A Hybrid Deep Learning Pipeline for Text Recognition on Greek Squeezes

Monika Radhakisan Chavan[1 [0009−0001−3151−435X]*], Pratiksha Manik Patil[1 [0009−0005−2420−3472]*], Vineetha Mulamoottil Mathews[1 [0009−0002−0313−1316]*], Thomas Gorges[1 [0009−0007−0573−0992]], Janne van der Loop[2 [0000−0001−7486−669X]], and Vincent Christlein[1 [0000−0003−0455−3799]]

[1] Pattern Recognition Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
{monika.r.chavan, pratiksha.m.patil, vineetha.mathews, thomas.gorges, vincent.christlein}@fau.de

[2] Buchwissenschaft, Johannes Gutenberg-Universität Mainz, Germany
jannevanderloop@uni-mainz.de

Abstract. Automatic transcription of ancient Greek squeeze images is challenging because characters are preserved as relief impressions rather than ink and are affected by low contrast, uneven illumination, damaged strokes, and ambiguous shapes. This paper presents our submission to the ICDAR 2026 Competition on Text Recognition on Greek Squeezes (TROGS-26), which generates Latin-proxy transcriptions from images captured under two orthogonal lighting rotations. The main contribution is a systematic evaluation of character-level recognition, line-level sequence modeling, rotation-based ensembling, and confidence-based fusion. The character-level branch combines a ResNet18 classifier with a secondary classifier for visually confusable symbols, while the line-level branch uses a CNN-BILSTM-CTC model. Using ground-truth-derived crops, the character-level, line-level, and fusion configurations achieved CER values of 6.40 %, 9.82 %, and 7.13 %, respectively, on the internal split-mode test partition. Fusion improved over the line-level model but did not outperform the character-level method, indicating that confidence-based fusion was insufficient in this setting when the component accuracies differed substantially. These results should be interpreted as internal evaluation outcomes rather than official competition scores, as the official test labels and evaluation results were unavailable at the time of writing.

Keywords: Greek squeeze recognition · Historical Document Analysis · Text recognition · OCR · TROGS-26

1 Introduction

Ancient Greek inscriptions are important historical and linguistic sources, many of which have been documented using paper impressions known as squeezes [15].

* These authors contributed equally.

A squeeze is produced by wetting paper, pressing it onto an inscribed stone surface, and beating it with a stiff brush so that the paper enters the incisions [7]. After drying, the paper retains a relief-like impression of the inscription. Such collections preserve surface details that may later become damaged, lost, relocated, or inaccessible [22,8].

Automatic transcription of squeeze images is difficult because characters are represented by shallow relief, surface deformation, and illumination-dependent shadows rather than ink. Their visibility depends on lighting direction, surface depth, paper texture, and the physical condition of the impression. Characters may therefore appear weak, broken, incomplete, or merged with the background, particularly in low-contrast regions [6]. These properties make Greek squeeze recognition fundamentally different from conventional printed text recognition and Handwritten Text Recognition (HTR).

The ICDAR 2026 Competition on Text Recognition on Greek Squeezes (TROGS-26) addresses this problem by evaluating systems that generate Latin-proxy transcriptions from scanned squeeze images [5]. The development set contains 224 annotated squeezes, each captured under two orthogonal lighting rotations, yielding 448 image rotation pairs. These rotations provide complementary visual evidence because character details that are weak in one view may be clearer in the other. The annotations include character-level bounding boxes, baseline angles, and line-level transcripts.

System performance is primarily measured using Character Error Rate (CER), which computes the edit distance between predicted and ground-truth transcriptions at the character level [5]. Word Error Rate (WER) is used as a secondary measure of sequence-level transcription quality. Both metrics are widely used in optical character recognition and sequence-recognition systems, particularly those trained with Connectionist Temporal Classification (CTC) [3,18,1].

We investigate a hybrid recognition pipeline combining YOLO11-based character localization [10], character-level classification, line-level CTC-based recognition, confidence-based fusion, and prediction-level merging. The character-level branch uses a ResNet18 classifier together with a secondary classifier for visually confusable proxy characters. The line-level branch employs a CNN-BILSTM-CTC model to capture sequence context across complete text lines.

This work contributes a systematic comparison of character-level and line-level recognition, ablation studies covering preprocessing, secondary classification, stochastic weight averaging, and rotation-based ensembling, and an analysis of detection, layout reconstruction, and confidence-based fusion as sources of transcription error. All quantitative results were obtained on the internal split-mode test partition described in Section 3.1. Since the official competition test labels and evaluation scores were unavailable at the time of writing, these results should not be interpreted as official competition scores.

2 Related Work

Automatic analysis of historical Greek documents includes character detection, recognition, manuscript transcription, and paleographical analysis [21,6]. These tasks are challenging because historical materials often exhibit degradation, low contrast, damaged writing regions, irregular spacing, and substantial variation in writing style [19]. Most previous computational studies have focused on papyri and manuscripts, whereas Greek squeeze recognition remains comparatively less explored [6].

Turnbull and Mannix [21] investigated Greek character detection and recognition for the ICDAR 2023 Competition on Detection and Recognition of Greek Letters on Papyri [17]. Their approach combined an ensemble of YOLOv8 models [24] with recognition refinement using a DeiT transformer and a ResNet-50 model trained through SimCLR-based self-supervised learning on unlabeled data. Their system won the recognition challenge and ranked second in detection, demonstrating the effectiveness of object detection, transformer-based refinement, self-supervised learning, and model ensembling for character-level analysis.

However, papyri and squeezes differ substantially in appearance and text formation. Papyri contain ink-based writing on relatively flat surfaces, whereas squeezes are paper impressions of stone inscriptions in which characters are represented through relief structures, shadows, and local surface deformation rather than ink [21,6,7]. Consequently, methods developed for papyrus manuscripts may not fully address the illumination-dependent appearance and material-specific challenges of Greek squeeze transcription.

Howe *et al.* [6] presented a directly related study using approximately 30,000 Greek squeeze images collected from museums and archaeological sites. Their pipeline combined scene-text recognition with line-based recognition and achieved a CER of approximately 13 %. The study demonstrated the feasibility of automatic squeeze transcription while highlighting the challenges caused by relief-based appearance, illumination variation, material degradation, and variable character visibility. The reported results are not directly comparable with ours because Howe *et al.* evaluated an end-to-end recognition pipeline, whereas our recognition experiments use an internal split with ground-truth-derived character and line crops.

More generally, character-level methods provide fine-grained local evidence but depend on accurate localization and may fail when symbols are damaged, touching, incomplete, or visually similar. In contrast, line-level convolutional and recurrent models trained with CTC avoid explicit character segmentation and exploit sequence context [3,18], although they remain sensitive to severe degradation, irregular spacing, rare symbols, and local visual ambiguity. Building on these directions, our work compares character-level recognition and line-level sequence modeling within a unified Text Recognition on Greek Squeezes (TROGS-26) pipeline and evaluates rotation-based ensembling, confidence-based fusion, automatic character detection, and layout reconstruction for end-to-end transcription.

3 Methodology

The pipeline consists of data preparation, preprocessing, character detection and baseline estimation, line-level recognition, character-level recognition, output fusion, and evaluation. Figure 1 illustrates the complete framework.

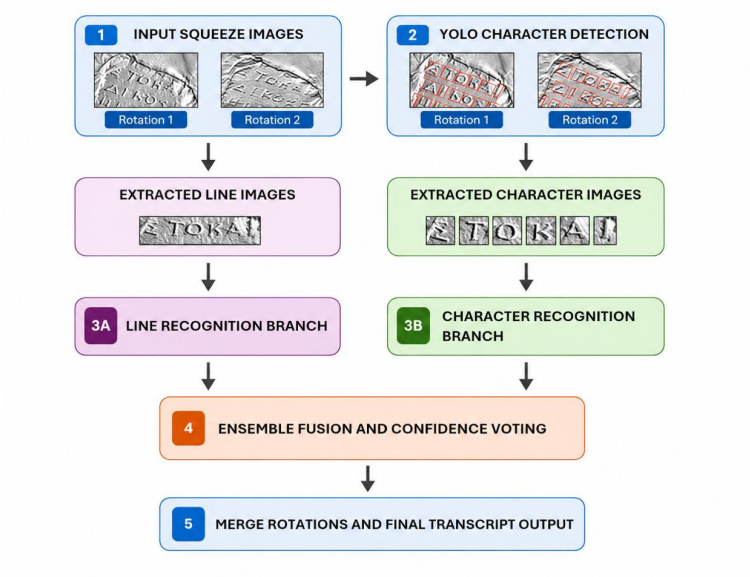


Fig. 1: Overview of the proposed hybrid recognition pipeline combining character-level and line-level prediction branches.

3.1 Data Preparation

The dataset was provided by the ICDAR 2026 Competition on Text Recognition on Greek Squeezes (TROGS-26) [5]. It contains 224 annotated squeezes, each captured under two orthogonal lighting conditions, referred to as Rotation 1 and Rotation 2, yielding 448 image rotation pairs. Each annotation file provides a baseline angle, four corner character bounding boxes, and line-level transcript labels. The competition uses a Latin-proxy encoding scheme in which the 24 Greek letters are represented by Latin symbols, while four additional labels describe non-letter or uncertain cases, as summarized in Table 1.

The development set was divided at the squeeze level into 181 training, 22 validation, and 21 internal test squeezes. Since each squeeze has two rotations, these partitions correspond to 362, 44, and 42 image rotation pairs, respectively. Both rotations of each squeeze were assigned to the same partition, ensuring that no rotated version of a training squeeze appeared in the validation or internal test partition. This strategy was applied to both the character-level and line-level experiments.

Table 1: Latin-proxy labels used for non-letter and uncertain cases.

Proxy label	Meaning
:	Retained as-is
)	Retained as-is
&	Miscellaneous or other characters
*	Partial or illegible characters

For architecture selection and ablation studies, the original training and validation partitions were used separately, with the internal test partition reserved for evaluation. The final character model used three-fold cross-validation on the combined training and validation data, while the YOLO11 detector was trained on the same combined partitions. This *split mode* includes 21 test squeezes (42 image–rotation pairs). Unless stated otherwise, recognition results use ground-truth-derived crops; YOLO11 predictions were used only for end-to-end error analysis and the official test set.

For the final submission, all labeled development data were re-split into 80 % training and 20 % validation subsets. The split was performed at the squeeze level to ensure that both rotations were assigned to the same subset. The models were then applied to the official test set of 50 squeezes, corresponding to 100 image–rotation pairs.

3.2 Data Preprocessing

Using the baseline angle recorded in each annotation file, each squeeze image was rotated to align the text horizontally before crop extraction. Character bounding boxes were grouped into rows to generate line crops, which were padded and resized to 64×1024 pixels. Individual character crops were extracted from the four-corner bounding boxes and resized to 64×64 pixels.

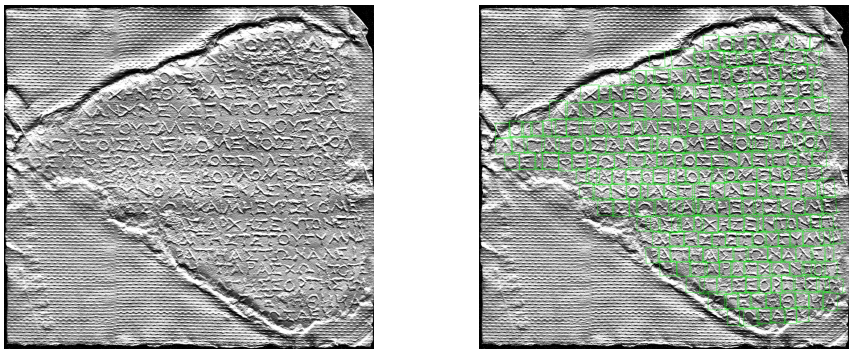
For character recognition, samples of the * class were excluded because no character-level bounding boxes were available for this label. The resulting 27-class dataset contained 106,923 training, 8,108 validation, and 9,006 internal test character crops in the original split. The distribution was strongly imbalanced, ranging from more than 11,000 samples for frequent classes such as A/α, O/ο, and I/ι, to fewer than 100 samples for rare classes such as : and &.

The original grayscale input and four preprocessing alternatives were evaluated: (1) a full enhancement pipeline combining bilateral filtering [20], median filtering, flat-field illumination correction, Contrast Limited Adaptive Histogram Equalization (CLAHE) [25], frequency separation, and unsharp masking; (2) median filtering followed by CLAHE; (3) CLAHE followed by Sauvola adaptive thresholding [16]; and (4) pixel-level rotation fusion based on the pixel-wise minimum of the aligned orthogonal views. Their relative performance is reported in Section 4.2.

3.3 YOLO11-Based Character Detection and Baseline Estimation

A YOLO11-based detection pipeline [10] was used to localize individual characters and estimate the text baseline, as shown in Step 2 of Figure 1. The predicted boxes were used to evaluate the end-to-end character pipeline and to generate character and line crops for the official competition test set. The corresponding detection results are reported in Section 4.1.

For the internal split-mode experiments, the detector was trained using the combined training and validation partitions and evaluated on the internal test partition. The input images were resized to the required YOLO11 resolution and augmented using scaling, small rotations, brightness variations, and minor geometric transformations. Each character was treated as an individual object instance. Figure 2 shows an example of the original squeeze image and the resulting detections.



(a) (b)

Fig. 2: Example of YOLO11-based character detection on a sample image from the TROGS-26 dataset [5]. (a) Original squeeze image; (b) detected character bounding boxes.

The reading order followed the normalized order defined by the TROGS-26 annotations [5], rather than an assumed historical writing direction. To estimate the baseline, the bottom-center point of each detected bounding box was extracted, and a straight line was fitted through these points. The slope of the fitted line was converted into a baseline angle using the arctangent function.

For each image, the detected bounding boxes and estimated baseline angle were stored in an annotation file. In split mode, these predictions were evaluated on the internal test partition. For the official competition submission, the detector was retrained using all labeled development data and applied to the official test set, where ground-truth bounding boxes and baseline angles were unavailable.

3.4 Line-Based Approach

The line-based branch in Step 3A of Figure 1 predicts complete text-line transcriptions from deskewed line images. Unlike the character-based branch, which classifies individual crops independently, it processes the full line and captures contextual information between neighboring characters. Each line image was converted to grayscale and resized to 64×1024 pixels.

Network Architecture The convolutional backbone extracts visual features while preserving the horizontal writing direction. It consists of residual-style convolutional blocks and max-pooling layers configured as

$$[[2, 64], M, [3, 128], M, [2, 256]],$$

where $[n, c]$ denotes a block with n convolutional layers and c output channels, and M denotes max pooling. After feature extraction, the height dimension is compressed and the width dimension is interpreted as a sequence of feature vectors.

The sequence is processed by a three-layer bidirectional LSTM with hidden size 256. A final linear layer maps the recurrent features to the 27 proxy classes and the CTC blank symbol. CTC decoding removes repeated symbols and blanks to produce the final transcription.

Training Procedure The model was trained on line images and their corresponding transcriptions using CTC loss [3]. One rotation per squeeze was used during training to reduce near-duplicate samples, since both rotations share the same transcription and closely related visual content.

The evaluated architectures included Convolutional Neural Network (CNN)-BiLSTM-CTC, ResNet34-BiLSTM-CTC [4], CNN-LSTM-CTC, Transformer-CTC [23], and CNN-Stem-ViT-CTC [2]. Training used a warm-up cosine learning-rate schedule [12]. Additional experiments used Stochastic Weight Averaging (SWA) [9], in which weights from later epochs were averaged to improve generalization.

3.5 Character-Based Approach

The character-based recognition module in Step 3B of Fig. 1 classifies individual deskewed character crops independently into one of the 27 proxy classes.

Network Architecture Two architectures were evaluated for character classification: ResNet18 [4] and a Vision Transformer (ViT) [2]. Both models were adapted to grayscale 64×64 character crops, trained from scratch, and equipped with output layers for the 27 proxy classes. The architecture with the lowest validation CER was selected.

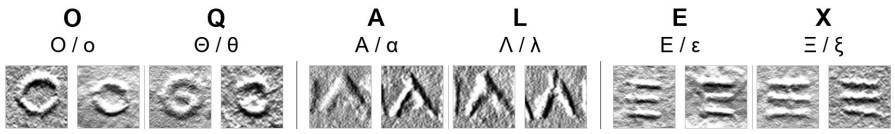


Fig. 3: Example character crops illustrating visually confusable proxy pairs from the TROGS-26 dataset [5].

Two-Stage Classification To improve recognition of visually similar symbols, a two-stage classification strategy was employed. The primary classifier predicts the complete 27-class label set, while a secondary specialized classifier operates on 17 frequently confused classes: O/ο, Θ/ϑ, N/ν, H/η, A/α, Λ/λ, Z/ζ, Ξ/ξ, I/ι,), :, E/ε, Σ/σ, P/ρ, Π/π, T/τ, and Υ/υ. Figure 3 shows representative examples. The secondary classifier uses the same architecture as the selected primary model, with its output layer restricted to the confusable subset.

Training Procedure Four training configurations were evaluated by combining cross-entropy (CE) or focal loss ($\gamma = 2.0$) [11] with either random sampling or balanced batch sampling with augmentation. Class-dependent augmentation was applied with probabilities of 70 % for rare classes and 30 % for common classes. The augmentation pipeline included dilation or erosion, blurring, translation, scaling, and rotation. All models were optimized using AdamW [13] with a cosine learning-rate schedule and warm-up [12].

The primary and secondary classifiers followed the same three-stage procedure. First, both candidate architectures were evaluated under the four training configurations, and the best architecture was selected using validation CER. Second, the selected architecture was evaluated with the five input variants described in Section 3.2. Finally, three-fold stratified cross-validation was performed on the combined training and validation data using the selected architecture, training configuration, and preprocessing strategy. Softmax probabilities from the three-fold models were averaged during inference.

Inference Strategy Each crop was first processed by the primary classifier. If the predicted class belonged to the confusable subset and its confidence was below a selected threshold, the crop was reclassified by the secondary classifier and its prediction replaced the primary output. Thresholds of 0.55, 0.75, 0.85, and 0.95 were evaluated, together with an always reclassify variant.

Test-time augmentation used the original crop, dilation, erosion, blurring, and affine jitter. The corresponding softmax probabilities were averaged. Character predictions were then assembled using the annotation-defined reading order in split mode and the YOLO11 derived order in the end-to-end and official competition pipelines.

3.6 Ensemble Approach

Final predictions were obtained by combining the character-based and line-based branches, as shown in Steps 4 and 5 of Figure 1. The character-based branch averaged probabilities across three cross-validation folds for both the primary and secondary classifiers, while the line-based CNN-BiLSTM-CTC model used SWA [9]. Its per-character confidence scores were derived from the softmax probabilities at each decoded position.

For each text line, the two predictions were merged using confidence-based voting. For equal-length sequences, matching characters were retained and conflicts were resolved using the higher local confidence. For unequal-length sequences, Needleman-Wunsch global alignment [14] was first applied to establish character correspondences, followed by confidence-based selection at each aligned position. After model-level fusion, predictions from the two orthogonal rotations were combined through a prediction-level rotation merge.

3.7 Evaluation Metrics

CER was the primary metric for model selection and evaluation. It is defined as

$$\mathrm{CER} = \frac{S + D + I}{N}, \tag{1}$$

where S, D, and I denote substitutions, deletions, and insertions, respectively, and N is the number of characters in the ground-truth transcription.

For the line-based branch, WER was also reported to measure transcription errors at the word level. Both metrics are commonly used in Optical Character Recognition (OCR) and CTC-based sequence-recognition systems [3,18,1].

4 Experimental Results

This section presents the character-detection, character-recognition, line-recognition, and ensemble results obtained on the internal split-mode test partition. Unless otherwise stated, the character and line recognition results use crops generated from the ground-truth annotations, as described in Section 3.1.

4.1 YOLO11-Based Character Detection Results

Table 2 shows that the high precision reflects few false detections, whereas the F1-score indicates the overall balance between precision and recall. The lower recall indicates that some ground-truth characters remain undetected. The relatively low mAP@0.50:0.95 indicates that the predicted bounding boxes are not consistently well aligned with the ground-truth boxes across stricter Intersection over Union (IoU) thresholds. Overall, missed and inaccurately localized characters are the detector's main limitations and contribute to errors in crop extraction, reading order, and layout reconstruction in the end-to-end recognition pipeline.

Table 2: YOLO11 character-detection results.

Metric	Value	Interpretation
Precision@0.50	93.89 %	Correctness of predicted boxes.
F1-score@0.50	80.03 %	Balance between precision and recall.
Recall@0.50	69.73 %	Coverage of ground-truth characters.
mAP@0.50:0.95	34.32 %	Localization robustness across multiple IoU thresholds.

4.2 Character-Based Classification Results

This section reports validation and test results for the two-stage character-recognition pipeline described in Section 3.5, where a primary classifier predicts all proxy classes and a secondary classifier re-evaluates uncertain predictions from visually confusable classes.

Architecture Comparison ResNet18 and ViT were trained from scratch under four configurations, combining CE or focal loss with either random sampling or balanced batch sampling with class-dependent augmentation. Table 3 reports the corresponding CER values. ResNet18 outperformed ViT in every configuration, achieving the best validation CER of 6.80% with CE loss and balanced sampling, compared with 16.60% for the best ViT configuration. This difference may reflect the greater data requirements of transformer architectures, while ResNet18 may be better suited to learning local stroke patterns from small character crops under limited-data conditions.

Table 3: Architecture comparison on the development validation set and internal split-mode test partition. CE = cross-entropy.

Architecture	Loss	Sampling	Aug.	CER (%)	
				Val	Test
ResNet18	CE	Balanced	Yes	**6.80**	**6.40**
ResNet18	Focal	Balanced	Yes	7.03	6.41
ResNet18	CE	Random	No	7.42	7.02
ResNet18	Focal	Random	No	8.04	7.40
ViT	Focal	Balanced	Yes	16.60	16.02
ViT	CE	Balanced	Yes	17.12	16.37
ViT	CE	Random	No	20.77	21.01
ViT	Focal	Random	No	21.60	21.17

Balanced sampling improved both architectures, particularly for rare classes. Focal loss slightly underperformed CE for ResNet18 but marginally improved

Table 4: Validation and test CER (%) across pre-processing strategies and training configurations. CE = cross-entropy; Bal = balanced sampling with class-dependent augmentation; Rand = random sampling. *Rotation fusion uses fewer samples because paired orthogonal views are merged.

Strategy	**CER (%)**							
	CE+Bal		**Focal+Bal**		**CE+Rand**		**Focal+Rand**	
	Val	**Test**	**Val**	**Test**	**Val**	**Test**	**Val**	**Test**
Original (none)	**6.80**	6.40	7.03	6.41	7.42	7.02	8.04	7.40
Median + CLAHE	7.18	6.87	7.13	6.57	8.05	7.14	8.04	7.20
Rotation fusion*	7.45	7.00	7.18	**6.33**	8.24	7.95	8.76	8.37
CLAHE + Sauvola	9.93	9.82	9.93	10.44	10.85	10.76	11.37	11.16
Full enhancement	10.22	9.60	10.61	10.80	10.69	10.86	11.68	11.00

ViT. Because the loss-dependent differences remained below one CER point, sampling had a greater effect than the loss function. A confidence threshold of 0.75 produced the best two-stage results, showing that selective reclassification was more effective than aggressive reclassification. ResNet18 was therefore selected for subsequent experiments.

Pre-processing Strategy Comparison Five pre-processing strategies were evaluated with ResNet18 under the same four training configurations. Table 4 shows that the original grayscale input generally outperformed enhancement and binarisation. Among the alternatives, median filtering with CLAHE and rotation fusion achieved the lowest validation CER values of 7.13% and 7.18%, respectively.

Although rotation fusion used fewer training samples because paired views were merged, it remained competitive, indicating that the two lighting directions provide complementary character information. In contrast, CLAHE with Sauvola thresholding and the full enhancement pipeline increased CER above 9%. Squeeze characters are represented by subtle relief-induced intensity variations rather than ink; therefore, CLAHE may amplify background texture, while binarisation may remove informative shading and shadow cues. These results support retaining the original grayscale input.

Balanced sampling consistently outperformed random sampling, whereas CE and focal loss were comparable overall. The final system therefore used original grayscale crops, ResNet18, CE loss, balanced sampling, and class-dependent augmentation. Although rotation fusion achieved a slightly lower test CER of 6.33%, the original grayscale configuration had the best validation CER and was selected to avoid test-set-based model selection.

Effect of the Secondary Classifier The secondary classifier was evaluated by comparing the primary-only ResNet18 model trained with CE loss and balanced sampling against the complete two-stage pipeline at the selected threshold of 0.75.

The primary-only model achieved validation and test CER values of 7.07% and 7.27%, respectively, which decreased to 6.80% and 6.40% with the secondary classifier. Across the 17 confusable classes in the validation set, the two-stage pipeline reduced the total error count from 458 to 435, corresponding to a net reduction of 23 errors. The largest gains were observed for O/ο (6.7% → 5.8%, −8 errors), Λ/λ (18.6% → 16.2%, −7), H/η (7.2% → 5.2%, −6), and Σ/σ (7.2% → 6.5%, −5). In contrast, error rates increased slightly for P/ρ (4.0% → 5.5%, +5 errors) and Ξ/ξ (25.0% → 33.3%, +2), indicating that low confidence does not always correspond to an incorrect primary prediction and that the secondary classifier can occasionally replace correct primary outputs.

Detection vs. Recognition Error Analysis Using ground-truth character boxes, the selected ResNet18 classifier achieved a CER of 6.40 %, representing recognition performance without automatic detection and layout-reconstruction errors. Replacing the ground-truth boxes with YOLO11 detections increased the end-to-end CER to 34.27 %. The resulting 27.87 percentage-point gap reflects the combined effect of missed or inaccurate detections, reading-order errors, character-grouping errors, and layout reconstruction. These results indicate that automatic localization and layout reconstruction are the main bottlenecks in the end-to-end pipeline.

4.3 Line-Based Recognition Results

This section reports results for the line-based pipeline described in Section 3.4. Architecture selection and ablation experiments used the original split containing 3,771 training lines, 322 validation lines, and 316 internal test lines. Model selection was based on validation CER.

Architecture Comparison Five sequence-recognition architectures were evaluated under the same 100-epoch training protocol without SWA. Table 5 summarizes the results. Without SWA, ResNet34-BiLSTM-CTC achieved the lowest validation CER (14.97 %), followed by CNN-LSTM-CTC (17.49 %) and Transformer-CTC (21.36 %). The CNN-BiLSTM-CTC model without SWA showed substantial overfitting at epoch 100; however, the same architecture with SWA reduced the validation CER to 14.11 % and achieved a test CER of 9.82 % (Table 6), supporting its selection for the final line-based system.

The selected CNN-BiLSTM-CTC model with SWA achieved a lower validation CER than the CNN-LSTM-CTC baseline. However, because the two configurations differ in both recurrent directionality and the use of SWA, the improvement cannot be attributed to bidirectionality alone.

Within the evaluated configurations, the transformer-based architectures underperformed the recurrent alternatives. This may reflect the limited number of available line samples, although differences in architecture and optimization prevent attributing the performance gap solely to dataset size.

Table 5: Architecture comparison for line-based recognition.

Model	Params	Val		Test	
		CER	WER	CER	WER
ResNet34-BiLSTM-CTC	24.1 M	14.97	62.42	10.35	58.86
CNN-LSTM-CTC	4.6 M	17.49	75.16	13.79	72.15
Transformer-CTC	6.3 M	21.36	85.40	16.50	79.75
CNN-BiLSTM-CTC*	7.3 M	38.21	98.45	31.11	97.78
CNN-Stem-ViT-CTC	5.1 M	27.33	90.06	22.19	86.71

*The same architecture achieves 9.82 % test CER with SWA; see Table 6.

Effect of Stochastic Weight Averaging Table 6 presents the effect of SWA on the two selected line-recognition architectures. For CNN-BiLSTM-CTC, SWA substantially reduced the overfitting observed at epoch 100, lowering the validation CER from 38.21 % to **14.11 %** and yielding a test CER of 9.82 %.

In contrast, SWA degraded performance for ResNet34-BiLSTM-CTC, with the validation CER increasing from 14.97 % to 23.06 %. One possible explanation is that the constant SWA learning rate of 10^{-4} was too high for the larger 24.1 M-parameter model, although this was not evaluated independently.

Overall, CNN-BiLSTM-CTC with SWA achieved the best observed line-level test CER. Because this configuration differs from the ResNet34 alternative in both architecture and SWA usage, the performance difference cannot be attributed to either factor independently.

Table 6: Effect of SWA on selected line-recognition architectures.

Model	SWA	Val CER	Val WER	Test CER	Test WER
CNN-BiLSTM-CTC	×	38.21	98.45	31.11	97.78
CNN-BiLSTM-CTC	✓	**14.11**	**64.29**	**9.82**	**60.44**
ResNet34-BiLSTM-CTC	×	14.97	62.42	10.35	58.86
ResNet34-BiLSTM-CTC	✓	23.06	99.07	16.68	96.84

Values are reported in %. SWA runs use weight averaging from epoch 75 with a learning rate of 10^{-4}.

Error Analysis The WER of ∼60% despite a CER below 10% indicates that errors are distributed across many lines, often as one or a few character-level mistakes per word or line. Most errors were single-character substitutions among visually similar proxy pairs, such as O/ο↔Θ/ϑ, I/ι↔Λ/λ, T/τ↔Π/π, and Λ/α↔Λ/λ. Short lines with fewer than five characters and rare symbols such as &, :, Ψ/ψ, and Z/ζ were disproportionately affected. This suggests that the remaining errors

are mainly due to local visual ambiguity and limited examples for rare classes, rather than to complete line-level recognition failure.

4.4 Final Ensemble and Test Set Results

Since ground-truth annotations for the official competition test set were not released, all quantitative results in this subsection are reported on the internal split-mode test partition comprising 21 squeezes and 42 image–rotation pairs. The final competition submission used the same selected configuration retrained on the pooled labeled development data, as described in Section 3.1.

The character-based component is a ResNet18 classifier using the selected configuration, comprising CE loss, balanced sampling, class-dependent augmentation, and the two-stage framework for confusable characters. Its predictions were obtained using three-fold cross-validation with probability averaging across folds at inference time. The line-based component is a CNN-BiLSTM-CTC model trained with SWA. The two branches were fused using confidence-based voting with Needleman–Wunsch alignment and prediction-level rotation merging, as described in Section 3.6.

On the internal split-mode test partition, the character-based branch achieved the lowest CER of 6.40 %, followed by the ensemble at 7.13 % and the line-based branch at 9.82 %. The ensemble therefore improved over the line-based branch but did not outperform the stronger character-based branch. A likely explanation is the substantial accuracy gap between the two components: because fusion relies on local confidence, a high-confidence error from the line-based model can replace a correct prediction from the character-based branch.

5 Discussion

The experimental results provide several insights into text recognition on ancient Greek squeezes.

Scope of contribution. The objective of this work was not to introduce a fundamentally new recognition architecture, but to systematically evaluate and combine established character-level and line-level approaches for Greek squeeze recognition. The contribution lies in the comparative evaluation and ablation analysis of these approaches, together with the identification of the limitations of confidence-based fusion when the component models have unequal recognition accuracy.

Character-level vs. line-level recognition. The character-based ResNet18 classifier achieved a lower test CER than the line-based CNN-BiLSTM-CTC model (6.40 % vs. 9.82 %). The character-based branch extracts individual character crops from character bounding boxes, whereas the line-based branch groups detected characters into rows to construct text-line crops; split-mode results use ground-truth boxes for both, while YOLO-based detections are used only for detection

error analysis in split mode and, in the final test pipeline, to generate character boxes and estimate the baseline angle. The branches thus provide complementary evidence: fine-grained symbol classification versus sequence context via CTC decoding over complete text lines.

Architecture selection. For character recognition, ResNet18 substantially outperformed ViT (6.40 % vs. 16.02 % test CER). Since both models were trained from scratch, this difference likely reflects the greater data requirements of transformer architectures rather than a fundamental limitation of attention-based models; pretrained or more recent ViT variants may perform differently with sufficient data. For line recognition, the CNN-BiLSTM-CTC configuration with SWA achieved the best observed test CER of 9.82 %, compared with 10.35 % for ResNet34-BiLSTM-CTC without SWA. Because these configurations differ in both architecture and the use of SWA, the performance difference cannot be attributed to either factor independently. The selected CNN-BiLSTM-CTC configuration also achieved an approximately 29 % relative reduction in CER compared with the unidirectional CNN-LSTM-CTC baseline (9.82 % vs. 13.79 %). However, because the configurations may differ in additional training components, this result should be interpreted as an overall configuration-level improvement rather than as an isolated effect of bidirectionality.

Ensemble behavior. The ensemble improved over the line-based branch but did not outperform the stronger character-based branch (7.13 % vs. 6.40 % CER). Symmetric confidence-based voting assumes that both branches have comparable reliability and that local confidence indicates which prediction should be selected. This assumption does not hold in the present setting: the less accurate line-based model can produce high-confidence errors that override correct character-based predictions. Future fusion methods should therefore account for branch-level accuracy, for example through reliability weighting or validation-calibrated confidence scores.

6 Conclusion

This paper presented a hybrid pipeline for text recognition on ancient Greek squeezes, developed for the ICDAR 2026 TROGS-26 competition. The system combines character-level ResNet18 classification with line-level CNN-BiLSTM-CTC recognition and uses confidence-based voting with Needleman–Wunsch alignment [14] to merge predictions from both branches. On the split-mode test partition, the character-based branch achieved the lowest CER of 6.40 %, followed by the ensemble at 7.13 % and the line-based branch at 9.82 %. The selected 7.3M-parameter CNN-BiLSTM-CTC configuration with SWA [9] achieved the best observed line-level test CER among the evaluated configurations. Because architecture and SWA were not varied independently, their individual contributions cannot be isolated from these experiments.

Although the two branches provide complementary evidence, symmetric confidence-based fusion did not outperform the stronger character-based branch

because the component models differed substantially in accuracy. In the end-to-end and official competition pipelines, both branches depend on YOLO11-based detections, which remain a shared source of error. Future work will therefore focus on improving detector robustness and developing fusion methods that account explicitly for differences in branch reliability.

Code availability. The source code, configuration files, and evaluation scripts used in this study are available at https://github.com/monikaits44/trogs_greek_text_recognition.

Acknowledgments. The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU). The hardware is funded by the German Research Foundation (DFG). The authors thank Smith College for providing the Greek Squeezes annotated dataset. The authors acknowledge support from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (Grant Agreement No. 101167454).

References

1. Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S.J., Lee, H.: What is wrong with scene text recognition model comparisons? dataset and model analysis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4714–4722. IEEE (2019)
2. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
3. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning. pp. 369–376 (2006)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
5. Howe, N.R.: TROGS-26: The ICDAR 2026 Competition on Text Recognition on Greek Squeezes. https://www.science.smith.edu/~nhowe/contest/trogs26.html (2026), last accessed: 26-05-2026
6. Howe, N.R., Chang, F., Falbo, I., Brown, T., Hershkowitz, A.: Character recognition for Greek squeezes. International Journal on Document Analysis and Recognition **28**, 345–356 (2025)
7. Iacchei, J.M.: Epigraphic squeezes: Part i. https://blogs.cornell.edu/culconservation/2013/12/13/epigraphic-squeezes-part-i/ (2013), last accessed: 26-05-2026
8. Institute for Advanced Study: IAS receives NEH grant for squeeze digitization project. https://www.ias.edu/press-releases/2020/neh-grant-squeeze (2020), last accessed: 26-05-2026
9. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., Wilson, A.: Averaging weights leads to wider optima and better generalization. arXiv preprint arXiv:1803.05407 (2018)

10. Khanam, R., Hussain, M.: Yolov11: An overview of the key architectural enhancements (2024), https://arxiv.org/abs/2410.17725
11. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2999–3007 (2017)
12. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. In: International Conference on Learning Representations (2017)
13. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019)
14. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology **48**(3), 443–453 (1970)
15. Osborne, R.: Greek inscriptions as historical writing. In: Feldherr, A., Hardy, G. (eds.) The Oxford History of Historical Writing: Volume 1: Beginnings to AD 600, pp. 97–121. Oxford University Press, Oxford (2011)
16. Sauvola, J., Pietikäinen, M.: Adaptive document image binarization. Pattern Recognition **33**(2), 225–236 (2000)
17. Seuret, M., Marthot-Santaniello, I., White, S.A., Serbaeva Saraogi, O., Agolli, S., Carrière, G., Rodriguez-Salas, D., Christlein, V.: ICDAR 2023 Competition on Detection and Recognition of Greek Letters on Papyri. In: Fink, G.A., Jain, R., Kise, K., Zanibbi, R. (eds.) Document Analysis and Recognition – ICDAR 2023. pp. 498–507. Springer Nature Switzerland, Cham (2023)
18. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(11), 2298–2304 (2017)
19. Sulaiman, A., Omar, K., Nasrudin, M.F.: Degraded historical document binarization: A review on issues, challenges, techniques, and future directions. Journal of Imaging **5**(4), 48 (2019)
20. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of the Sixth International Conference on Computer Vision. pp. 839–846 (1998)
21. Turnbull, R., Mannix, E.: Detecting and recognizing characters in Greek papyri with YOLOv8, DeiT and SimCLR. International Journal on Document Analysis and Recognition **28**, 277–285 (2025)
22. Tweten, L.: From stone to screen : digitizing epigraphic squeezes (Oct 2014), https://open.library.ubc.ca/collections/ubclibraryandarchives/67656/items/1.0075855
23. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 6000–6010. Curran Associates Inc. (2017)
24. Yaseen, M.: What is yolov8: An in-depth exploration of the internal features of the next-generation object detector (2024), https://arxiv.org/abs/2408.15857
25. Zuiderveld, K.J.: Contrast limited adaptive histogram equalization. In: Heckbert, P.S. (ed.) Graphics Gems IV, pp. 474–485. Academic Press (1994)