi, check ResNet. If ResNet disagrees (predicts the opposite one) with confidence > RN_THRESHOLD → override CLIP's decision. Swept RN_THRESHOLD from 0.50 to 0.95. | **86.24%**<br>At threshold=0.90 (optimum) | Overrides fired at 0.50: 16<br>Overrides fired at 0.90: 0<br>Lower thresholds → accuracy drops<br>Marathi: 63% (unchanged) | At the optimal threshold, zero overrides fire. When they do fire (lower thresholds), they hurt — ResNet's overrides are also wrong on the hard samples. Both models fail on identical samples. | 0 overrides at optimum<br>No reliable specialist signal exists. Hard H/M samples are genuinely ambiguous to both architectures simultaneously. |

## VERDICT OF PHASE 3

Every ensemble converges to approximately *CLIP-alone performance* (86.0–86.5 %). When the HM specialist override fires zero times at its optimal threshold, CLIP's predictions are already better calibrated than any ResNet alternative. The ensemble reaches its logically correct conclusion: **trust CLIP on everything.**

# Position & Ceiling

## Pipeline position · downstream impact

| SYSTEM | AVG WRR | CONTEXT |
|---|---|---|
| **Tesseract** | ~ 4 % | Fails on Indian scripts |
| **GPT-4 Vision** | ~ 15 % | Not optimised for Indian text |
| **IndicPhotoOCR · current** | ~ 44 % | Full pipeline today |
| **Google OCR** | ~ 54 % | Commercial baseline |
| **IndicPhotoOCR + Oracle SI** | ~ 70 %+ | 30 pp above Google OCR — the gap better SI closes |

## Deployability by model

**MobileNet-V3**
2.5 M params
Edge devices, mobile OCR: real time CPU inference.

**ResNet-18**
11.2 M params
Server API, cloud OCR pipeline: balanced accuracy / speed.

**CLIP ViT-B/16**
86 M params
High-accuracy offline batch: requires GPU server.

**ResNet + CLIP**
ensemble
Not recommended: adds latency for no reliable gain.

## What it would take to go beyond 86 %

Each path moves the problem from image classification to multimodal language understanding: the correct framing for the Hindi/Marathi sub-problem.

**A JOINT ARCHITECTURE THAT UNIFIES LOCAL STROKE ANALYSIS WITH GLOBAL SPATIAL REASONING IS WHAT THE PROBLEM TRULY DEMANDS.**

# Thank you!