OCRTurk: A Comprehensive OCR Benchmark for Turkish

Deniz Yılmaz[1], **Evren Ayberk Munis**[2], **Çağrı Toraman**[1],
Süha Kağan Köse[3], **Burak Aktaş**[3], **Mehmet Can Baytekin**[3], **Bilge Kaan Görür**[3]
[1]Middle East Technical University, Computer Engineering Department, Turkey
[2]Politecnico di Torino, Italy
[3]Roketsan Inc., Artificial Intelligence Technologies Unit, Turkey
deniz.yilmaz_12@metu.edu.tr, evrenayberk.munis@studenti.polito.it
ctoraman@metu.edu.tr, kagan.kose@roketsan.com.tr, burak.aktas@roketsan.com.tr
can.baytekin@roketsan.com.tr, kaan.gorur@roketsan.com.tr

Abstract

Document parsing is now widely used in applications, such as large-scale document digitization, retrieval-augmented generation, and domain-specific pipelines in healthcare and education. Benchmarking these models is crucial for assessing their reliability and practical robustness. Existing benchmarks mostly target high-resource languages and provide limited coverage for low-resource settings, such as Turkish. Moreover, existing studies on Turkish document parsing lack a standardized benchmark that reflects real-world scenarios and document diversity. To address this gap, we introduce OCRTurk, a Turkish document parsing benchmark covering multiple layout elements and document categories at three difficulty levels. OCRTurk consists of 180 Turkish documents drawn from academic articles, theses, slide decks, and non-academic articles. We evaluate seven OCR models on OCRTurk using element-wise metrics. Across difficulty levels, PaddleOCR achieves the strongest overall results, leading most element-wise metrics except figures and attaining the best Normalized Edit Distance scores in easy, medium, and hard subsets. We also observe performance variation by document type: models perform well on non-academic documents, while slideshows become the most challenging.

1 Introduction

Optical Character Recognition (OCR) is a technology that enables the extraction of text, tables, figures, and other structural elements from images of handwriting (Memon et al., 2020), receipts (Huang et al., 2019; Park et al., 2019), scenes (Munjal et al., 2021; Lunia et al., 2023), documents (Ouyang et al., 2024), and similar sources. Similarly, besides extracting raw text, extracting different elements is crucial for real-world applications. Table extraction (Anand et al., 2023; Patel, 2025; Pallavi et al., 2020) and equation extraction (Zhong et al., 2025) are among the most popular element-wise extraction tasks in document understanding. This technology bridges the gap between image-based content and computer-based processing, making text accessible for downstream applications, including large language models (LLMs).

Since the outputs of OCR models are used in daily life and as data for training machine learning systems, the reliability of these outputs is a very important issue. To address this and to observe model performance, several benchmarks are used (Ouyang et al., 2024; Poznanski et al., 2025; Fu et al., 2024). These benchmarks measure the capability of models to extract text in the correct format and reading order, and to recognize tables, mathematical formulas, and figures called Document Parsing in total and understanding of key information of the documents. However, most existing benchmarks are in English, which makes it difficult to evaluate model performance in low-resource languages, such as Turkish. Turkish morphology, flexible word order, and variation across sources introduce challenges not reflected in English benchmarks (Hakkani-Tür et al., 2002; Oflazer, 2014; Umutlu et al., 2025).

From a Turkish perspective, although studies on text recognition increase over the years (G. Öztürk et al., 2025), there are still only a limited number of datasets and a single OCR Document Parsing benchmark in Turkish (Yılmaz et al., 2025). This benchmark evaluates model performance in Turkish character confusion, word length effects, context-dependent errors, and recognition under distortions, using synthetic Turkish data. However, this benchmark only evaluates the model's performance on raw text, does not include elements, such as figures, tables, or mathematical equations, and uses synthetic data derived from a single source. For this reason, there is a clear gap in benchmarks that assess model performance beyond raw text and under realistic conditions.

In this work, to fill this gap, we introduce the first Turkish OCR benchmark[1] designed to reflect real-world document diversity, difficulty, and structure. The benchmark contains 180 document pages in PDF format from articles, theses, non-academic documents, and slideshows. These documents come from various sources shared on GitHub and include tables, mathematical equations, and figures. These documents are divided into three difficulty levels: easy, medium, and hard based on the complexity of document structure.

Using this benchmark, it is possible to evaluate:

Raw text analysis: The model's capability to recognize Turkish characters and extract raw text correctly.

Table recognition: The model's ability to reproduce tables in the correct format and with accurate content.

Mathematical equation recognition: The capability to reproduce formulas and equations from document images.

Figure recognition: The capability to extract figures correctly from document images.

Compared with earlier Turkish benchmark, we make three main contributions. First, we move beyond raw-text–only evaluation and build a benchmark that reflects real-world documents from diverse sources, including figures, tables, and equations. Second, we publicly release the dataset and evaluation scripts so that researchers can test their models on many practical scenarios before using them on real documents. We show that model performance differs across document types and difficulty levels. PaddleOCR achieves the best overall results. In addition, models perform the best on non-academic documents and the worst on slideshows.

2 Related Work

Document Parsing Benchmarks Document parsing is an important aspect of OCR, as it measures the capability of models to extract different structural elements from documents. OmniDocBench (Ouyang et al., 2024) is a benchmark that contains 981 PDF pages across nine distinct document types and evaluates raw text correction, table recognition, formula recognition, and reading order. It also uses adjacency-search matching to reduce the impact of paragraph splitting and an ignore-handling strategy to exclude parts of the text, such as headers and footers, in order to obtain more consistent metrics. Similarly, OlmOCR-Bench (Poznanski et al., 2025) is a widely used benchmark. Like OmniDocBench, it measures text presence and absence, natural reading order, table accuracy, and mathematical formula accuracy. It includes a total of 7,010 test PDF pages across these categories. All tests follow a pass/fail format, and the overall score is computed as the mean over all test categories. The CC-OCR (Yang et al., 2024) benchmark is designed for different real-world scenarios. It includes tasks for multi-scene OCR, multilingual OCR, document parsing, and key information recognition. KITAB (Heakl et al., 2025) is an Arabic OCR benchmark with 8,809 samples across nine domains. It includes tasks, such as table recognition and PDF-to-Markdown conversion, and it provides a strong example of an OCR benchmark for a low-resource language. In real-world document parsing, some documents contain photographs taken from different angles, which earlier benchmarks do not explicitly address. DocPT-Bench (Du et al., 2025) is introduced to fill this gap and evaluates document parsing abilities for various real-world cases, including digital-born documents, photographed documents, and unwrapped photographed documents.

Turkish Document Parsing Turkish document parsing studies are mainly grouped into two categories: Handwriting parsing and digital character parsing. In handwriting studies, some work uses Turkish handwritten characters that are created in computer environments (Kuncan et al., 2020; Al-Zubaidi et al., 2019). In these studies, characters are generated by using a mouse on a computer, and the handwritten characters are classified using Artificial Neural Networks (ANNs). Kizilirmak (2022) studies the same problem using Convolutional Neural Networks (CNNs) and BiLSTM architectures, and also introduces a new dataset that contains 2,600 neutral handwriting line images collected from 73 participants. On the other hand, Şevik (2019) studies Turkish character recognition in digital texts with different fonts. This work uses 13,000 Turkish letters with 38 different fonts, and classifies them with CNN-based models. Similarly, there are several studies that focus on non-handwritten documents. NacsoftOCR (Sayallar et al., 2023) is an example of Turkish receipt recognition, while Oral

[1] https://github.com/metunlp/ocrturk

et al. (2020) studies information extraction from bank documents, such as money transfer orders.

Turkish Document Parsing Datasets and Benchmarks There are Turkish LLM benchmarking efforts that involve RAG evaluation (Toraman et al., 2026). However, they are limited in this particular task.

THE dataset (Bartos et al., 2020) is a multilingual handwritten character dataset that contains Turkish, Hungarian, and English characters. The dataset is collected from 200 participants and includes 15,600 binary character images corresponding to 78 unique letters.

Türkçe Kitap (Zeer et al., 2024) is a dataset constructed from a collection of 100,000 books. It contains images taken from these books, as well as conversations between large language models and humans about the visual content.

To the best of our knowledge, the only benchmark in Turkish for OCR and vision-language models is the work by Yılmaz et al. (2025). They introduce a synthetic Turkish dataset that includes 6,600 images with printed, handwritten, and scene text. They use Mustafa Kemal Atatürk's book, Nutuk, as the text source, and place selected words and sentences onto background images taken from the COCO dataset (Lin et al., 2014) after removing all textual content from the images for the scene-text setting. They use the Text Recognition Data Generator[2] to create both handwritten and printed text. They evaluate the models with respect to Turkish character confusion, word-length effects, context-dependent errors, and distortion type, and they use Character Error Rate (CER) and Word Error Rate (WER) as evaluation metrics.

3 Dataset

Categorization of Data We categorize the dataset into four categories: academic documents, non-academic documents, theses, slideshows. While academic documents are only from arxiv[3] and DergiPark[4] non-academic documents include financial reports, course materials, manuals, and annual summary reports from various sources (which are available at the JSON files of each data in the dataset). All theses are from YÖK TEZ[5] and the slideshows are from MEB OGM Materyal[6] and Ankara Üniversitesi Açık Öğretim Materyali[7]. We generate equal amounts of data for each category (45 pages of data for each category).

Classification of Data We classify every data in the dataset as one of the three difficulty levels: easy, medium, and hard. We classify the documents that include only texts as easy. We classify the documents as medium if they include both texts and one of the following items once: one-line equation, tables without multi-columns or multi-rows, figures without sub-figures. Otherwise, we classify the data as hard. We generate equal amounts of data for each difficulty (60 pages of data for each difficulty).

Data Structure Observing that the OCR models return the outputs in Markdown format, we adopt this format in our dataset to standardize the data gathered from various sources. This allows us to use HTML format for the tables, LaTeX format for the equations, and PNG format for the figures. To generate statistics for the dataset, we store the tables, equations, and images in separate folders for each data.

Data Generation The benchmark is constructed from Turkish documents collected from multiple sources. The dataset is split into two disjoint subsets. Two annotators each converts one subset of the original documents into a unified Markdown representation. Prior to conversion, repetitive headers and footers are cropped from the original PDFs. This is done because a large portion of models do not take them into account while generating responses, and to ensure that the strengths of each model are evaluated fairly. This method is also adopted in (Ouyang et al., 2024). During this conversion, plain text is transcribed verbatim, tables are converted into HTML format, and mathematical expressions are converted into LaTeX. ChatGPT and Gemini are used as assistive tools during this conversion step to accelerate formatting, but all outputs are manually verified.

Following the conversion, each annotator performs a character-level manual verification of their own converted Markdown files against the original documents. This includes checking textual content, table structure, and mathematical expressions to ensure faithful transcription. Discrepancies were corrected manually.

[2]https://github.com/Belval/TextRecognitionDataGenerator
[3]https://arxiv.org/
[4]https://dergipark.org.tr/tr/
[5]https://tez.yok.gov.tr/UlusalTezMerkezi/
[6]https://ogmmateryal.eba.gov.tr/
[7]https://acikders.ankara.edu.tr/

Category	Difficulty			Total
	Easy	Medium	Hard	
Academic Docs.	15	15	15	45
Non-academic Docs.	15	15	15	45
Theses	15	15	15	45
Slideshows	15	15	15	45
Total	60	60	60	**180**

Table 1: The table summarizes the number of pages of data for each category and difficulty level. The totals in each category and in each difficulty level is given. The subtotal is given in the bottom-right corner in bold.

		Items			Total
		Equations	Tables	Figures	
Category	**Academic Docs.**	42	42	22	106
	Non-academic Docs.	0	45	16	61
	Theses	42	7	16	65
	Slideshows	8	36	3	47
	Total	**92**	**130**	**57**	**279**
Difficulty	**Easy**	0	0	0	0
	Medium	11	51	23	85
	Hard	81	79	34	194
	Total	**92**	**130**	**57**	**279**

Table 2: The table summarizes the total number of items (equations, tables, and figures) in each category and for each difficulty level.

As a quality control step, only samples that passes manual character-level consistency checks are included in the final benchmark. This manual checking step ensures strong consistency between the source documents and the benchmark annotations. Since the data are manually generated and carefully inspected character by character, the data construction process is highly time-consuming; as a result, the benchmark currently consists of 180 pages of data.

Summary The number of pages of data in each category and difficulty level and their totals are explained in Table 1. The total number of items (equations, tables, and figures) in each category and in each difficulty level and their respective totals are explained in Table 2.

4 Methodology

We evaluate the model accuracy based on how correctly they reproduce the texts, tables, equations, and figures in the given documents. To do this, we first extract the items (equations, tables, and figure tags) from the Markdown, to be used for the evaluations metrics afterwards, and end up with raw texts. Then we apply a number of post processing steps on the raw texts. We convert the misprinted Turkish characters. For example, we convert ˘ g (breve symbol followed by the letter g) to ğ, and ¸ s (cedilla symbol followed by the letter s) to ş. We remove the title, subtitle, etc. tags (#, ##, **). After these post-processing steps, we continue with the calculation of the performance scores with the evaluation metrics using the extracted items and cleaned raw texts.

4.1 Evaluation Metrics

4.1.1 Texts

For texts, we use two metrics: Normalized edit distance (NED) and Turkish character sensitivity (TCS). In *normalized edit distance*, we calculate the number of edits that we should apply to the model output to achieve the same result as the ground truth. Formally defined as

$$\text{NED} = \frac{d_{edit}(H, R)}{max\{|H|, |R|\}}$$

where $d_{edit}(H, R)$ shows the Levenshtein Distance between the model output H and the ground truth R. To obtain a number between 0 and 1 for the score, we divide this distance by the maximum of the number of characters in the model output and in the ground truth. While lower scores represent a higher similarity between the model output and the ground truth, higher scores indicates that a number of edits should be applied to the model output to achieve the same result as the ground truth. This metric is used in OmniDocBench. In *Turkish character sensitivity*, we try to measure how accurately models reproduce characters specific to Turkish (ç, ğ, ı, ö, ş, ü, Ç, Ğ, İ, Ö, Ş, Ü). To achieve this, we calculate the ratio of errors to the total number of characters specific to Turkish. Formally defined as

$$\text{TCS} = 1 - \frac{E}{N}$$

where E and N represents the number of errors and the total number of characters specific to Turkish, respectively. We subtract this index from 1 to obtain a score where a higher value indicates a better performance (e.g. 0 errors, $E = 0$, will return a score of 1).

4.1.2 Tables

For tables, we use two metrics: Tree edit distance based similarity (TEDS) and normalized edit distance (NED). In *Tree edit distance based similarity*, we first convert the tables to tree structures, then calculate the number of edits that we should apply to the tree generated from model output to achieve the same result as the tree generated from the ground truth. Formally defined as

$$\text{TEDS} = 1 - \frac{d_{edit}(T_H, T_R)}{max\{|T_H|, |T_R|\}}$$

where $d_{edit}(T_H, T_R)$ shows the tree edit distance between the tree generated from the model output T_H and the tree generated from the ground truth. Similarly to NED, we divide this by the maximum of the number of nodes in the trees generated from the model output and the ground truth. Since we measure the similarity score, we subtract this fraction from 1. Thus, higher scores represent a higher similarity between the tables. In *normalized edit distance*, we adopt a similar approach to the NED for texts. The difference is that for this metric, we consider the content of the table, such as cell values and table tags. Lower scores in this metric represent a higher similarity.

4.1.3 Equations

For equations, we use three metrics: Bilingual evaluation understudy score (BLEU), character detection matching (CDM), and normalized edit distance (NED). In *Bilingual evaluation understudy score*, we measure the overlap between the model output equation and the ground truth equation, both of which are in LaTeX format. Essentially, it's an n-gram based metric that evaluates the ratio of the symbols produced in the correct order. Formally defined as

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

where BP penalizes the model output for being shorter than the ground truth, brevity penalty. p_n, w_n, and N refers to the n-gram precision, weights for every n-gram (usually $w_n = 1/N$, and the maximum n-gram length (e.g. $N = 4$), respectively. Since the ordering of symbols is critical for correct equation interpretation, BLEU is used to assess the model's ability to generate syntactically correct expressions. This metric is used in OmniDocBench. In *normalized edit distance*, we adopt a similar approach to the NED for texts. While NED measures the structural similarity of the equations, it also takes the minor symbol differences into account. It's used to detect the character level errors especially for long and/or symbolically crowded equations (e.g. integrals, matrices). In *character detection matching*, we measure how precisely the models recognize the characters in equations. True positives, false negatives, and false positives are evaluated together. Formally defined as

$$\text{CDM} = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FP} + \text{FN}}$$

where TP, FN, and FP represent true positives, false negatives, and false positives, respectively. It reflects the performance drop of models precisely especially when the models misinterpret the symbols like subscripts ($_2$), superscripts (^{2}), and special symbols (σ). This metric is used in OmniDocBench.

4.1.4 Figures

For figures, we use two metrics: Mean squared error (MSE), and DreamSim's evaluation score (DS) (Fu et al., 2023). We use *mean squared error* to quantify the average squared difference between the model output and the ground truth image. Formally defined as

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(R'_i - H'_i)^2$$

where N is the total number of elements (pixels $\times$ channels) in the images. H'_i and R'_i represent the i^{th} element in the images H' and R', respectively. We obtain the images H' and R' by normalizing the pixel values of the original model output H and the ground truth R to $[0, 1]$ range. Lower MSE scores indicate lower average squared differences between the model output and the ground truth image, showing a higher similarity between them. We adopt the *DreamSim evaluation metric*, a deep learning-based similarity metric, to evaluate perceptual quality, as recent studies indicate that it correlates highly with human visual assessment and is robust to cropped images compared to other metrics. (Wickrema et al., 2025).

4.2 Experimental Setup

In this paper, we evaluate OlmOCR2[8], DeepSeek-OCR[9], NanonetsOCR2[10], PaddleOCR[11], Docling[12], and NVIDIA Nemotron v1.1[13],

[8]https://huggingface.co/allenai/OlmOCR2-2-7B-1025
[9]https://github.com/deepseek-ai/DeepSeek-OCR
[10]https://huggingface.co/nanonets/Nanonets-OCR2-3B
[11]https://github.com/PaddlePaddle/PaddleOCR
[12]https://github.com/docling-project/docling
[13]https://huggingface.co/nvidia/NVIDIA-Nemotron-Parse-v1.1

HuanyanOCR[14] Parse using the proposed benchmark.

DeepSeekOCR We run DeepSeekOCR inference using the Transformers model.infer method, where the base size is set to 1,024 and the image size is set to 640. The crop mode and test compression options are enabled, while all other inference parameters are kept at their default values.

DoclingOCR For DoclingOCR, we use Docling's official library and perform document conversion through the DocumentConverter, specifying PDF processing via the PdfFormatOption with customized pipeline options. In particular, the image scale is set to 2, and both page image generation and picture image generation are enabled. All other pipeline and conversion parameters are kept at their default values.

PaddleOCR For PaddleOCR, we use the official PaddlePaddle OCR library and perform inference with the PaddleOCRVL using its default configuration.

HuanyanOCR For HuanyanOCR, we perform text generation using the Hugging Face Transformers library, where the maximum number of new tokens is set to 16,384 and sampling is disabled. All other generation parameters are kept at their default Transformers settings.

Nanonets OCR2 For Nanonets OCR, we perform text generation using the Hugging Face Transformers library, where the maximum number of new tokens is set to 15,000 and sampling is disabled. All other generation parameters are left at their default Transformers values.

Nvidia Nemotron v1.1 For NVIDIA OCR, we perform text generation using the Hugging Face Transformers library with a customized configuration. Specifically, the beginning-of-sequence token identifier is set to 0, the decoder start token identifier and end-of-sequence token identifier are both set to 2, the forced end-of-sequence token identifier is set to 2, and the padding token identifier is set to 1. In addition, the maximum number of new tokens is set to 9,000, sampling is disabled, beam search is performed with a single beam, and a repetition penalty of 1.1 is applied. All other generation parameters are left at their default Transformers values.

[14]https://huggingface.co/tencent/HunyuanOCR

OlmOCR2 For OlmOCR2, we perform text generation using the Hugging Face Transformers library, where the temperature is set to 0.1, the maximum number of new tokens is set to 15,000, a single output sequence is generated, and sampling is enabled. All other generation parameters are kept at their default Transformers settings.

All model parameters are adopted from the official example usages provided in the models' Hugging Face pages or GitHub repositories. This choice is motivated by the goal of following developer-recommended configurations to obtain the best possible outputs under standard settings. All outputs are generated in the medium of L4 22.5 GB GPU and A100 40 GB GPU.

5 Results

After obtaining the scores of the OCR models for each evaluation metric within 180 pages of data, we calculate metrics of these with all data. We give the detailed results in Table 3.

5.1 Element Based Results

Raw Texts PaddleOCR, HuanyanOCR, and OlmOCR2 lead the raw-text evaluation with HuanyanOCR achieving the highest TCS score, indicating that it performed best with Turkish charaters. They achieve low Normalized Edit Distance and high Turkish Character Sensitivity , indicating that their outputs require minimal edits and that they better preserve Turkish-specific characters. While Docling attains a comparable NED, its lower TCS suggests weaknesses in handling Turkish characters. Nvidia Nemotron v1.1 performs worst in this section (NED = 0.27, TCS = 0.47), indicating substantial errors and limited sensitivity to Turkish characters.

Equations PaddleOCR remains the strongest on the equation benchmark, consistent with its raw-text performance. NanonetsOCR2 joins the top group and leads overall, achieving the best scores across all three metrics (NED = 0.05, BLEU = 0.94, CDM = 0.95). Although DeepSeekOCR falls behind PaddelOCR and NanonetsOCR2, it shows a relatively good performance. The remaining models cluster in a mid-range performance band, with NED in the 0.11–0.15 range and BLEU/CDM between 0.85 and 0.89, indicating solid equation transcription quality but a clear gap to the leaders.

Tables Most models perform very well on table extraction, with NED values around 0.05 and

OCR Model	Raw Texts		Equations			Tables		Figures	
	NED ↓	TCS ↑	NED ↓	BLEU ↑	CDM ↑	NED ↓	TEDS ↑	MSE ↓	DS ↓
DeepSeekOCR	0.12	0.81	0.08	0.91	0.92	**0.05**	**0.87**	**0.04**	**0.06**
Docling	0.13	0.71	0.15	0.85	0.85	**0.05**	0.86	0.06	0.12
PaddleOCR	**0.08**	0.82	0.06	**0.94**	**0.95**	**0.05**	**0.87**	0.05	0.10
HuanyanOCR	0.09	**0.88**	0.11	0.89	0.89	0.06	0.85	-	-
NanonetsOCR2	0.17	0.79	**0.05**	**0.94**	**0.95**	0.11	0.80	-	-
Nvidia Nemotron v1.1	0.27	0.47	0.14	0.85	0.86	0.07	0.86	-	-
OlmOCR2	0.09	0.80	0.12	0.87	0.88	**0.05**	0.85	-	-

Table 3: The summary of the average scores of the models DeepSeekOCR, Docling, PaddleOCR, HuanyanOCR, NanonetsOCR2, Nvidia Nemotron v1.1, and OlmOCR using the evaluation metrics NED (normalized edit distance), TCS (Turkish character sensitivity), BLEU (bilingual evaluation understudy), CDM (character detection matching), TEDS (tree edit distance based similarity), MSE (mean squared error), and DS (DreamSim's evaluation score). The upward arrows (↑) next to the metric abbreviations indicate a better performance if the value is higher, while the downward arrows (↓) indicate a better performance if the value is lower. The best result(s) within each metric is written in bold, accordingly. We indicate the values for the image metric columns with dashes if the models do not return images in their outputs.

TEDS scores close to 0.87, that shows model has capability to extract both contents and the tree level structure correctly. NanonetsOCR2 is the main exception, showing lower performance (NED = 0.11, TEDS = 0.80).

Figures Since only DeepSeekOCR, Docling, and PaddleOCR are able to extract figures from the documents, we evaluate only these models in the figure section. DeepSeekOCR leads on figure extraction, achieving the lowest MSE (0.04) and the lowest DreamSim score (DS = 0.06). All three models obtain very low MSE values, indicating that their extracted figures are highly similar to the ground truth at the pixel level. However, PaddleOCR and Docling show high DreamSim scores, suggesting lower perceptual similarities despite comparable pixel-level accuracies.

5.2 Difficulty Level Based Results

Difficulty-based results are summarized in Figure 1. For each model, we first compute the average NED across raw text, tables, and equations, and then report a unified score as $1 - \text{avg}(\text{NED})$, so that higher values indicate better performance and comparisons are more intuitive.

In the easy category, all models achieve near-perfect performance, with an average score of 0.98. In the medium category, performance remains high with an average score of 0.91. DeepSeekOCR, Docling, HuanyanOCR, OlmOCR2, and PaddleOCR perform above the overall average, even though the documents include basic structure, such as simple tables and short equations. In the hard category, PaddleOCR achieves the best score (0.89), despite the presence of more complex content, such as multi-line equations, plots with subplots, and larger tables. Overall, 5 out of 7 models score above the average in the hard set, while Docling and Nvidia Nemotron v1.1 fall below the average.

5.3 Category Based Results

Category-based results are summarized in Figure 2. For each model, we first compute the average NED across raw texts, tables, and equations. We then report a unified score as $1 - \text{avg}(\text{NED})$, so that higher values indicate better performance and comparisons are more intuitive.

Among the four categories (academic documents, non-academic documents, theses, and slideshows), non-academic documents show the highest average score with 0.94 whereas slideshows show the lowest average score with 0.86. Among the seven models (DeepSeekOCR, Docling, PaddleOCR, HuanyanOCR, NanonetsOCR2, Nvidia Nemotron v1.1, and OlmOCR2), PaddleOCR either has the highest or tied-highest scores in three of the four categories. Nvidia Nemotron v1.1 shows the lowest scores in academic documents, non-academic documents, and theses. While HuanyanOCR and PaddleOCR performed the best, OlmOCR2 performed the worst in slideshows. For Docling, we observe the highest gap (0.12) between the two categories: academic docs (0.85) and non-academic docs (0.97).

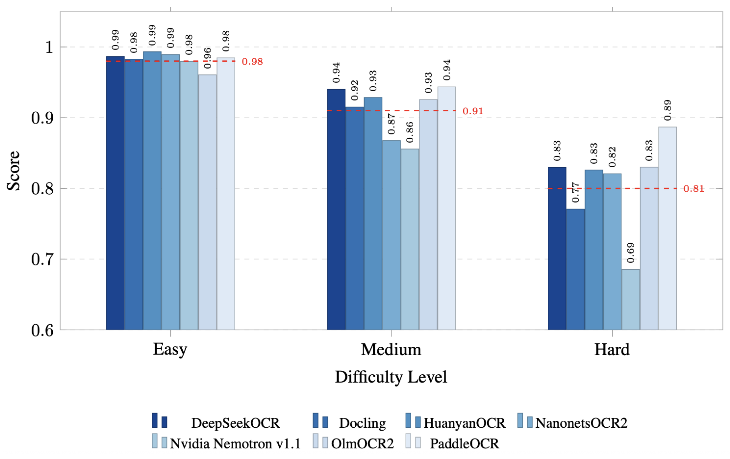


Figure 1: Comparison of the models DeepSeekOCR, Docling, PaddleOCR, HuanyanOCR, NanonetsOCR2, Nvidia Nemotron v1.1, and OlmOCR2 under easy, medium, and hard data. The scores are the averages of the NED metric scores for raw texts, tables, and equations of the data within the same difficulty. The NED scores are subtracted from 1 (Score $= 1 - \text{NED}$) for better comparison. The average score of the models within each difficulty level is given as the dashed red line.

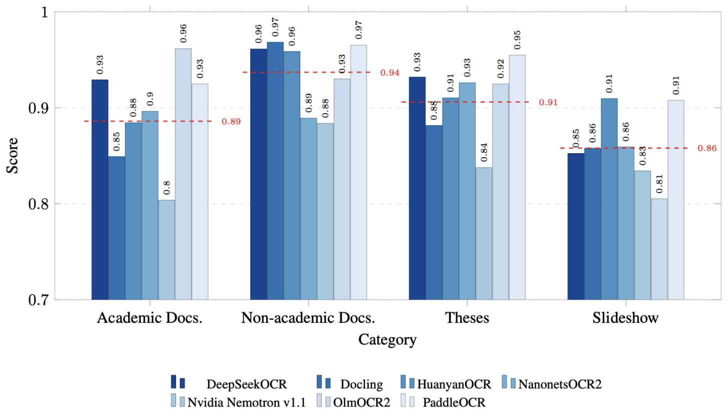


Figure 2: Comparison of the models DeepSeekOCR, Docling, PaddleOCR, HuanyanOCR, NanonetsOCR2, Nvidia Nemotron v1.1, and OlmOCR2 under the categories academic documents, non-academic documents, theses, and slideshows. The scores are the averages of the NED metric scores for raw texts, tables, and equations of the data within the same difficulty. The NED scores are subtracted from 1 (Score $= 1 - \text{NED}$) for better comparison. The average score of the models within each category is given as the dashed red line.

Overall, only PaddleOCR performed above the average in all four categories. The observed performance gap between slideshow documents and other document types can be attributed to the inherently unstructured nature of slideshows. Unlike theses and academic papers, which typically follow well-defined formatting standards, slideshows often contain highly unstructured text, irregular layouts, and ambiguous or incomplete table representations. This lack of standardized structure increases extraction difficulty and leads to degraded performance compared to more consistently formatted documents.

5.4 Error Analysis

The models share recurring errors when extracting Turkish special characters, leading to missing titles, line skips and line-break issues. They produce incorrect variables in equations (such as range shifts in integrals or sums) and sometimes write the equations or mathematical variables in text mode instead of in math mode, resulting in an incorrect number of equations in the document. The models sometimes skip the tables or fail to recognize them as tables, treating them as text instead. They occasionally generate data for empty cells or create extra columns. In cases where a column of the document includes an image, the models often interpret the entire section as an image. All of these errors combined results in penalties to these models' accuracy scores. A detailed error analysis with examples is provided in Appendix A.

6 Conclusion

Prior Turkish document parsing research highlights a clear gap: the absence of a Turkish document parsing benchmark that reflects real-world scenarios. To address this need, we introduce OCRTurk, the first, to the best of our knowledge, Turkish document parsing benchmark. OCRTurk comprises 180 documents spanning multiple document types and three difficulty levels, designed to capture the diversity and challenges of real-world Turkish documents. We evaluate seven OCR models on OCRTurk.

Most of the models show their strength for different element types, such as NanonetsOCR2 performing the best for equation, but the worst for tables. However, PaddleOCR achieves the strongest overall performance on element-level metrics, while DeepSeekOCR leads in figure extraction and HuanyanOCR leads in Turkish Charater Sensitivity. Under difficulty-based evaluation, DeepSeekOCR, HuanyanOCR, and PaddleOCR consistently outperform the overall average across all difficulty levels. Performance also varies substantially by document type: models perform best on non-academic documents and theses, and worst on slideshows.

Future work will focus on expanding the dataset by incorporating additional samples and a broader range of document types. In parallel, we plan to release the benchmark through an online leaderboard, enabling systematic evaluation and comparison of newly developed models.

Limitations

This benchmark contains 180 documents. Because generating and manually correcting ground truth annotations is time-consuming, we limit the benchmark to 180 documents in this release. Expanding the dataset would increase its diversity and enable evaluation on a wider range of edge-case scenarios. Similarly, extending the set of document types would better capture the variety of real-world documents and further strengthen the benchmark's coverage.

Ethical Considerations

OCRTurk is constructed using publicly accessible documents from platforms, such as ArXiv, DergiPark, YÖKTez, Ankara Üniversitesi Açık Kaynak Materyal, MEB OGM Materyal, and similar websites. The data are collected from a wide range of domains, including artificial intelligence, geography, finance, mathematics, and others, to reduce domain bias and evaluate models across diverse document types and writing styles. All data are collected in compliance with the terms of use of these platforms, and no private, sensitive, or restricted information is included. The dataset is intended strictly for academic research and for evaluating the performance of OCR and document parsing models; it has no commercial or financial objectives. Since the documents are already publicly available, the collection process does not violate individual privacy rights.

Regarding data annotation, Large Language Models, including GPT-5 and Gemini 2.5 Flash, are utilized to generate initial drafts of LaTeX and HTML structures. Recognizing the risk of model hallucinations, every annotation is manually reviewed and corrected by the authors to ensure the ground truth's absolute accuracy. Additionally, LLMs are used solely for proofreading and refining the grammatical clarity of the human-written text to improve readability.

Generative AI (ChatGPT[15] and Gemini[16]) is used in writing of this study to assist with language editing. All scientific contributions, data construction, data analysis, and interpretations presented in this work are original and were conducted entirely by the authors.

Acknowledgments

We gratefully acknowledge the support of Google Academic Program for providing Google Cloud[17] credits that facilitated this research. We express our sincere gratitude to Roketsan Inc.[18] for their valuable support throughout the development of this work.

References

E. A. Al-Zubaidi, M. M. Mijwil, and A. Sh. Alsaadi. 2019. Two-dimensional optical character recognition of mouse drawn in Turkish capital letters using multi-layer perceptron classification. *Journal of Southwest Jiaotong University*, 54(4).

[15]https://chatgpt.com/
[16]https://gemini.google.com
[17]https://cloud.google.com/
[18]http://www.roketsan.com.tr/

Avinash Anand, Raj Jaiswal, Pijush Bhuyan, Mohit Gupta, Siddhesh Bangar, Md. Modassir Imam, Rajiv Ratn Shah, and Shin'ichi Satoh. 2023. TC-OCR: TableCraft OCR for efficient detection & recognition of table structure & content. In *Proceedings of the 1st International Workshop on Deep Multimodal Learning for Information Retrieval*, MM '23, pages 11–18. ACM.

G. E. Bartos, Y. Hoşcan, A. Kauer, and É. Hajnal. 2020. A multilingual handwritten character dataset: T-h-e dataset. *Acta Polytechnica Hungarica*, 17(9):141–160.

Y. Du, P. Chen, X. Ying, and Z. Chen. 2025. Docptbench: Benchmarking end-to-end photographed document parsing and translation.

L. Fu, B. Yang, Z. Kuang, J. Song, Y. Li, L. Zhu, Q. Luo, X. Wang, H. Lu, M. Huang, Z. Li, G. Tang, B. Shan, C. Lin, Q. Liu, B. Wu, H. Feng, H. Liu, C. Huang, and J. Tang. 2024. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning.

S. Fu, N. Tamir, S. Sundaram, L. Chai, R. Zhang, T. Dekel, and P. Isola. 2023. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. *Preprint*, arXiv:2306.09344.

M. G. Öztürk, Durmus Ö. Sahin, and Erdal Kiliç. 2025. Turkish optical character recognition under the lens: A systematic review of language-specific challenges, dataset scarcity, and open-source limitations. *IEEE Access*, 13:168977–168997.

Dilek Z. Hakkani-Tür, Kemal Oflazer, and Gökhan Tür. 2002. Statistical morphological disambiguation for agglutinative languages. *Computers and the Humanities*, 36(4):381–410.

A. Heakl, A. Sohail, M. Ranjan, R. Hossam, G. Shazan Ahmad, Mohamed El-Geish, Omar Maher, Zhiqiang Shen, Fahad Khan, and Salman Khan. 2025. Kitab-bench: A comprehensive multi-domain benchmark for Arabic ocr and document understanding. *Preprint*, arXiv:2502.14949.

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and C. V. Jawahar. 2019. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE.

F. Kizilirmak. 2022. Offline handwriting recognition using deep learning with emphasis on data augmentation effects. Master's thesis, Sabancı University.

M. Kuncan, E. Vardar, K. Kaplan, and H. M. Ertunç. 2020. Turkish handwriting recognition system using multi-layer perceptron. *Journal of Mechatronics and Artificial Intelligence in Engineering*.

T.-Y. Lin, M. Maire, S. Belongie, and 1 others. 2014. *Microsoft COCO: Common Objects in Context*. Springer International Publishing.

Harsh Lunia, Ajoy Mondal, and C. V. Jawahar. 2023. *IndicSTR12: A Dataset for Indic Scene Text Recognition*, page 233–250. Springer Nature Switzerland.

J. Memon, M. Sami, and R. A. Khan. 2020. Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). *Preprint*, arXiv:2001.00139.

Rachit S. Munjal, Arun D. Prabhu, Nikhil Arora, Sukumar Moharana, and Gopi Ramena. 2021. Stride: Scene text recognition in-device. In *2021 International Joint Conference on Neural Networks (IJCNN)*, page 1–8. IEEE.

Kemal Oflazer. 2014. Turkish and its challenges for language processing. *Language resources and evaluation*, 48(4):639–653.

Berke Oral, Erdem Emekligil, Seçil Arslan, and Gülşen Eryiğit. 2020. Information extraction from text intensive and visually rich banking documents. *Information Processing and Management*, 57(6).

L. Ouyang, Y. Qu, H. Zhou, J. Zhu, R. Zhang, Q. Lin, B. Wang, Z. Zhao, M. Jiang, X. Zhao, J. Shi, F. Wu, P. Chu, M. Liu, Z. Li, C. Xu, B. Zhang, B. Shi, Z. Tu, and C. He. 2024. Omnidocbench: Benchmarking diverse PDF document parsing with comprehensive annotations.

Smita Pallavi, Raj Ratn Pranesh, and Sumit Kumar. 2020. A conglomerate of multiple ocr table detection and extraction. *Preprint*, arXiv:2010.08591.

Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaeheung Surh, Minjoon Seo, and Hwalsuk Lee. 2019. Cord: A consolidated receipt dataset for post-ocr parsing.

Parshva Dhilankumar Patel. 2025. Design and implementation of an ocr-powered pipeline for table extraction from invoices. *Preprint*, arXiv:2507.07029.

J. Poznanski, A. Rangapur, J. Borchardt, J. Dunkelberger, R. Huff, D. Lin, C. Wilhelm, K. Lo, and L. Soldaini. 2025. olmocr: Unlocking trillions of tokens in PDFs with vision-language models.

Cagri Sayallar, Ahmet Sayar, and Nurcan Babalik. 2023. An ocr engine for printed receipt images using deep learning techniques. *International Journal of Advanced Computer Science and Applications*, 14(2).

A. Şevik. 2019. Derin öğrenme ile Türkçe font ve karakter tanıma. Master's thesis, Düzce University.

Çağrı Toraman, Ahmet Kaan Sever, Ayşe Aysu Cengiz, Elif Ecem Arslan, Görkem Sevinç, Mete Mert Birdal, Yusuf Faruk Güldemir, Ali Buğra Kanburoğlu, Sezen Felekoğlu, Osman Gürlek, Sarp Kantar, Birsen Şahin Kütük, Büşra Tufan, Elif Genç, Serkan Coşkun, Gupse Ekin Demir, Muhammed Emin Arayıcı, Olgun Dursun, Onur Gungor, and 3 others. 2026. Turkbench: A benchmark for evaluating Turkish large language models. *Preprint*, arXiv:2601.07020.

Elif Ecem Umutlu, Ayse Aysu Cengiz, Ahmet Kaan Sever, Seyma Erdem, Burak Aytan, Busra Tufan, Abdullah Topraksoy, Esra Darıcı, and Cagri Toraman. 2025. Evaluating the quality of benchmark datasets for low-resource languages: A case study on Turkish. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 471–487, Vienna, Austria and virtual meeting. Association for Computational Linguistics.

C. Wickrema, S. Leary, S. Sarkar, M. Giglio, E. Bianchi, E. Mace, and M. Twardowski. 2025. Benchmarking image similarity metrics for novel view synthesis applications. *Preprint*, arXiv:2506.12563.

Z. Yang, J. Tang, Z. Li, P. Wang, J. Wan, H. Zhong, X. Liu, M. Yang, P. Wang, S. Bai, L. Jin, and J. Lin. 2024. Cc-ocr: a comprehensive and challenging OCR benchmark for evaluating large multimodal models in literacy.

Y. Yılmaz, E. G. Hanoğlu, A. G. Özkan, and K. Öztoprak. 2025. Benchmarking OCR and vision-language models for Turkish text recognition: A comprehensive evaluation using synthetic data.

Ahmed Zeer, Eren Dogan, Yusuf Erdem, Elif İnce, Osama Shbib, M. Egemen Uzun, Atahan Uz, M. Kaan Yuce, H. Toprak Kesgin, and M. Fatih Amasyali. 2024. Cosmos-llava: Chatting with the visual. In *2024 8th International Artificial Intelligence and Data Processing Symposium (IDAP)*, pages 1–7. IEEE.

Yufeng Zhong, Zhixiong Zeng, Lei Chen, Longrong Yang, Liming Zheng, Jing Huang, Siqi Yang, and Lin Ma. 2025. Doctron-formula: Generalized formula recognition in complex and structured scenarios. *Preprint*, arXiv:2508.00311.

A Error Analysis

Texts Models tend to show the Turkish characters (such as ğ and ş) as two separate characters. For instance, they interpret ğ as a breve symbol followed by a g, and ş as a cedilla symbol followed by an s. For example, "BEŞERİ" is interpreted as "BE¸ SERİ" and "DOĞAL" is interpreted as "DO˘ GAL". Another observation is that the uppercase i character (İ, not I as in English) in Turkish, is interpreted as Ì by some the models. For example, "GSYİH" is interpreted as "GSYÌH". Even though we can solve this problem with post-processing, it might be challenging to predict how other models than the ones we evaluate will behave when encountered with such characters. We observe that some duplicated letters are not taken into account by the models. For example, "Laplace'ın *Kuvvet* Fonksiyonları" is interpreted as "Laplace'ın *Kuvet* Fonksiyonları" (missing a "v"). Also, models tend to ignore whitespaces and inline math equations as well. For example, "m *ve* n'nin reel sayı olması" is interpreted as "*mven*'nin reel sayı olması".

Equations We observe incorrect variables and missing conditions for the partial expressions in the models. For example, The expression

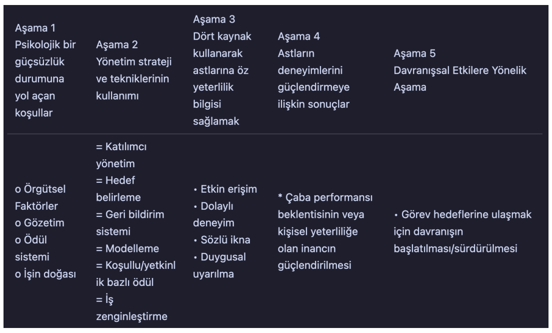
$$\int_a^b f(t)\Delta t = \begin{cases} \sum_{k=\frac{a}{h}}^{\frac{b}{h}-1} f(kh)h, & a < b \text{ ise,} \\ 0, & a = b \text{ ise,} \\ -\sum_{k=\frac{b}{h}}^{\frac{a}{h}-1} f(kh)h, & a > b \text{ ise} \end{cases}$$

is interpreted as

$$\int_a^b f(t)\Delta t = \left\{ \begin{array}{c} \sum_{k=\frac{a}{h}}^{\frac{b}{h}-1} f(kh)h, \\ 0, \\ -\sum_{k=\frac{a}{h}}^{\frac{a}{h}-1} f(kh)h, \end{array} \right.$$

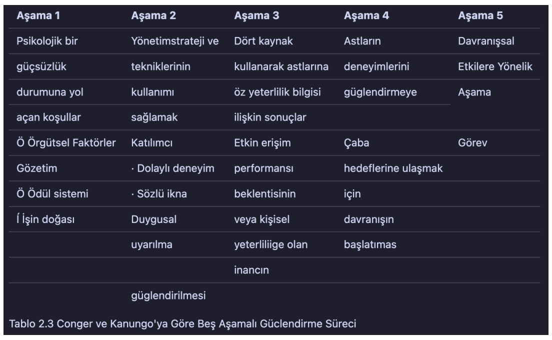
Tables We encounter structural mistakes in the tables of the model outputs. These mistakes include incorrect split of the cells, unnecessary rows or columns, wrongly merged rows or columns. An example of the structural mistake is shown in Figure 3.

Figures Models interpret multiple figures as a one complete figure. An example of this figure combination error is shown in Figure 4. Here the original PDF includes 5 images in a table. However, the model interpreted as a whole image.



Aşama 1 Psikolojik bir güçsüzlük durumuna yol açan koşullar	Aşama 2 Yönetim strateji ve tekniklerinin kullanımı	Aşama 3 Dört kaynak kullanarak astlarına öz yeterlilik bilgisi sağlamak	Aşama 4 Astların deneyimlerini güçlendirmeye ilişkin sonuçlar	Aşama 5 Davranışsal Etkilere Yönelik Aşama
o Örgütsel Faktörler o Gözetim o Ödül sistemi o İşin doğası	= Katılımcı yönetim = Hedef belirleme = Geri bildirim sistemi = Modelleme = Koşullu/yetkinlik bazlı ödül = İş zenginleştirme	• Etkin erişim • Dolaylı deneyim • Sözlü ikna • Duygusal uyarılma	* Çaba performansı beklentisinin veya kişisel yeterliliğe olan inancın güçlendirilmesi	• Görev hedeflerine ulaşmak için davranışın başlatılması/sürdürülmesi

(a) Ground Truth



Aşama 1	Aşama 2	Aşama 3	Aşama 4	Aşama 5
Psikolojik bir	Yönetimstrateji ve	Dört kaynak	Astların	Davranışsal
güçsüzlük	tekniklerinin	kullanarak astlarına	deneyimlerini	Etkilere Yönelik
durumuna yol	kullanımı	öz yeterlilik bilgisi	güglendirmeye	Aşama
açan koşullar	sağlamak	ilişkin sonuçlar		
Ö Örgütsel Faktörler	Katılımcı	Etkin erişim	Çaba	Görev
Gözetim	· Dolaylı deneyim	performansı	hedeflerine ulaşmak	
Ö Ödül sistemi	· Sözlü ikna	beklentisinin	için	
Í İşin doğası	Duygusal	veya kişisel	davranışın	
	uyarılma	yeterliliige olan	başlatımas	
		inancın		
	güglendirilmesi			

Tablo 2.3 Conger ve Kanungo'ya Göre Beş Aşamalı Güclendirme Süreci

(b) Model Output

Figure 3: Comparison of the tables between the ground truth (a) and the model output (b). As the structure of the table in the model output is incorrect, it spans to more than 50 lines. Thus, we ignore the rest of the model output in this figure.

GQA	CosmosVQA	VG	COCO	CC3M
Kullanıcı: Gökyüzü karanlık mı? **Model:** Evet **Kullanıcı:** Sokakta ne var? **Model:** Araba **Kullanıcı:** Caddede otobüs var mı? **Model:** Hayır **Kullanıcı:** Trafik sinyali sağda mı yoksa solda mı? **Model:** Sol 	**Kullanıcı:** Kitabın ismini öğrenebilir miyim? **Model:** Masumiyet Müzesi **Kullanıcı:** Bu kitap hangi türde bir kitaptır? **Model:** Aşk romanı **Kullanıcı:** Bu kitabın yazarı kimdir? **Model:** Orhan Pamuk 	**Kullanıcı:** Lütfen bu bölge için kısa bir açıklama yapınız: [0.69, 0.35, 0.79, 0.69]. **Model:** Bu bir insan. **Kullanıcı:** Lütfen bu cümlenin tanımladığı bölgenin sınırlayıcı kutu koordinatını verin: tenis kortundaki beyaz çizgi. **Model:** [0.0, 0.7, 0.9, 0.82] 	**Kullanıcı:** Lütfen bu bölge için kısa bir açıklama yapınız: [0.3, 0.36, 0.62, 0.59]. **Model:** Fotoğrafçıya bakan bir bilgisayar ekranı. **Kullanıcı:** Lütfen bu cümlenin tanımladığı bölgenin sınırlayıcı kutu koordinatını verin: en sağdaki ekran. **Model:** [0.3, 0.36, 0.62, 0.59] 	**Kullanıcı:** Sunulan görüntüyü temsil eden kompakt bir anlatı oluşturun. **Model:** çiftler için toplu nikah töreni düzenlendi.

Figure 4: A figure of the model output which shows the misinterpretation of the table as an image, which should have been five separate images in a table, mixed with texts.