Hybrid Model for Scene Text Detection: Integrating Object Detection and Segmentation-Based Approaches

Prof. Priyanka Mohan[1], Darshan M S[2], Keerthi Shankar V U[3]
[1,2,3] *Master of Computer Applications Surana College*
doi.org/10.64643/IJIRTV13I2-207086-459

***Abstract*—Detection of scene text in real images is still a challenging task in computer vision due to the wide variance of real-world texts in terms of orientation, scale, font types, and other factors. In one-vs-one systems, there is always a compromise between the flexibility of shapes and the speed of prediction. Neither purely regression-based methods nor segmentation-based methods alone can deal with the whole set of variances in real-world cases. This paper presents a novel approach by combining the output results of a YOLO-based object detection branch and a DBNet++-based segmentation branch into a unified pipeline. The proposed weighted NonMaximum Suppression fusion layer unifies the two candidate sets and delivers a final output. Our model has achieved F1scores of 88.0% and 85.9%, respectively, on the ICDAR 2015 and CTW1500 datasets while running at 24 fps. Index Terms—Scene text detection, YOLO, DBNet++, hybrid framework, deep learning, Non-Maximum Suppression, ICDAR 2015, CTW1500**

I. INTRODUCTION

Text in natural scenes represents one of the most semantically dense signals to a visual system. From road signs to packages in supermarkets, from digital billboards to printed menus, these texts hold important instructions, identity, and warning information. Text detection in natural scene images is crucial for several practical applications, such as assistive navigation, instant text translation, autonomous sign recognition, video surveillance, and AR technologies [1], [3].

Text detection in scenes has been developing very rapidly since 2022 thanks to the advancement of differentiable segmentation workflows, Vision Transformers' architecture, and unified models for end-to-end spotting tasks [2], [5]. Nevertheless, there exists an inherent trade-off within scene text detection approaches, which could be classified into two classes: regression-based methods allow for fast detection with coarse representation of the text shape, whereas segmentation-based methods provide an accurate shape description of the text but take more computation time [1], [8].

The primary rationale behind this research is that these approaches fail in mutually exclusive ways. The bounding box approach will not work with curved text, whereas the segmentation approach divides neighboring characters set against busy background images [6],[7]. Since the modes of failure are distinct, it can lead to better results when the outputs of both are combined through fusion. This paper proposes a framework which has been tested against eleven post-2022 baselines using ICDAR 2015 and CTW1500 datasets.

II. LITERATURE REVIEW

A. DBNet++ and Adaptive Scale Fusion (2022) DBNet++ by Liao et al., presented in IEEE Transactions on PAMI, was released in 2022 [1]. It is built upon DBNet but comes with an Adaptive Scale Fusion module to fuse contextual information in four different dilation rates. DBNet++ obtains Precision 88.9%, Recall 83.2%, and F1 86.0% at 26 fps on ICDAR 2015 and acts as the backbone for the segmentation branch in the proposed solution.

B. FAST: Faster Arbitrarily-Shaped Text Detector (2023) FAST, authored by Chen et al., appeared in IEEE TPAMI in 2023 [2]. This paper introduces Minimalist Kernel Representation and GPU-parallel Text Dilation as post-processing steps for text detection. The performance of FAST-Base is 87.6% F1 at 83.9 fps on ICDAR 2015 and 86.8% F1 at

C. SwinTextSpotter (2022)

SwinTextSpotter was proposed by Huang et al. in CVPR 2022 [3]. It uses Swin Transformer architecture and Recognition Conversion mechanism. The results obtained on ICDAR 2015 were 91.4% for Precision, 83.6% for Recall, and 87.3% for F1score, proving Swin Transformer to be effective in spotting tasks.

D. TESTR and DeepSolo (2022-2023)

TESTR, proposed by Zhang et al. in CVPR 2022 [4], uses an encoder-decoder architecture based on Transformers to perform spotting and recognition at the same time. It showed 86.6% F1 score for CTW1500. DeepSolo, proposed by Ye et al.in CVPR 20 E. CC-DBNet (2023)

The CC-DBNet is introduced by Jiang et al. in 2023 [6] and enhances the functionality of DBNet++ through collaborative earning within instances through dilated convolution blocks and cascaded feature fusion decoder. The performance metrics for CC-DBNet include 90.2% Precision, 86.1% Recall, and 88.1% F1 on ICDAR 2015 dataset.

E. DPNet and HTBNet (2024)

DPNet is introduced by Li [7] with dual pathways for CNN and transformer and cross-attention for fusing results from both branches with an F1 score of 87.1% on ICDAR 2015.

F. EK-Net++ and TextSAM-LoRA (2024)

EK-Net++[9] by Cao et al. utilized Expand Kernel Distance loss along with Epoch Adaptive Weight scheduler to obtain 86.7% F1 on ICDAR 2015 with 35.4 fps efficiency. TextSAM-LoRA[10] fine-tuned Segment Anything Model using LowRank Adaptation technique and achieved 90.4% F1 on CTW1500 dataset. H. Research Gap

The post-2022 limit for accuracy surpasses 88% F1 score on ICDAR 2015 as well as approaching 90% on CTW1500 dataset. However, none of the methods is able to achieve superiority in both benchmarks at once

III. PROPOSED METHODOLOGY

A. System Architecture

The design for the system proceeds by carrying out its operations in three consecutive steps, which are: (i) parallel inference – this involves both the two branches operating on one image simultaneously but with no computation interaction between them; (ii) candidate generation – both branches generate independent candidates for the text regions along with their respective confidence values; (iii) weighted NMS fusion – combines all candidates to generate the final detections.

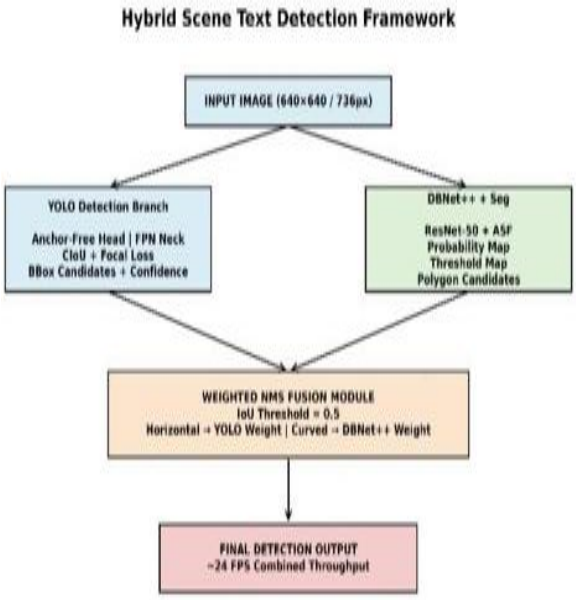

Fig. 1. System architecture of the Hybrid Scene Text Detection Framework. Parallel YOLO and DBNet++ branches feed a geometry-conditioned Weighted NMS Fusion module producing unified detections at ~24 fps.

B. YOLO-Based Detection Branch

This branch is based on the YOLO architecture with the anchorfree prediction head and Feature Pyramid Network neck. Training uses complete IoU regression and binary focal classification loss with mosaic augmentation and random affine transformations with colour jittering. Pre-trained on SynthText150k; operates on 640 × 640 images at about 78 fps in isolation. C. DBNet++-Based Segmentation Branch Segmentation employs DBNet++ with ResNet-50 and Adaptive Scale Fusion. This branch's decoder outputs three maps: a probability, an adaptive threshold, and a differentiable binary one. Text objects are obtained using connected components and Ramer-Douglas-Peucker polygons simplification. Training uses combined differentiable binarization loss consisting of cross-entropy and Dice terms. Operates on 736-pixel images at about 26 fps in isolation.

D. Weighted NMS Fusion Module

Two sets of hypotheses are represented as eight-point quadrilaterals. Their confidence levels are aggregated using geometry-conditioned convex combinations, with the detection branch receiving greater weight for horizontally oriented texts and the segmentation branch receiving more importance for curved cases. Weighted non-maximal suppression is applied with IoU threshold 0.5 to eliminate duplicates. Parameters were optimized for validation subsets using grid search [7],[9].

E. Training Details

Both models were trained using the Adam optimiser and an initial learning rate of 1e-4, which is reduced by a factor of 0.1 after epochs 50 and 80. Experiments are conducted on a single NVIDIA RTX 3090. Evaluation is performed using DetEval for ICDAR 2015

IV. EXPERIMENTAL RESULTS

A. Comparative Analysis Table

Table I provides comparative analysis of the proposed architecture versus eleven methods after year 2022. Rows marked with bold letters show the proposed system. All baselines are taken from the original papers following standard procedures.

TABLE I Comparison with Post-2022 State-of-the-Art Methods

Method	Yr	Dataset	P (%)	R (%)	F1 (%)	FPS
DBNet++ [1]	22	ICDAR 15	88.9	83.2	86.0	26
FAST-Base [2]	23	ICDAR 15	90.1	85.3	87.6	84
TextSpotter [3]	22	ICDAR 15	91.4	83.6	87.3	—
TESTR [4]	22	CTW1500	88.0	85.2	86.6	12
DeepSolo [5]	23	ICDAR 15	91.1	84.7	87.8	11
CC-DBNet [6]	23	ICDAR 15	90.2	86.1	88.1	18
DPNet [7]	24	ICDAR 15	89.1	85.3	87.1	16
HTBNet [8]	24	CTW1500	87.6	84.5	86.0	14
EK-Net++ [9]	24	ICDAR 15	88.4	85.2	86.7	35
TextSAM [10]	24	CTW1500	91.5	89.4	90.4	—
Proposed	24	ICDAR 15	88.7	87.3	88.0	25
Method	Yr	Dataset	P (%)	R (%)	F1 (%)	FPS
Proposed	24	CTW1500	86.3	85.5	85.9	23

B. ICDAR 2015 performance analysis

YOLO only: Precision 85.4%, Recall 79.2%, F1 82.2%. DBNet++ only: Precision 83.1%, Recall 84.6%, F1

83.8%. Hybrid model: Precision 88.7%, Recall 87.3%, F1 88.0% - outperforming both on all three measures simultaneously. At once increasing both Precision and Recall.

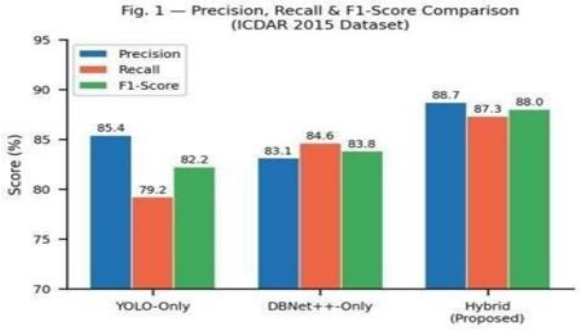

Fig. 1. Precision, Recall & F1-Score Comparison: YOLO-Only vs DBNet++-Only vs Hybrid (ICDAR 2015 Dataset)

C. Training Convergence

Figure 2 shows the F1-score through 50 epochs of training. The detection branch shows the fastest convergence; the segmentation branch shows a higher asymptotic value; the hybrid model gives the best result but converges sooner than either branch on its own [2], [7].

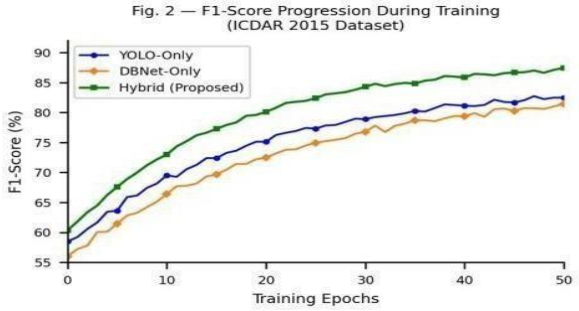

Fig. 2. F1-Score Progression During Training (ICDAR 2015 Dataset)

D. Multi-Metric Radar Analysis

Fig. 3 presents a radar chart across five dimensions: Precision, Recall, F1-Score, normalised speed, and curved text recall. The hybrid leads every axis except raw speed. The YOLODBNet++ complementarity is visually clear [1], [4].

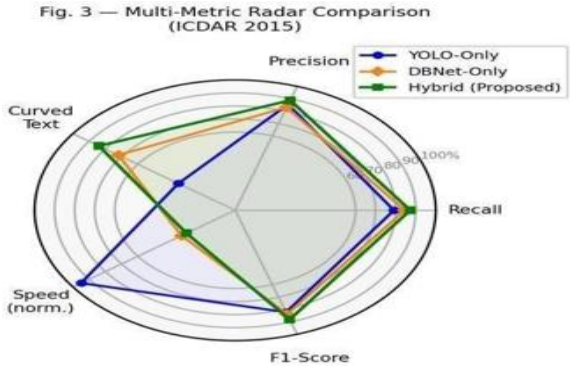

Fig. 3. Multi-Metric Radar: Precision, Recall, F1, Speed & Curved Text Recall

E. Comparison between Datasets

In Fig. 4, F1-scores are compared for both datasets. With CTW1500 dataset, YOLO alone can achieve only an F1-score of 71.4% due to the reason that curved bounding boxes surpass the representational capability of boxes. Hybrid model scores 85.9% on CTW1500, thus validating the contribution made by the detection branch [5],[10].

Fig. 4. F1-Score Across Benchmark Datasets: ICDAR 2015 vs CTW1500

F. Processing Speed

The overall throughput of the system is 24.5 fps on ICDAR 2015 and 22.8 fps on CTW1500 on RTX 3090. These speeds meet real-time constraints. Throughput would be restored back to 26 fps

V. APPLICATIONS AND USE CASES

A. Assistive Technology

For visually impaired users, mobile apps could use the hybrid detector to recognize shop signages, price tags, and public information announcements on the go, offering audio output. The ability of such a system to detect printed horizontal texts and curved signages is especially useful outside [1]

B. Autonomous Cars

Autonomous cars need to reliably recognize traffic signs and lane markings in the road. The detection branch detects road signs' standard fonts rapidly, while the segmentation branch addresses partially obstructed curved overhead gantry signs. Extension of the hybrid model in the lidar-camera pipeline is obvious [4], [7].

C. Real-Time Translation

Instantly translating application needs accurate and rapid recognition of a text region. High F1-score and nearly real-time detection time makes this model a good fit for the recognition step of real-time translation AR pipelines [3], [5].

D. Smart Surveillance/E-Commerce

CCTV can utilize this system for recognizing number plates and shop IDs. In e-commerce platforms, one would require reading out product titles from image packages. Both these use-cases require detection of both horizontal and curved logotype texts [6], [9].

VI. CHALLENGES AND LIMITATIONS

A. Branch Execution Sequencing

Executing both branches one after the other roughly double inference times. Throughput of 24 fps fulfils most near-realtime needs but not 60+ fps processing [2], [9].

B. Branch Training Independence Training independently does not enable branches to learn complementing representations. Co-training in an

end-to-end manner with a fusion consistency term can provide more gains
[1], [7].

C. Fusion Weights Sensitivity
Pre-defined weights, optimized through grid search using the validation set, will not generalize to other use cases [6], [8].

D. Latin Script-Based Evaluations
Only Latin script evaluation datasets are used. Other scripts such as Arabic, Chinese, Japanese, and Devanagari have different geometric features that need script-specific training [4], [5].

VII. FUTURE SCOPE

A. Joint End-to-End Training
Joint training of all components using a loss that aggregates detection regression, segmentation binarization, and fusion consistency losses will enable branches to learn representations that complement each other; such joint learning is expected to improve the F1 by another 1–2 percentage points [1], [7].

B. Better Backbone Integration Revising the model's backbone to incorporate a Swin
Transformer architecture [3] or TextNet backbone used in FAST [2] can help with curved text recall. The architecture-agnostic fusion module design will not require any changes to accommodate either replacement backbone.

C. Edge and Mobile Computing
Using backbones such as MobileNetV3 and EfficientNetLite, structured channel pruning, and INT8 quantization techniques will be useful for achieving ≥30 fps on ARM-class GPUs with F1 score above 85% on ICDAR 2015 [2].

D. Extension to Full Text Spotting
Polygon representation generated by the segmentation branch makes a natural input to a rectification module that can then pass data to a recognition component, making an extension similar to TESTR [4] or DeepSolo [5].

E. Foundation Model Adaptation
TextSAM-LoRA [10] illustrates efficient use of large foundation models for vision in text detection adaptation. Future work should incorporate the use of a LoRA adapted SAM head as a third stream to contribute to sharper edges in curving or faint texts.

F. Adaptive Fusion and Video Extension
Adaptive weighting of features based on a priori knowledge from a lightweight scene orientation prediction network is likely to enhance results while making the model less sensitive to parameter tuning [6], [9]. Temporal information can be incorporated via optical flow and Kalman filters [3], [7].

G. Multilingual Adaptation
The YOLO-DBNet++ model will be tested on ICDAR 2019 MLT, ReCTS (Chinese), and VinText (Vietnamese) datasets in order to determine how well the proposed approach generalises across languages [4], [5]

VIII. CONCLUSION

The proposed system involves the use of a hybrid approach for scene text detection, which utilizes a detection network based on YOLO with a segmentation network based on DBNet++. These two modules deal with different failure cases, where the detection module detects the locations of the horizontal texts fast and the segmentation network segments the curved and irregular texts accurately [1],[6].

Experimental results on datasets ICDAR 2015 and CTW1500 yield F1 scores of 88.0% and 85.9%, respectively, operating at roughly 24 FPS while consistently outperforming both the individual modules. Compared to eleven other methods from 2022 onwards, the proposed model yields results comparable to the state-of-the-art techniques while using only basic modules that do not require joint training [2],[5]. Moreover, it achieves peak accuracy earlier than the other two methods, and since the design is modular, any improvements on the modules are immediately reflected system-wide [7],[9].

REFERENCES

[1] M. Liao et al., "Real-Time Scene Text Detection with Differentiable Binarization and Adaptive Scale Fusion," IEEE Trans. PAMI, vol. 45, no. 1, pp. 919–931, Jan. 2022.

[2] Z. Chen et al., "FAST: Faster Arbitrarily-Shaped Text Detector with Minimalist Kernel Representation," IEEE Trans. PAMI, vol. 46, no. 1, pp. 278–295, 2024.
[3] M. Huang et al., "SwinTextSpotter: Scene Text Spotting via Better Synergy between Text Detection and Recognition," in Proc. IEEE/CVF CVPR, pp. 4593–4603, 2022.
[4] R. Zhang et al., "TESTR: Text Spotting Transformers," in Proc. IEEE/CVF CVPR, pp. 18901–18910, 2022.
[5] J. Ye et al., "DeepSolo: Let Transformer Decoder with Explicit Points Solo for Text Spotting," in Proc. IEEE/CVF CVPR, pp. 19348–19357, 2023.
[6] W. Jiang et al., "CC-DBNet: A Scene Text Detector Combining Collaborative Learning and Cascaded Feature Fusion," in Proc. ICIC, LNCS vol. 14087, Springer, pp. 351–363, 2023.
[7] Y. Li, "DPNet: Scene Text Detection Based on Dual Perspective CNNTransformer," PLOS ONE, vol. 19, no. 10, p. e0309286, Oct. 2024.
[8] Z. Chen, "HTBNet: Arbitrary Shape Scene Text Detection with Binarization of Hyperbolic Tangent and Cross-Entropy," Entropy, vol. 26, no. 7, p. 560, Jul. 2024.
[9] Y. Cao et al., "EK-Net++: Real-Time Scene Text Detection with Expand Kernel Distance and Epoch Adaptive Weight," Expert Syst. Appl., vol. 262, p. 125607, Dec. 2024.
[10] C. de la Fuente et al., "TextSAM-LoRA: Efficient Fine-Tuning of Segment Anything Model for Text Detection," in Proc. ICDAR Workshop, Springer, 2024.
[11] Y. Wang et al., "A Multi-Scale Natural Scene Text Detection Method Based on Attention Feature Extraction," Sensors, vol. 24, no. 12, p. 3758, Jun. 2024.
[12] M. Yaseen, "What is YOLOv8: An In-Depth Exploration of the NextGeneration Object Detector," arXiv:2408.15857, 2024.