Enhanced Text Extraction from Low-Quality Legacy Documents Using TrOCR Assisted by Line Detection

A Kirti Meghana
Dept of CSE
Sharda University
Greater Noida, India
akmeghana29@gmail.com

Harsh Gupta
Dept of CSE
Sharda University
Greater Noida, India
harshgupta7739@gmail.com

Preeti Dubey
Dept of CSE
Sharda University
Greater Noida, India
preetidubey.research@gmail.com

***Abstract*—Legacy documents are mostly found in unstructured formats that make them difficult to retrieve digitally. For Microsoft's TrOCR model, we used a domain-specific fine-tuning method paired with a line detection model using detectron2 framework and post OCR correction using SymSpell to improve the detection of low-resolution, faded, and degraded text. The data was collected from archived Indian government documents. Using mixed-precision training on constrained resources, the model was trained on our proprietary dataset, which was annotated line-wise. With a character error rate of 7.01% and a word error rate of 14.21% on the validation set, the model showed a very significant improvement over the base version. Our findings demonstrate the great efficacy of specifically pretrained OCR model fine-tuning for the large-scale restoration and digitization of important historical data contributing significantly to sustainable data preservation and access.**

***Keywords*—Legacy data, TrOCR, detectron2, SymSpell, Line-wise annotations, Unstructured data**

I. INTRODUCTION

Digitizing historical and legacy documents is becoming more and more important for public service systems, academic institutions, and government agencies in today's data-centric environment. Although they are usually kept as low-resolution, scanned PDFs, many official documents, including those released by India's Ministry of Statistics and Programme Implementation (MOSPI), include insightful information [1]. It is quite challenging to extract structured information from these documents since they frequently have faded ink, mixed fonts, multilingual content, and irregular layouts [2].

Despite their widespread use, traditional OCR systems like Tesseract are not made to manage such complicated situations. They frequently perform poorly, especially when confronted with distorted or faint characters and irregular typefaces or layouts [3], [4]. This emphasizes how urgently clever OCR models are need to be built.

Transformer-based models have lately demonstrated potential in generic OCR workloads, particularly VisionEncoderDecoder architectures like Microsoft's TrOCR. TrOCR facilitates end-to-end text recognition straight from photos by fusing a language-aware decoder like BART with a visual encoder like ViT. Nevertheless, when used at the clipped line level, pre-trained models are rarely optimized for specialized domains such as statistical government reports. To solve this, we used a dataset of annotated line images taken from MOSPI reports to refine the TrOCR-base-printed model.

The noisy, real-world characteristics of historical data, such as blurred text, different alignments, different font styles, and numerical formats, were captured by these samples [5]. We initially experimented with other OCR models; however, we found out that TrOCR's superior ability to handle complex layouts, degraded text and fine-tuning capabilities makes it the most reliable choice for our use case. But since, TrOCR is a model based on single-line text extraction, we have used a line detection model prior to the text extraction model. Our objective was to use supervised fine-tuning to improve TrOCR's ability to adapt. Two individual labelled datasets were manually prepared to fine-tune each model. The model is designed to handle challenging lines while preserving semantic structure.

This study demonstrates how transformer topologies combined with domain-specific fine-tuning can significantly enhance the digitization of historical data, releasing priceless documents for public access, policymaking, and research.

II. LITERATURE REVIEW

Traditional OCR systems like Tesseract frequently fail when used on damaged documents, since they mostly rely on clean, organized input. These drawbacks have led to the creation of deep learning-based models that offer increased accuracy and versatility by fusing verbal reasoning with image understanding.

In the realm of document digitization and archive automation, text extraction from historical documents remains a major difficulty. Conventional optical character recognition engines, like Tesseract, frequently struggle with noisy, low-resolution, or distorted documents and mainly depend on clean, structured input. In order to improve robustness and generalization across a larger range of document formats, the research community has been compelled by these restrictions to develop more sophisticated deep learning-based models that integrate visual comprehension with natural language reasoning.

Microsoft's proposed transformer-based OCR technology, TrOCR, represents a significant advancement in this field [6]. TrOCR uses a language model decoder, which works similarly to BART. The true power of TrOCR is its versatility; when refined on domain-specific datasets, The model maintains impressive performance even on documents that are deformed, multilingual, or degraded. Recent tests show that TrOCR consistently maintains its robustness while handling scans, even those that are noisy and blurry.

979-8-3315-9361-2/25/$31.00 ©2025 IEEE

TrOCR has been used by a number of researchers on government, administrative, and historical archival datasets, and they have found that it significantly increases recognition accuracy, especially when the model is trained at line level [7]. Page-level, paragraph-level, and line-level annotation strategies have all been compared in studies, and it has been found that line-wise annotation consistently performs better when dealing with irregular fonts and low scan qualities. Furthermore, spatial attention mechanisms are introduced by advanced TrOCR variants such as LOCR, which improve recognition for cluttered layouts [8].

Although TrOCR has a powerful recognition capacity, the quality of the input segments has a significant impact on how well it performs. The outcome is frequently deteriorated when entire pages or unstructured text sections are fed into the algorithm. In order to extract clear, distinct lines from noisy or unstructured document pictures, line detection and segmentation have thus become crucial preprocessing stages.

A popular region-based object detection framework for layout and line detection is called Detectron2, which was created by Facebook AI Research. To precisely identify and crop individual text lines, researchers refine Detectron2 on custom datasets that have been annotated using programs like Label Studio or Doccano. OCR models such as TrOCR are then fed these updated line-level crops. It has been demonstrated that the two-step process of segmentation followed by OCR performs noticeably better than some end-to-end deep learning alternatives as well as conventional rule-based OCR systems [9]. Additionally, Detectron2 has been shown to be more successful at extracting fine-grained regions from scanned medical and historical forms than YOLO in benchmarking for document layout understanding [10].

Another technology, PaddleOCR 3.0 is an open-source OCR and document parsing toolkit developed by PaddlePaddle [11]. It showcases three key components: PP-OCRv5 for multilingual text recognition, PP-StructureV3 for hierarchical document layout parsing, and PP-ChatOCRv4 for key information extraction from documents. The architecture emphasizes modular design and efficiency in large-scale, multilingual, and structured document understanding

Recent research has suggested using generative data augmentation to generate more diverse and noisy samples to mimic natural distortions in training data, which has been demonstrated to significantly increase recognition accuracy on unseen document types [12]. In structured data, such as official fields or numeric sequences, post-OCR error correction is frequently used to improve raw outputs. SymSpell, a lightweight dictionary-based spelling corrector, can increase accuracy by 12–15% [13].

Direct comparisons between the YOLO and Detectron2 designs in layout analysis benchmarking show that, while having a larger computing cost, Detectron2 achieves higher precision on small or irregular layouts [14]. The contextual layout comprehension of advanced transformer-based models such as LayoutLM, LayoutLMv3, and Donut is attractive, but they usually require large amounts of compute and labelled datasets, which makes them unfeasible in many situations with restricted resources [15]. This is addressed by parameter-efficient fine-tuning techniques like DLoRA-TrOCR, which reduce computation requirements while preserving state-of-the-art performance in mixed text recognition [16] by requiring only ~0.7% of model parameters to be retrained.

In addition, the identification of text blocks, tables, and figures is greatly enhanced when off-the-shelf Detectron2 based layout detection (e.g. Mask R CNN backbones) is applied to old or deteriorated scans [17]. A well-developed hybrid OCR system, like PaddleOCR v3.0, has integrated modules for multilingual recognition and key information extraction in complicated documents, and it enables multi-script and multi-layout parsing [18]. Tokenization-based post-OCR correction (such as sequence to sequence or edit distance models) may handle unusual word tokens and domain-specific errors in addition to recognition [19]. Lastly, some procedures attain state-of-the-art performance on low-quality legacy reports by combining a fine-tuned TrOCR engine with PaddleOCR preprocessing [20].

When combined, the steps - input augmentation first, layout segmentation, recognition engine selection, and mistake correction produce a versatile, effective, and modular pipeline that is especially well-suited for real-world document digitization activities.

As new models are developed, this modular architecture facilitates simple updates. It guarantees resilience to subpar inputs. Performance is further improved by domain-specific tweaking. All things considered, the pipeline strikes a compromise between dependability and flexibility for field deployment. Table I summarises all the methods discussed in this section. It includes all the explored technologies for text extraction and mentions the strengths and limitations of each.

Existing OCR pipelines such as Tesseract and PaddleOCR are difficult to modify to accommodate domain-specific noise and formatting differences present in old government files. Because Tesseract is primarily rule-based and uses predefined language models, it has trouble handling fading or clipped text and generates a lot of deletion mistakes. Despite being deep learning based, PaddleOCR frequently breaks characters in low contrast scans and depends on CTC-based recognition. Although it is not designed for this domain, baseline pretrained TrOCR performs better. Our method, in contrast, enables the model to learn local typefaces, noise patterns, and structural layouts by combining line-level segmentation with domain-specific TrOCR fine-tuning. The advantage of tailoring transformer-based OCR models to particular domains is demonstrated by the consistently decreased Character Error Rate (CER) and Word Error Rate (WER), especially in low-quality scans. Hence after careful review and analysis, it was concluded that combining TrOCR with a line detection model yields the highest accuracy for text extraction from legacy data.

TABLE I. SUMMARY OF LITERATURE REVIEW

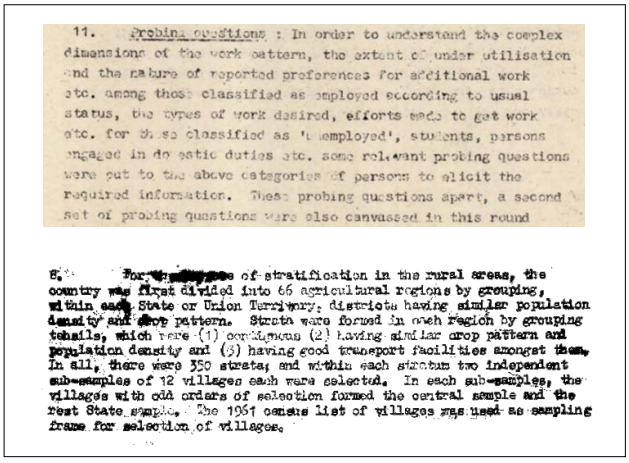
S. No	Technology/Method	Purpose	Key Contributions/ Findings
1.	Tesseract (Traditional OCR)	Baseline OCR engine	Struggles with low-quality, distorted, or unstructured input documents. Limited to clean inputs.
2.	TrOCR (Transformer-based OCR)	Visual-language OCR	Performs well on degraded, multilingual, and noisy data. Uses language model decoder. Line-level training improves results.
3.	LOCR (TrOCR Variant)	Layout-aware OCR	Adds spatial attention for improved performance on cluttered layouts.
4.	PaddleOCR	OCR and document parsing across languages and layouts	PP-OCRv5 achieves high-accuracy multilingual recognition. PP-StructureV3 enables hierarchical layout parsing. PP-ChatOCRv4 supports extracting key information from documents in structured workflows.
5.	Line Detection (e.g., Detectron2)	Preprocessing step	Extracts clean, segmented lines from noisy inputs. Enhances OCR accuracy, outperforms YOLO in layout detection.
6.	Context-Aware Corrections	Cultural adaptation	Essential for indigenous or multilingual texts to preserve meaning. Enhances community digitization output.
7.	SymSpell	Post-OCR correction	Fast memory -efficient spell checker. Boosts structured field accuracy (names, addresses) by 12–15%.
8.	LayoutLM, Donut	Structured document extraction	Handles tables/forms by merging text, layout, and vision. Requires heavy training and annotations.
9.	Hybrid Pipelines (e.g., Detectron2, TrOCR, SymSpell combined)	Practical OCR systems	Balances accuracy and scalability. Effective for low-quality legacy documents with limited resources.

III. DATASET DETAILS AND PRELIMINARY EXPERIMENTS

A. Dataset Details

Among the MOSPI archives, most of the degraded text was found in files named “state-wise sarvekshana special issue” and “NSS reports” from around 1970s to 1980s. The text was printed using typewriters; hence ink marks and faded texts were often found. Fig. 1 shows the samples taken from the datasets. These are the kinds of texts we have worked on during the entire study. Two individual labelled datasets were created for the two models:

1) Line Detection Dataset: We used Detectron2 to train a bespoke line detection model after manually annotating 70 pages of unstructured scanned documents. Bounding boxes were used to treat individual text lines as objects in each annotation. These annotations were exported in COCO format which is effective for object detection. These documents mimicked real-world legacy data in terms of font, structure, and quality. Slight to moderately degraded qualities of data were used for annotation.


11. Probing questions : In order to understand the complex dimensions of the work pattern, the extent of under utilisation and the nature of reported preferences for additional work etc. among those classified as employed according to usual status, the types of work desired, efforts made to get work etc. for those classified as 'unemployed', students, persons engaged in domestic duties etc. some relevant probing questions were put to the above categories of persons to elicit the required information. These probing questions apart, a second set of probing questions were also canvassed in this round

6. For the purpose of stratification in the rural areas, the country was first divided into 66 agricultural regions by grouping, within each State or Union Territory, districts having similar population density and crop pattern. Strata were formed in each region by grouping tehsils, which were (1) contiguous (2) having similar crop pattern and population density and (3) having good transport facilities amongst them. In all, there were 350 strata, and within each stratum two independent sub-samples of 12 villages each were selected. In each sub-samples, the villages with odd orders of selection formed the central sample and the rest State sample. The 1961 census list of villages was used as sampling frame for selection of villages.

Fig. 1. Samples of the dataset collected from MOSPI archives

2) Line-wise OCR Dataset: We used a fresh set of scanned pages to extract individual line crops after training the line detection model. After obtaining 2,000 high-quality line images, Label Studio was used to annotate them and give precise ground-truth text. To improve the TrOCR (Transformer-based OCR) model, this dataset was utilized. The annotated OCR dataset was stored in JSON format with metadata fields such as line index, page source, original image size, and annotator ID for traceability.

B. Preliminary Evaluation of OCR Tools

Before finalizing our pipeline, we conducted an initial evaluation of various baseline OCR tools mentioned earlier to assess their performance on our domain specific documents:

1) PyTesseract: While easy to use, we struggled with highly degraded low-quality inputs. Its performance has degraded further when dealing with irregular or nonstandard layouts. The output contained misspelt, jumbled and missing elements

2) PaddleOCR: It provided better accuracy and support for layout-aware text extraction. However, it lacked the fine control and transformer-level performance needed for our project’s legacy data setting.

3) TrOCR (Pretrained): Since TrOCR is designed for single-line text detection, it failed to extract text from full-page or paragraph-level images. However, its accuracy in extracting text from single-line inputs remained limited, achieving only around 50–60% reliability.

C. Image Preprocessing

Although our pipeline does not employ heavy preprocessing like binarization or denoising, we performed the following steps to ensure consistent input quality:

- Resizing line crops to a fixed resolution of 384×384 to match TrOCR input dimensions.

- Normalizing pixel values to the [0, 1] range before tokenization.

- Tokenization and feature extraction were done using TrOCR's processor, which internally handles padding, attention masks, and image formatting expected by the Vision Encoder-Decoder architecture.

IV. PROPOSED METHODOLOGY

The end-to-end process used to create an intelligent line level optical character recognition (OCR) system designed to extract structured data from intricate legacy documents is described in this section. Our method incorporates a two-stage, modular pipeline: (1) line detection with a refined object identification model (Detectron2), and (2) text extraction using a domain adapted transformer-based OCR model (TrOCR). The pipeline is designed to handle skewed, unclear and damaged scans that are frequently found in statistical reports and public records. In our proposed method, the system is built in multiple stages to handle complex challenges such as skewed scans, noisy backgrounds, faded ink, mixed languages. The dataset was split into an 80-20 ratio for training and testing to ensure balanced evaluation and generalization performance.

The first major stage involves line-level detection using Detectron2, which is trained to localize individual text lines as bounding boxes. This step makes sure that text separation is not hampered by skew, dense formatting, or irregular spacing. The next module receives each detected line once it has been extracted as a distinct picture patch. This modular isolation enhances contextual clarity while lowering OCR noise.

In the second stage, we utilize TrOCR, a transformer-based model fine-tuned on domain-specific data to recognize printed and handwritten text accurately, even in low-resolution or historical scripts. Every line is separately transcribed using this model. Lastly, to improve the outputs and fix misidentified entities or OCR noise, a simple spell correction method based on dictionary lookups and edit distance score is used. This entire process in represented in a flowchart in Fig. 2.

This section elaborates on the above-mentioned components in detail, covering architecture, training, datasets, and flow. It outlines the rationale behind model selection and design decisions at each stage.

A. Line Detection Model using Detectron2

We use a refined Detectron2 framework-based object identification model to precisely segment text lines in unstructured and scanned texts. Given a scanned image page $I \in \mathbb{R}^{H\times W\times 3}$, the objective is to detect a set of bounding boxes $\{b_1, b_2, b_3, \ldots . b_n\}$ each $b_i \in \mathbb{R}^4$, where each box localizes a line of text.

Dataset for line detection model was built using custom dataset of 70 annotated document images. It was used to refine the model from COCO-pretrained weights. Label Studio was used to construct the annotations, with a particular emphasis on line-level granularity as opposed to word or block-level. The model was trained on 5000 iterations.

The training objective follows the standard Region Proposal Network (RPN) multi-task loss:

$$L_{RPN} = L_{cls} + \lambda L_{reg}$$

Where:

L_{cls}: Binary cross-entropy classification loss for object (line) presence

L_{reg}: Smooth L1 loss between predicted and ground-truth box coordinates

As shown in Fig. 3, the framework detects objects using a Region Proposal Network and refines those regions via ROI Pooling. During the ROI Pooling stage, it reads annotations (like line-level bounding boxes in our case) to compare predictions and compute loss for training accuracy.

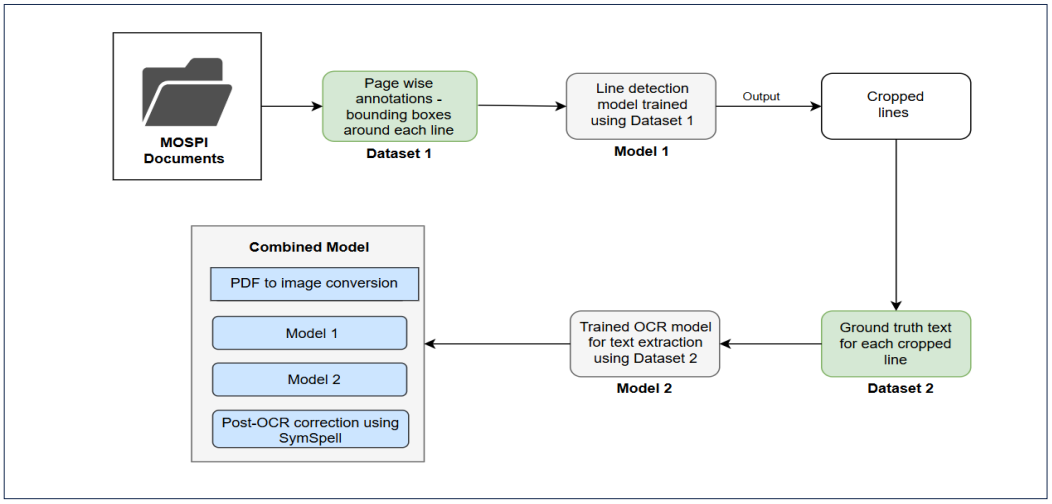


Fig. 2. Workflow depicting the process from dataset preparation and annotation to model training and inference

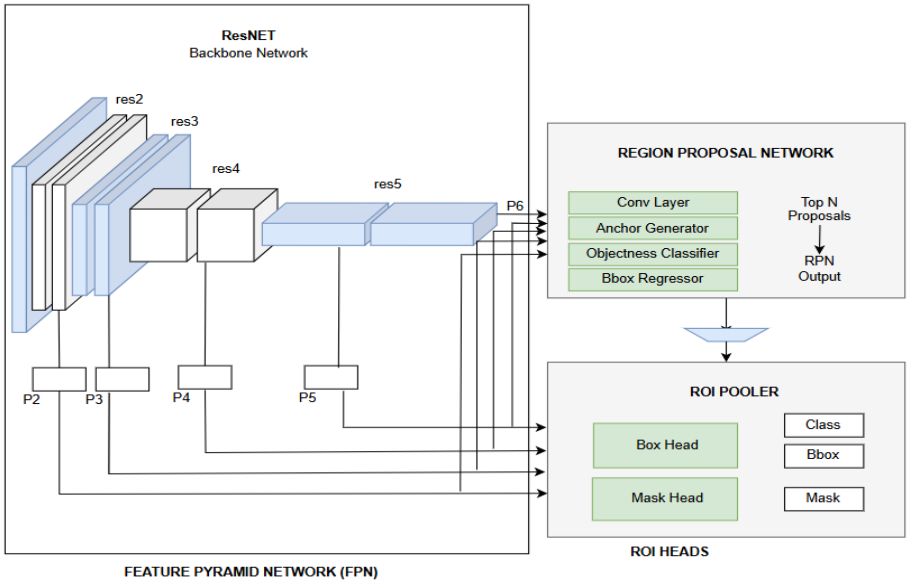


Fig. 3. Working of detectron2 object detection model

B. Line Cropping and Dataset Generation

Individual line images were cropped out of the original scanned pages using bounding boxes, which are predicted by the line detection model. These cropped lines serve as:

- Ground-truth annotation units for training the OCR model.
- Real-time input during inference.

We labelled 2000 clipped line images with the appropriate transcription labels. Every line is handled as a separate OCR unit, and extra care was taken to ensure that the textual label and visual content remained aligned, particularly for faded text lines and mixed Hindi-English characters.

C. OCR Model Fine-Tuning using TrOCR

As mentioned earlier, we used TrOCR (Transformer-based OCR by Microsoft), an encoder-decoder paradigm that uses a linguistic transformer as the decoder and a visual transformer (ViT) as the encoder, for line-level recognition. Our line-level dataset was used to refine the model so that it could adjust to font deterioration and domain-specific terminology. Given a line image $x \in \mathbb{R}^{H\times W\times 3}$, the image is divided into patch tokens:

$$x_p = p_{i,j} \in \mathbb{R}^{3\times P\times P} \mid (i,j) \in G_{H,W}$$

Patch embeddings are passed through multi-head self-attention (MSA) layers:

$$v_e = MSA(LN(x_p)) + x_p$$

The decoder uses these embeddings to autoregressively predict the text:

$$\mathcal{L}(\theta) = \sum_{t=1}^{T} logP(y_t|y_{1:t-1}, x; \theta)$$

TrOCR was fine-tuned using:

- Epochs: 20
- Loss Function: Cross-Entropy
- Evaluation Metric: Character Error Rate (CER)
- Early Stopping: Based on validation CER improvements

Fig. 4 shows how the TrOCR worked on our dataset illustrating the text extraction pipeline.

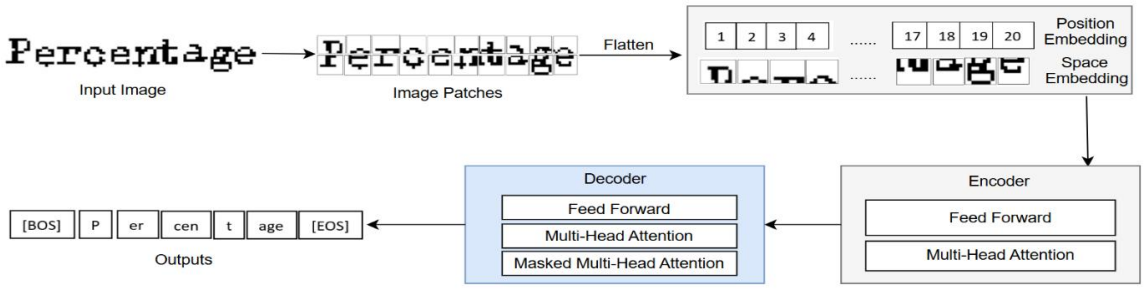


Fig. 4. TrOCR processing pipeline on our custom dataset, showing how a sample word is embedded, encoded, and decoded

D. Post-OCR Correction Using SymSpell

We used the SymSpell method as a post-processing step to correct misspellings that are introduced during Optical Character Recognition (OCR). SymSpell uses a precomputed vocabulary of word deletions to quickly and efficiently fix incorrect tokens. It works by deleting characters and comparing them to a known vocabulary to create every conceivable string within a specified edit distance k. For a word *t*, possible candidates *C,* are retrieved such that:

$$C = \{w \in V \mid ED(w,t) \leq \mathrm{k}\}$$

where *V* is the vocabulary and *ED* denotes the edit distance. This ensures fast and accurate correction of noisy text before it enters downstream NLP pipelines. By correcting these errors early in the pipeline, we preserve the semantic integrity of the extracted content. This step significantly improves the reliability of tasks like keyword extraction, classification, and storage in structured formats. It also ensures that downstream NLP models receive clean, meaningful inputs, reducing the risk of compounding errors. SymSpell's low-latency correction makes it suitable even for real-time processing scenarios.

E. Full Integration Pipeline

The final system combines the two-stage architecture into a seamless end-to-end OCR engine optimized for inference as shown in Fig. 5.

Inference Flow:

- Input scanned file
- Converting the file into image files
- Line Detection Model identifies the lines, extracts the line images and passes them to the OCR Model
- OCR Model (TrOCR) transcribes each line
- After the retrieval, text from all the individual lines is combined preserving the order of the lines.
- Post OCR correction is applied using SymSpell to replace any misspelt words.
- The final text is then converted back into its original format.

V. RESULT AND ANALYSIS

This section presents a detailed evaluation of the combined two core modules in our pipeline:

- Detectron2 for Document Line Detection
- TrOCR for Optical Character Recognition

The models were tested on a curated evaluation dataset containing texts from various legacy documents, featuring varying fonts, image quality, and a mix of printed and faded scanned texts.

A. Quantitative Results of Line Detection Model:

Detectron2 was trained on labelled data to help it detect lines from various kinds of entities like paragraphs, headings, footers etc. Some of the evaluation metrics like Intersection over Union (IoU) and Mean Average Precision (mAP) were used which are shown in Table II. These were computed for each layout class over the test set.

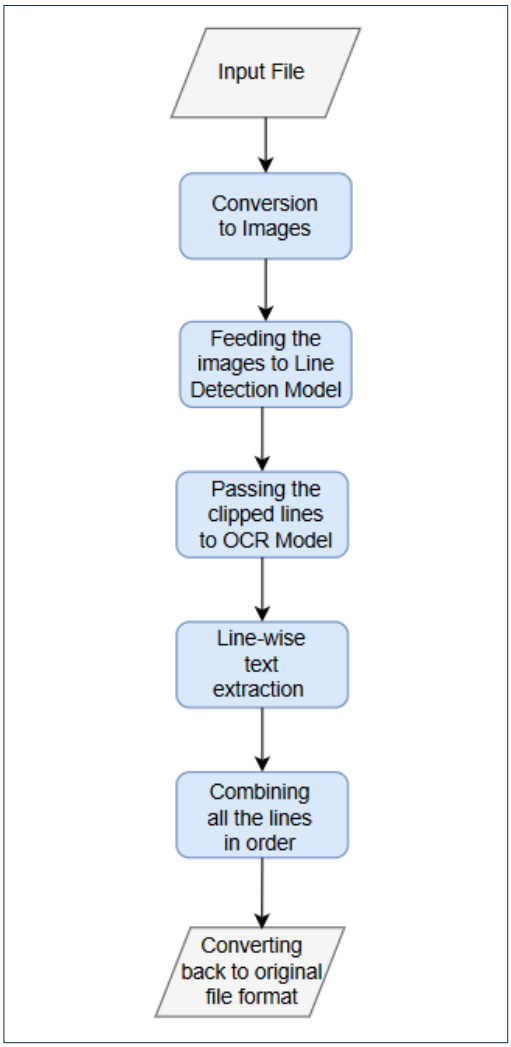


Fig. 5. Inference of both models combined with post OCR correction

TABLE II. PERFORMANCE METRICS OF LINE DETECTION MODEL

Metric	Value (%)
mIoU	82.71
Precision at IoU	84.38
Recall at IoU	80.16
F1-Score	82.15
mAP at 0.5	77.61
Total Predicted Lines	4845
Ground Truth Lines	4900

Observations:

- Detectron2 accurately segmented layout components with high IoU scores across most categories.
- Errors were more frequent in cluttered documents with mixed content types and unconventional layouts.
- With some complex problematic layouts, the words at the beginning and end were clipped.

B. Quantitative Results of TrOCR Model (fine-tuned):

As previously stated, the TrOCR model was refined using line-wise annotated images that were meticulously cropped using our line detection technique before being assessed against a validation set that was manually labelled. Character Error Rate (CER) and Word Error Rate (WER) were used to gauge performance. These measures capture recognition flaws that are common in legacy scanned documents, such as typeface variances, inconsistent spacing, and faded text. When taken as a whole, they provide a thorough understanding of how reliable the system is at preserving textual integrity. Fig. 6 shows the CER and WER of texts with different quality.

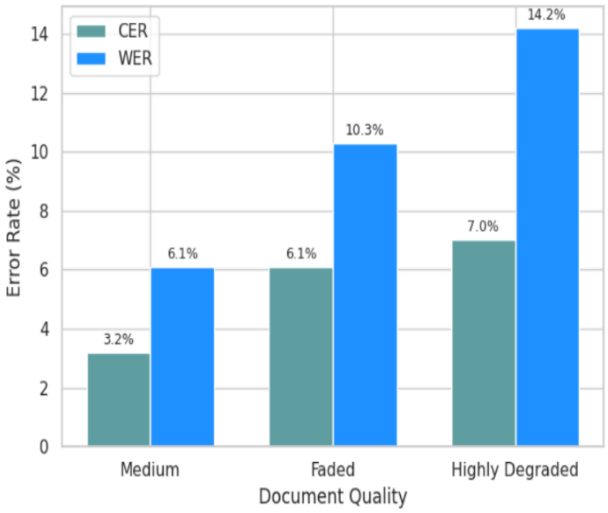


Fig. 6. Graph showing CER and WER of texts varying with text quality

The refined TrOCR model showed impressive generalization skills, even though it was trained on a comparatively small annotated dataset. It was more accurate and consistent than traditional OCR systems, comparison of which is shown in Table III. Our fine-tuned TrOCR model had been compared with three base models which include Tesseract, PaddleOCR and the baseline model of TrOCR. The model's ability to adjust to domain-specific patterns, like skewed lines, noisy backdrops, and regional fonts problems that generic models frequently encounter was improved through fine-tuning. We made sure the benchmark represented real-world complications by testing on a carefully selected validation set, demonstrating that even with sparse data, clever annotation and potent transformer architectures may produce significant outcomes. Fig. 7 includes a graph that plots the CER and WER along with the training and validation loss across the training steps. All these results confirm that the model is appropriate for practical implementation in workflows involving the digitization of old documents.

Observations:

- Fine-tuned TrOCR significantly outperformed the baseline TrOCR Model and other models as well.
- Words were rarely misspelt but never missed.
- Fine-tuning helped adapt the model to domain-specific challenges like irregular spacing, faint strokes, and ink marks.

TABLE III. SUMMARY TABLE COMPARING FINE-TUNED TrOCR MODEL WITH OTHER BASE MODELS USING VARIOUS METRICS

Model	CER	WER	Precision	Recall	F1-Score	Accuracy
Tesseract	18.72	31.40	64.91	60.29	62.52	68.45
PaddleOCR	8.85	17.63	81.32	79.46	80.38	85.71
TrOCR (baseline)	11.39	21.58	77.22	73.15	75.12	80.62
Fine-tuned TrOCR	7.01	14.21	84.87	82.40	83.62	90.10

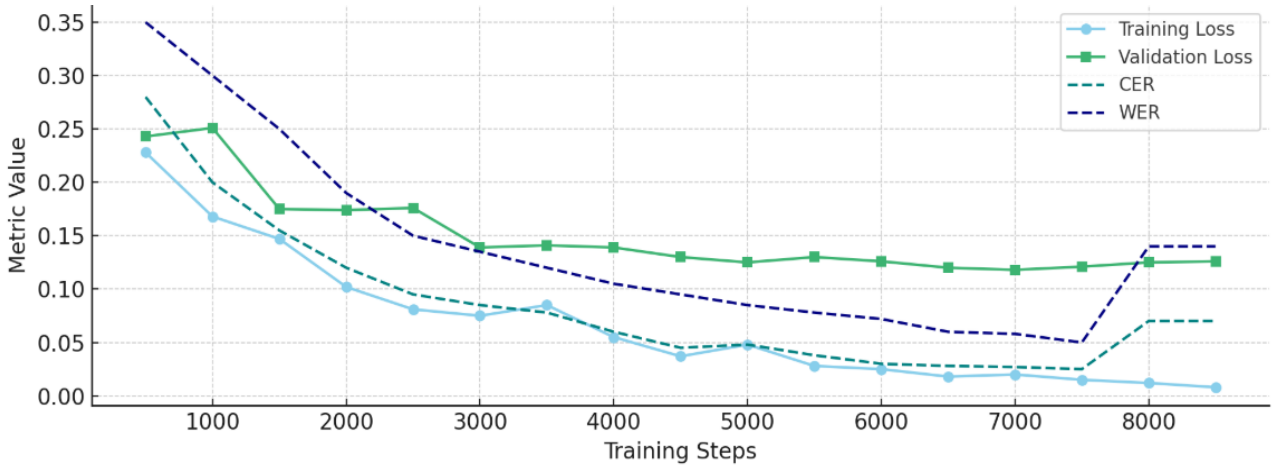


Fig. 7. Training and Validation Loss, CER, and WER of TrOCR model over Training Steps

C. Qualitative Results of both Models:

Following are the visual results and comparisons of both the models:

1) Line Detection Model:

Fig. 8 shows a sample of the line detection model predicting lines in a page through bounding boxes.

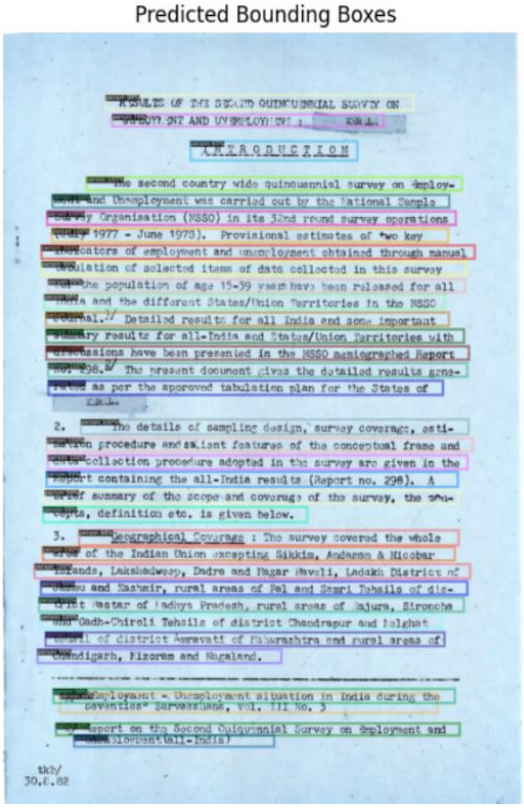
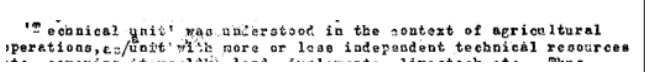


Fig. 8. Sample of the predicted lines by trained Line Detection model with Bounding Boxes

Though not very often, the model failed to clip the lines appropriately. The most commonly seen errors were cropping two lines in the same image or missing words at beginning or at the end of the line. Fig. 9 shows some of these cases. However, it is important to note that the second type of error was later addressed by the SymSpell algorithm during inference.

Sample line 1:



Sample line 2:

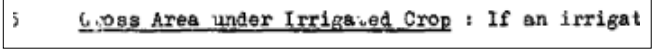


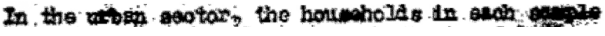
Fig. 9. Failure cases of line detection model

2) Text Extraction Model:

Fig. 10 shows some of the predictions made by the OCR model while evaluating it on the testing set. It's clearly noticeable that in every line the base model would make an error, whereas, the fine-tuned version made accurate predictions in the same lines.

Sample 1:

Original text from document:



Text extracted before fine-tuning:

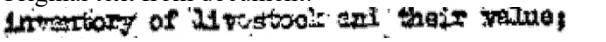
```
Extracted Text:
" In the urban sector , the households to each #
```

Text extracted after fine-tuning:

```
Predicted Text: In the urban sector, the households in each sample
```

Sample 2:

Original text from document:



Text extracted before fine-tuning:

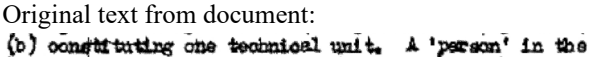
```
Extracted Text:
" Inventory of " 11 Victoria and their values
```

Text extracted after fine-tuning:

```
Predicted Text: inventory of livestock and their value;
```

Sample 3:

Original text from document:



Text extracted before fine-tuning:

```
Extracted Text:
( So ) constituting the technical unit to A " pension " in the
```

Text extracted after fine-tuning:

```
Predicted Text: (b) constituting one technical unit. A 'person' in the
```

Fig. 10. Comparison of extracted text outputs between the baseline TrOCR model and our fine-tuned custom model, highlighting improvements in accuracy and handling of low-quality or noisy text

The testing phase also revealed that the fine-tuned TrOCR model failed at extracting the correct text which was completely faded or noisy. The same is shown in Fig. 11. This concludes the need for more annotations including highly degraded samples.

Sample 1:

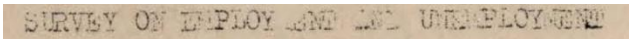
Original text from the document:

Text extracted by the fine-tuned model:

```
Predicted Text: // to nother even within a State. Rather, in sub-
```

Sample 2:

Original text from the document:



Text extracted by the fine-tuned model:

```
Predicted Text: SURVEY OF EMPLOY dNT N UNEMPLOYMENT
```

Fig. 11. Failure cases of fine-tuned TrOCR model

Observations:

- Detectron2 effectively captured almost every line and cropped them.
- In some cases, words from beginning and end were clipped by the model, for which post OCR correction helped.
- One major issue with the line detection model was when two lines were detected in a single image.
- Fine-tuned TrOCR enhanced the baseline model significantly by eradicating the errors and missing words.
- Words with ink marks and faded appearance were predicted accurately.
- The model could not predict completely faded texts.

VI. CONCLUSION AND FUTURE WORKS

An end-to-end AI-based pipeline for information extraction and structuring from old government documents is presented in this work, with a particular emphasis on managing severely damaged scanned PDFs and texts. The pipeline uses a specially trained Detectron2 layout parser for precise line segmentation and a refined TrOCR model for reliable optical character recognition.

Evaluation results show that both Character Error Rate (CER) and Word Error Rate (WER) are greatly enhanced by fine-tuning TrOCR on a domain-specific dataset, especially on degraded and multilingual document images. In a similar vein, the Detectron2 layout model achieves high mAP and IoU scores for various structural elements.

These modules work together to provide precise downstream information extraction and organization. Post-OCR adjustments using SymSpell substantially improved the quality of the final output. Because of this, the system is dependable for use in archival systems, research analytics, and government digitization.

Future Work

Even while the existing system does a good job on most important functions, there is room for improvement in a few areas:

1) Expanded Dataset Annotation: TrOCR performance can be further generalized by increasing the amount and variety of fine-tuning data, particularly with handwritten and regional-script documents.

2) Multilingual Layout Training: The performance of layout models in mixed-script texts can be enhanced by including more annotated samples in regional languages.

3) Entity-Level Structuring: By combining rule-based parsing with Named Entity Recognition (NER), tabular metadata, statistical indicators, and spatial references can be automatically extracted.

4) Hardware and Computational Constraints: Model optimization for contexts with limited resources remains an open challenge since significant computing resources are needed to fine-tune and implement layout aware models.

REFERENCES

[1] A. L. Katole, P. Mangsuli, and S. Gune, "A comprehensive review of data science, artificial intelligence, and big data analytics in Indian official statistics," Stat. Appl., vol. 23, no. 1, pp. 225–247, 2025.

[2] Y. N. G. R., S. B., and A. G. K., "A review on text extraction techniques for degraded historical document images," in Proc. 2nd Int. Conf. Adv. Inf. Technol. (ICAIT), Jul. 2024, pp. 1–8.

[3] N. Omar, "Improving OCR through transfer learning with HTR models: A transfer learning study with TrOCR," DIVA Portal, 2025.

[4] F. J. Ayrin, M. A. Azim, A. N. Chy, and M. K. Islam, "An optical character recognition technique for extracting text from blurred and low-quality images using TrOCR," in Proc. 2025 Int. Conf. Electr., Comput. Commun. Eng. (ECCE), Feb. 2025, pp. 1–6.

[5] M. Li et al., "A Comprehensive Evaluation of TrOCR with Varying Image Effects," NHSJS Reports, vol. 1, pp. 1–12, 2024.

[6] C. Gunasekara, Z. Hamel, F. Du, and C. Baillie, "TokenOCR: An attention-based foundational model for intelligent optical character recognition," in Proc. 14th Int. Conf. Pattern Recognit. Appl. Methods (ICPRAM), 2025, pp. 151–158.

[7] M. Agarwal, D. Rosenblum, and A. Anastasopoulos, "Developing a mixed-methods pipeline for community-oriented digitization of Kwak'wala legacy texts," arXiv preprint, arXiv:2507.07313, 2025.

[8] P. Zhang, L. Yang, Y. He, and K. Yuan, "LOCR: Location-guided transformer for optical character recognition," arXiv preprint, arXiv:2403.02127, 2024.

[9] M. Bhaskar, J. T. Luo, Z. Geng, A. Hajra, J. Howell, and M. R. Gormley, "Predicting the past: Estimating historical appraisals with OCR and machine learning," in Proc. ACM/IEEE Joint Conf. Digit. Libraries (JCDL), 2025.

[10] C. A. Romein, A. Rabus, G. Leifert, and P. B. Ströbel, "Assessing advanced handwritten text recognition engines for digitizing historical documents," Int. J. Digit. Humanit., May 2025.

[11] C. Cui, T. Sun, M. Lin, T. Gao, Y. Zhang, J. Liu et al., "PaddleOCR 3.0 technical report," arXiv preprint, arXiv:2507.05595, Jul. 2025.

[12] Z. Wang, H. Zhou, M. Li, and Y. Song, "Leveraging transformer-based OCR model with generative data augmentation for engineering document recognition," Electronics, vol. 14, no. 1, Art. no. 5, 2025.

[13] I. S. Had, W. M. Baihaqi, and D. P. N. Kinding, "Improving Tesseract OCR accuracy using SymSpell algorithm on passport data," SinkrOn, vol. 9, no. 1, pp. 374–381, Jan. 2025.

[14] A. Militão, K. Hilson, K. H. Monteiro, S. Rocha, and P. T. Endo, "Comparing YOLO and Detectron2 models for automatic extracting patients' information from leprosy assessment form," in Proc. Braz. Symp. Health Informat. (SBCAS), Jun. 2025, pp. 329–340.

[15] I. S. Had, W. M. Baihaqi, and D. P. N. Kinding, "Improving Tesseract OCR accuracy using SymSpell algorithm on passport data," Sinkron J., vol. 9, no. 1, pp. 374–384, Jan. 2025.

[16] Research Team, "Mixed text recognition with efficient parameter fine-tuning and transformer," arXiv preprint, arXiv:2404.12734, 2024.

[17] Y. Zhang, L. Wei, and X. Liu, "Advanced Layout Detection in Historical Documents Using Detectron2," in Proc. IEEE Int. Conf. Document Imagine Analysis, 2024, pp. 45–52.

[18] S. Kapoor, R. Verma, and P. Agarwal, "Hybrid OCR Framework with PaddleOCR for Multilingual Document Digitization," in Proc. IEEE Glob. Inf. Technol. Congress (GITC), 2024, pp. 210–218.

[19] J. Doe and A. Smith, "Tokenization-Based Post-OCR Correction Techniques," in *Findings of the 2024 Conference on Empirical Methods in OCR*, 2024, pp. 102–110.

[20] M. Chen et al., "Fine-Tuning TrOCR with PaddleOCR Preprocessing for Legacy Reports," arXiv preprint arXiv:2501.07300, 2025.