ICDAR 2023 Competition on Hierarchical Text Detection and Recognition

Shangbang Long(✉), Siyang Qin, Dmitry Panteleev, Alessandro Bissacco, Yasuhisa Fujii, and Michalis Raptis

Google Research, Mountain View, CA, USA
{longshangbang,qinb,dpantele,bissacco,yasuhisaf,mraptis}@google.com

Abstract. We organize a competition on hierarchical text detection and recognition. The competition is aimed to promote research into deep learning models and systems that can jointly perform text detection and recognition and geometric layout analysis. We present details of the proposed competition organization, including tasks, datasets, evaluations, and schedule. During the competition period (from January 2nd 2023 to April 1st 2023), at least 50 submissions from more than 20 teams were made in the 2 proposed tasks. Considering the number of teams and submissions, we conclude that the HierText competition has been successfully held. In this report, we will also present the competition results and insights from them.

Keywords: OCR · Text Detection and Recognition · Layout Analysis

1 Introduction

Text detection and recognition systems [10] and geometric layout analysis techniques [11,12] have long been developed separately as independent tasks. Research on text detection and recognition [13–16] has mainly focused on the domain of natural images and aimed at single level text spotting (mostly, word-level). Conversely, research on geometric layout analysis [11,12,17,18], which is targeted at parsing text paragraphs and forming text clusters, has assumed document images as input and taken OCR results as fixed and given by independent systems. The synergy between the two tasks remains largely under-explored.

Recently, the Unified Detector work by Long et al. [19] shows that the unification of line-level detection of text and geometric layout analysis benefits both tasks significantly. StructuralLM [20] and LayoutLMv3 [26] show that text line grouping signals are beneficial to the downstream task of document understanding and are superior to word-level bounding box signals. These initial studies demonstrate that the unification of OCR and layout analysis, which we term as *Hierarchical Text Detection and Recognition (HTDR)*, can be mutually beneficial to OCR, layout analysis, and downstream tasks.

Given the promising potential benefits, we propose the **ICDAR 2023 Competition on Hierarchical Text Detection and Recognition**. In this competition, candidate systems are expected to perform the unified task of text detection and recognition and geometric layout analysis. Specifically, we define the

G. A. Fink et al. (Eds.): ICDAR 2023, LNCS 14188, pp. 483–497, 2023.
https://doi.org/10.1007/978-3-031-41679-8_28

unified task as producing a hierarchical text representation, including word-level bounding boxes and text transcriptions, as well as line-level and paragraph-level clustering of these word-level text entities. We defer the rigorous definitions of word/line/paragraph later to the dataset section. Figure 1 illustrates our notion of the unified task.

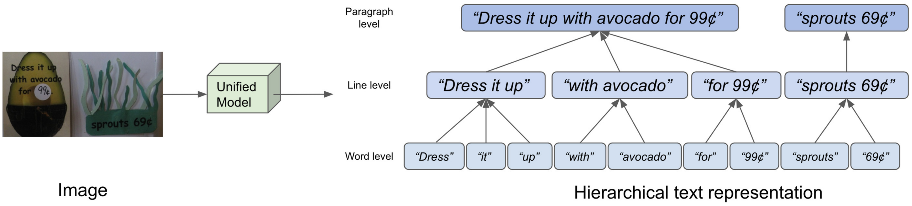


Fig. 1. Illustration for the proposed unified task: **Hierarchical Text Detection and Recognition (HTDR)**. Given an input image, the unified model is expected to produce a hierarchical text representation, which resembles the form of a forest. Each tree in the forest represents one paragraph and has three layers, representing the clustering of words into lines and then paragraphs.

We believe this competition will have profound and long-term impact on the whole image-based text understanding field by unifying the efforts of text detection and recognition and geometric layout analysis, and furthering providing new signals for downstream tasks.

The competition started on January 2nd 2023, received more than 50 submissions in 2 tasks in total, and closed on April 1st 2023. This report provides details into the motivation, preparation, and results of the competition. We believe the success of this competition greatly promotes the development of this research field. Furthermore, the dataset except the test set annotation and evaluation script are made publicly available. The competition website[1] remains open to submission and provides evaluation on the test set.

2 Competition Protocols

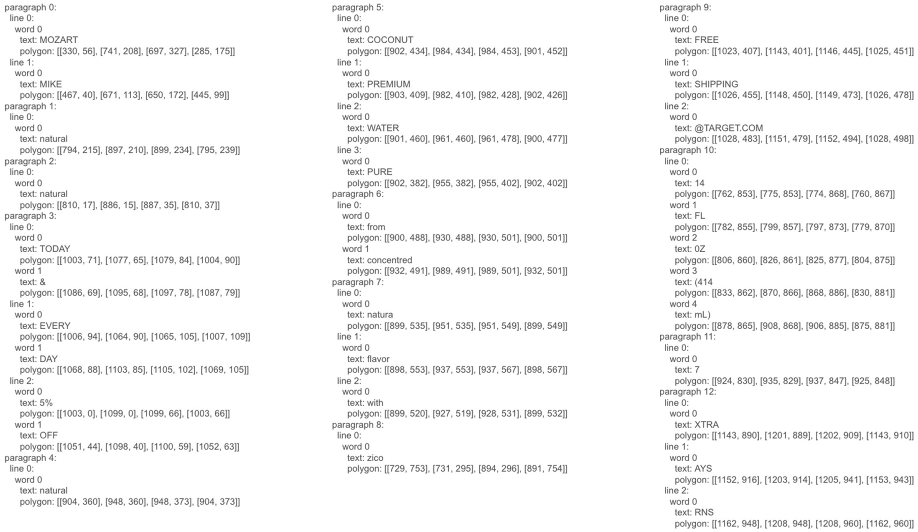
2.1 Dataset

The competition is based on the HierText dataset [19]. Images in HierText are collected from the Open Images v6 dataset [27], by first applying the *Google Cloud Platform (GCP) Text Detection API*[2] and then filtering out inappropriate images, for example those with too few text or non-English text. In total, 11639 images are obtained. In this competition, we follow the original split of 8281/1724/1634 for *train*, *validation*, *test* sets. Images and annotations of the train and validation set are released publicly. The test set annotation is kept private and will remain so even after the end of the competition.

[1] https://rrc.cvc.uab.es/?ch=18.

[2] https://cloud.google.com/vision/docs/ocr.

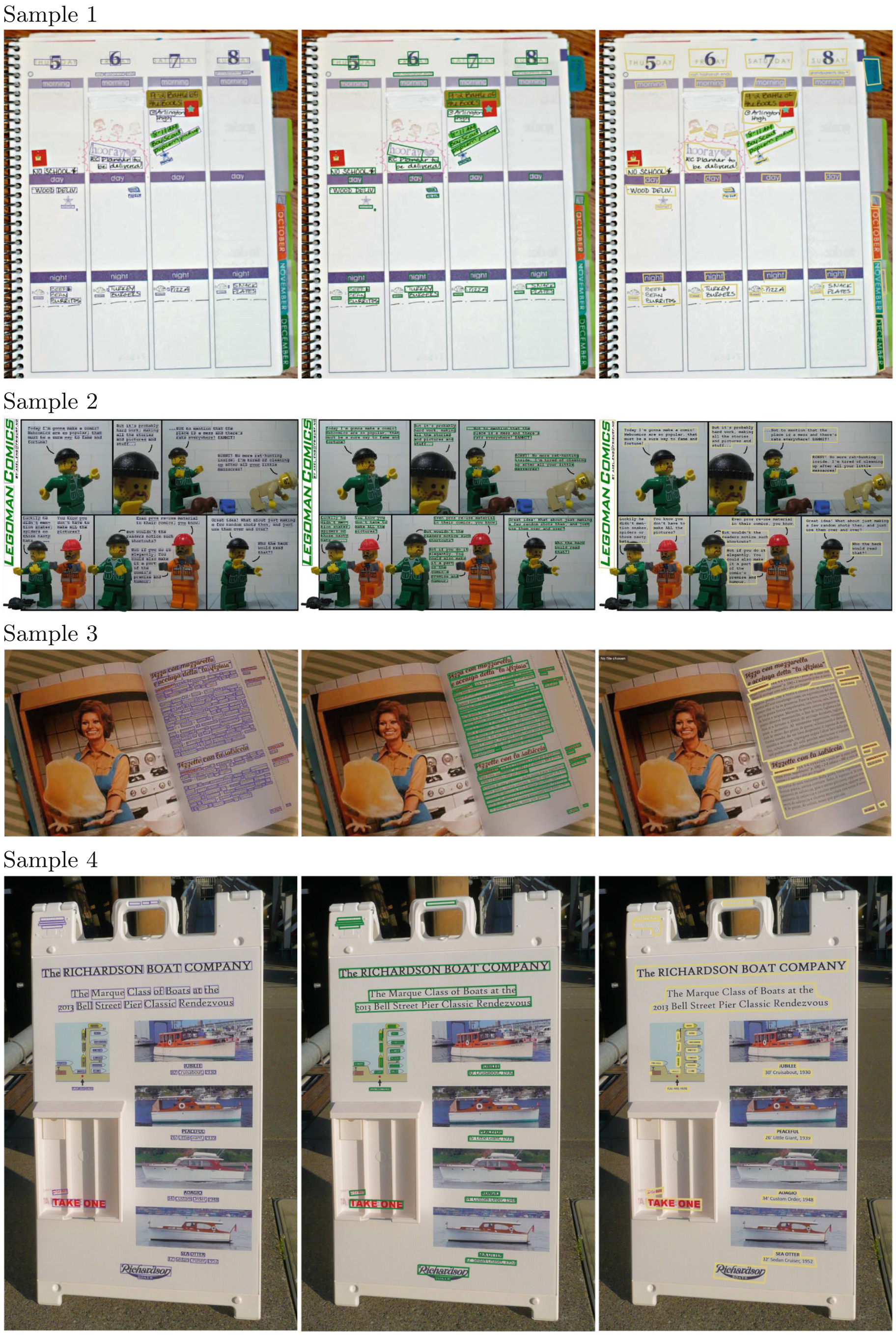
As noted in the original paper [19], we check the cross-dataset overlap rates with the two other OCR datasets that are based on Open Images. We find that 1.5% of the 11639 images we have are also in TextOCR [28] and 3.6% in Intel OCR [29]. Our splits ensure that our training images are not in the validation or test set of Text OCR and Intel OCR, and vice versa.



```
paragraph 0:
 line 0:
  word 0
   text: MOZART
   polygon: [[330, 56], [741, 208], [697, 327], [285, 175]]
 line 1:
  word 0
   text: MIKE
   polygon: [[467, 40], [671, 113], [650, 172], [445, 99]]
paragraph 1:
 line 0:
  word 0
   text: natural
   polygon: [[794, 215], [897, 210], [899, 234], [795, 239]]
paragraph 2:
 line 0:
  word 0
   text: natural
   polygon: [[810, 17], [886, 15], [887, 35], [810, 37]]
paragraph 3:
 line 0:
  word 0
   text: TODAY
   polygon: [[1003, 71], [1077, 65], [1079, 84], [1004, 90]]
  word 1
   text: &
   polygon: [[1086, 69], [1095, 68], [1097, 78], [1087, 79]]
 line 1:
  word 0
   text: EVERY
   polygon: [[1006, 94], [1064, 90], [1065, 105], [1007, 109]]
  word 1
   text: DAY
   polygon: [[1068, 88], [1103, 85], [1105, 102], [1069, 105]]
 line 2:
  word 0
   text: 5%
   polygon: [[1003, 0], [1099, 0], [1099, 66], [1003, 66]]
  word 1
   text: OFF
   polygon: [[1051, 44], [1098, 40], [1100, 59], [1052, 63]]
paragraph 4:
 line 0:
  word 0
   text: natural
   polygon: [[904, 360], [948, 360], [948, 373], [904, 373]]
paragraph 5:
 line 0:
  word 0
   text: COCONUT
   polygon: [[902, 434], [984, 434], [984, 453], [901, 452]]
 line 1:
  word 0
   text: PREMIUM
   polygon: [[903, 409], [982, 410], [982, 428], [902, 426]]
 line 2:
  word 0
   text: WATER
   polygon: [[901, 460], [961, 460], [961, 478], [900, 477]]
 line 3:
  word 0
   text: PURE
   polygon: [[902, 382], [955, 382], [955, 402], [902, 402]]
paragraph 6:
 line 0:
  word 0
   text: from
   polygon: [[900, 488], [930, 488], [930, 501], [900, 501]]
  word 1
   text: concentred
   polygon: [[932, 491], [989, 491], [989, 501], [932, 501]]
paragraph 7:
 line 0:
  word 0
   text: natura
   polygon: [[899, 535], [951, 535], [951, 549], [899, 549]]
 line 1:
  word 0
   text: flavor
   polygon: [[898, 553], [937, 553], [937, 567], [898, 567]]
 line 2:
  word 0
   text: with
   polygon: [[899, 520], [927, 519], [928, 531], [899, 532]]
paragraph 8:
 line 0:
  word 0
   text: zico
   polygon: [[729, 753], [731, 295], [894, 296], [891, 754]]
paragraph 9:
 line 0:
  word 0
   text: FREE
   polygon: [[1023, 407], [1143, 401], [1146, 445], [1025, 451]]
 line 1:
  word 0
   text: SHIPPING
   polygon: [[1026, 455], [1148, 450], [1149, 473], [1026, 478]]
 line 2:
  word 0
   text: @TARGET.COM
   polygon: [[1028, 483], [1151, 479], [1152, 494], [1028, 498]]
paragraph 10:
 line 0:
  word 0
   text: 14
   polygon: [[762, 853], [775, 853], [774, 868], [760, 867]]
  word 1
   text: FL
   polygon: [[782, 855], [799, 857], [797, 873], [779, 870]]
  word 2
   text: 0Z
   polygon: [[806, 860], [826, 861], [825, 877], [804, 875]]
  word 3
   text: (414
   polygon: [[833, 862], [870, 866], [868, 886], [830, 881]]
  word 4
   text: mL)
   polygon: [[878, 865], [908, 868], [906, 885], [875, 881]]
paragraph 11:
 line 0:
  word 0
   text: 7
   polygon: [[924, 830], [935, 829], [937, 847], [925, 848]]
paragraph 12:
 line 0:
  word 0
   text: XTRA
   polygon: [[1143, 890], [1201, 889], [1202, 909], [1143, 910]]
 line 1:
  word 0
   text: AYS
   polygon: [[1152, 916], [1203, 914], [1205, 941], [1153, 943]]
 line 2:
  word 0
   text: RNS
   polygon: [[1162, 948], [1208, 948], [1208, 960], [1162, 960]]
```

Fig. 2. Example of hierarchical annotation format of the dataset.

The images are annotated in a hierarchical way of *word*-to-*line*-to-*paragraph*, as shown in Fig. 2. *Words* are defined as a sequence of textual characters not interrupted by *spaces*. *Lines* are then defined as *space*-separated clusters of *words* that are logically connected and aligned in spatial proximity. Finally, *paragraphs* are composed of *lines* that belong to the same semantic topic and are geometrically coherent. Figure 3 illustrates some annotated samples. Words are annotated with polygons, with 4 vertices for straight text and more for curved text depending on the shape. Then, words are transcribed regardless of the scripts and languages, as long as they are legible. Note that we do not limit the character sets, so the annotation could contain case-sensitive characters, digits, punctuation, as well as non-Latin characters such as Cyrillic and Greek. After word-level annotation, we group words into lines and then group lines into paragraphs. In this way, we obtain a hierarchical annotation that resembles a forest structure of the text in an image.


Sample 1

Sample 2

Sample 3

Sample 4

THURSDAY
FRIDAY
SATURDAY
SUNDAY
5
6
7
8
morning
day
night
NO SCHOOL
WOOD DELIV.
hooray
RC Planner to be delivered
@Arlington High
BEEF & BEAN BURRITOS
TURKEY BURGERS
PIZZA
SNACK PLATES
OCTOBER
NOVEMBER
DECEMBER
LEGOMAN COMICS
Today I'm gonna make a comic! Webcomics are so popular, that must be a sure way to fame and fortune!
But it's probably hard work, making all the stories and pictures and stuff...
...Not to mention that the place is a mess and there's rats everywhere! DAMMIT!
ROBBY! No more rat-hunting inside. I'm tired of cleaning up after all your little massacres!
Luckily he didn't mention that spiders or those nasty bats
You know you don't have to make ALL the pictures?
Even pros re-use material in their comics, you know.
But wouldn't the readers notice such shortcuts?
Not if you do it elegantly. You could also make it a part of the comic's premise and humour.
Great idea! What about just making a few random shots then, and just use them over and over?
Who the heck would read that?!
Pizza con mozzarella e acciuga della "la sfiziosa"
Pizzette con la salsiccia
The RICHARDSON BOAT COMPANY
The Marque Class of Boats at the
2013 Bell Street Pier Classic Rendezvous
JUBILEE
30' Cruisabout, 1930
PEACEFUL
26' Little Giant, 1939
ADAGIO
34' Custom Order, 1948
SEA OTTER
32' Sedan Cruiser, 1952
TAKE ONE
Richardson

Fig. 3. Illustration for the hierarchical annotation of text in images. From **left** to **right**: **word**, **line**, **paragraph** level annotations. Words (blue) are annotated with polygons. Lines (green) and paragraphs (yellow) are annotated as hierarchical clusters and visualized as polygons. Images are taken from the train split.

2.2 Tasks

Our challenge consists of 2 competition tracks, **Hierarchical Text Detection** and **Word-Level End-to-End Text Detection and Recognition**. In the future, we plan to merge them into a single unified Hierarchical Text Spotting task that requires participants to give a unified representation of text with layout.

Task 1: Hierarchical Text Detection. This task itself is formulated as a combination of 3 tasks: word detection, text line detection, and paragraph detection, where lines and paragraphs are represented as clusters of words hierarchically.

In this task, participants are provided with images and expected to produce the hierarchical text detection results. Specifically, the results are composed of **word-level bounding polygons** and **line and paragraph clusters** on top of words. The clusters are represented as forests, as in Fig. 1, where each paragraph is a tree and words are leaves. For this task, participants do not need to provide text recognition results.

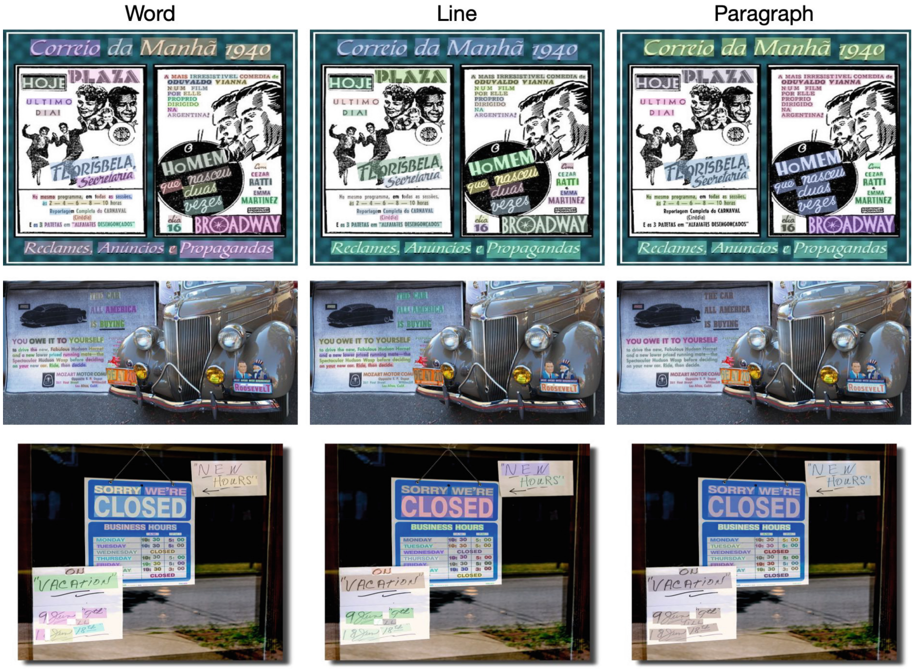


Fig. 4. Illustration of how hierarchical text detection can be evaluated as 3 instance segmentation sub-tasks. The coloring of each column indicates the instance segmentation for each sub-task.

As illustrated in Fig. 4, we evaluate this task as 3 instance segmentation subtasks for word, line, and paragraph respectively. For word level, each word is one instance. For line level, we take the union of each line's children words as one instance. For paragraph level, we aggregate each paragraph's children lines, and take that as one instance. With this formulation, all the 3 sub-tasks will be evaluated with the PQ metric [30] designed for instance segmentation, as specified in [19]:

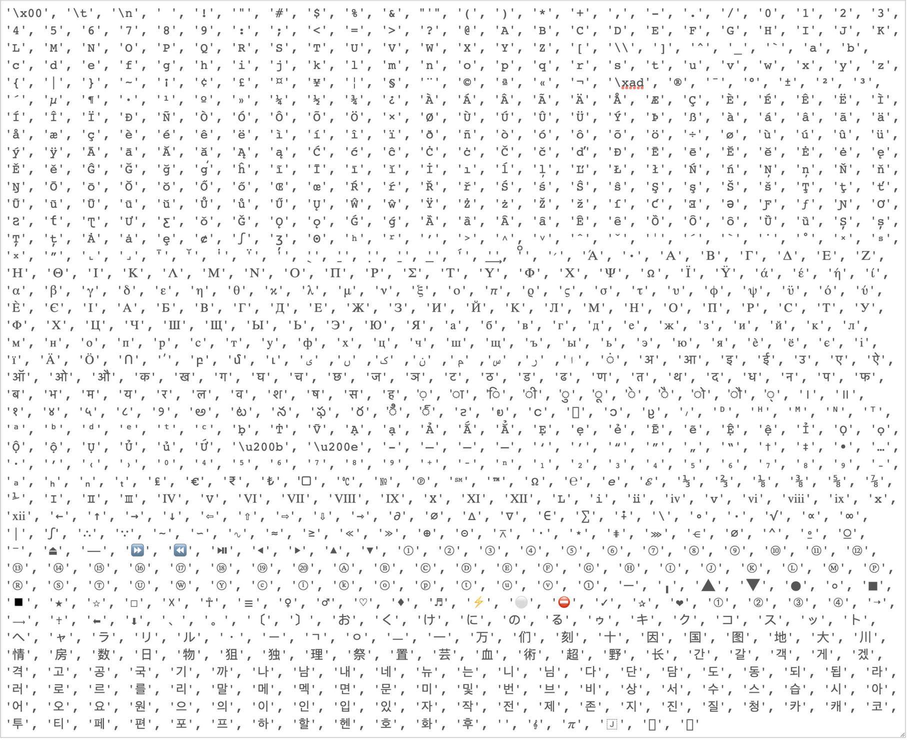
$$PQ = \frac{\sum_{(p,g)\in TP} IoU(p,g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \tag{1}$$

where TP, FP, FN represent true positives, false positives, and false negatives respectively. We use an IoU threshold of 0.5 to count true positives. Note that the PQ metric is mathematically equal to the product of the *Tightness* score, which is defined as the average IoU scores of all TP pairs, and the *F1*, score which is commonly used in previous OCR benchmarks. Previous OCR evaluation protocols only report F1 scores which do not fully reflect the detection quality. We argue that tightness is very important in evaluating hierarchical detection. It gives an accurate measurement of how well detections match ground-truths. For words, a detection needs to enclose all its characters and not overlap with other words, so that the recognition can be correct. The tightness score can penalize missing characters and oversized boxes. For lines and paragraphs, they are represented as clusters of words, and are evaluated as unions of masks. Wrong clustering of words can also be reflected in the IoU scores for lines and paragraphs. In this way, using the PQ score is an ideal way to accurately evaluate the hierarchical detection task.

Each submission has 3 PQ scores for word, line, and paragraph respectively. There are 3 rankings for these 3 sub-tasks respectively. For the final ranking of the whole task, we compute the final score as a harmonic mean of the 3 PQ scores (dubbed *H-PQ*) and rank accordingly.

Task 2: Word-Level End-to-End Text Detection and Recognition. For this task, images are provided and participants are expected to produce word-level text detection and recognition results, i.e. a set of word bounding polygons and transcriptions for each image. Line and paragraph clustering is not required. This is a challenging task, as the dataset has the most dense images, with more than 100 words per image on average, 3 times as many as the second dense dataset TextOCR [28]. It also features a large number of recognizable characters. In the training set alone, there are more than 960 different character classes, as shown in Fig. 5, while most previous OCR benchmarks limit the tasks to recognize only digits and case-insensitive English characters. These factors make this task challenging.

For evaluation, we use the F1 measure, which is a harmonic mean of word-level prediction and recall. A word result is considered true positive if the IoU with ground-truth polygon is greater or equal to 0.5 and the transcription is the same as the ground-truth. The transcription comparison considers all characters and will be case-sensitive. Note that, some words in the dataset are marked as illegible words. Detection with high overlap with these words (IoU larger than 0.5) will be removed in the evaluation process, and ground-truths marked as illegible do not count as false negative even if they are not matched.



```
'\x00', '\t', '\n', ' ', '!', '"', '#', '$', '%', '&', "'", '(', ')', '*', '+', ',', '-', '.', '/', '0', '1', '2', '3',
'4', '5', '6', '7', '8', '9', ':', ';', '<', '=', '>', '?', '@', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K',
'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', '[', '\\', ']', '^', '_', '`', 'a', 'b',
'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z',
'{', '|', '}', '~', '¡', '¢', '£', '¤', '¥', '¦', '§', '¨', '©', 'ª', '«', '¬', '\xad', '®', '¯', '°', '±', '²', '³',
'´', 'µ', '¶', '·', '¹', 'º', '»', '¼', '½', '¾', '¿', 'À', 'Á', 'Â', 'Ã', 'Ä', 'Å', 'Æ', 'Ç', 'È', 'É', 'Ê', 'Ë', 'Ì',
'Í', 'Î', 'Ï', 'Ð', 'Ñ', 'Ò', 'Ó', 'Ô', 'Õ', 'Ö', '×', 'Ø', 'Ù', 'Ú', 'Û', 'Ü', 'Ý', 'Þ', 'ß', 'à', 'á', 'â', 'ã', 'ä',
'å', 'æ', 'ç', 'è', 'é', 'ê', 'ë', 'ì', 'í', 'î', 'ï', 'ð', 'ñ', 'ò', 'ó', 'ô', 'õ', 'ö', '÷', 'ø', 'ù', 'ú', 'û', 'ü',
'ý', 'ÿ', 'Ā', 'ā', 'Ă', 'ă', 'Ą', 'ą', 'Ć', 'ć', 'ĉ', 'Ċ', 'ċ', 'Č', 'č', 'ď', 'Đ', 'Ē', 'ē', 'Ĕ', 'ĕ', 'Ė', 'ė', 'ę',
'Ě', 'ě', 'Ĝ', 'Ğ', 'ğ', 'ģ', 'ĥ', 'ĩ', 'Ī', 'ī', 'ĭ', 'İ', 'ı', 'Ĺ', 'ļ', 'Ľ', 'Ł', 'ł', 'Ń', 'ń', 'Ņ', 'ņ', 'Ň', 'ň',
'Ŋ', 'Ō', 'ō', 'Ŏ', 'ŏ', 'Ő', 'ő', 'Œ', 'œ', 'Ŕ', 'ŕ', 'Ř', 'ř', 'Ś', 'ś', 'Ŝ', 'ŝ', 'Ş', 'ş', 'Š', 'š', 'Ţ', 'ţ', 'ť',
'Ũ', 'ũ', 'Ū', 'ū', 'ŭ', 'Ů', 'ů', 'Ű', 'Ų', 'Ŵ', 'ŵ', 'Ÿ', 'Ż', 'ż', 'Ž', 'ž', 'ſ', 'Ƈ', 'Ǝ', 'ə', 'Ƒ', 'ƒ', 'Ɲ', 'Ơ',
'ƨ', 'ƭ', 'Ʈ', 'Ư', 'Ʃ', 'ǒ', 'Ǧ', 'Ǫ', 'ǫ', 'Ǵ', 'ǵ', 'Ȁ', 'ȁ', 'Ȃ', 'ȃ', 'Ȇ', 'ȇ', 'Ȍ', 'Ȏ', 'ȏ', 'Ȗ', 'ȗ', 'Ș', 'ș',
'Ț', 'ț', 'Ȧ', 'ȧ', 'ȩ', 'ȼ', 'ʃ', 'ʒ', 'ʘ', 'ʰ', 'ʳ', 'ʻ', '˃', '˄', '˅', 'ˆ', 'ˇ', 'ˈ', 'ˊ', 'ˋ', '˙', '˚', '˟', 'ˢ',
'ˣ', '˝', '˪', '˩', '̄', '̆', '̇', '̈', '̓', '̖', '̠', '̣', '̱', '̲', '́', '͢', '̊', '͵', 'Ά', '·', 'Α', 'Β', 'Γ', 'Δ', 'Ε', 'Ζ',
'Η', 'Θ', 'Ι', 'Κ', 'Λ', 'Μ', 'Ν', 'Ο', 'Π', 'Ρ', 'Σ', 'Τ', 'Υ', 'Φ', 'Χ', 'Ψ', 'Ω', 'Ϊ', 'Ϋ', 'ά', 'έ', 'ή', 'ί',
'α', 'β', 'γ', 'δ', 'ε', 'η', 'θ', 'ϰ', 'λ', 'μ', 'ν', 'ξ', 'ο', 'π', 'ϱ', 'ς', 'σ', 'τ', 'υ', 'φ', 'ψ', 'ϋ', 'ό', 'ύ',
'Ѐ', 'Є', 'І', 'А', 'Б', 'В', 'Г', 'Д', 'Е', 'Ж', 'З', 'И', 'Й', 'К', 'Л', 'М', 'Н', 'О', 'П', 'Р', 'С', 'Т', 'У',
'Ф', 'Х', 'Ц', 'Ч', 'Ш', 'Щ', 'Ы', 'Ь', 'Э', 'Ю', 'Я', 'а', 'б', 'в', 'г', 'д', 'е', 'ж', 'з', 'и', 'й', 'к', 'л',
'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ', 'ъ', 'ы', 'ь', 'э', 'ю', 'я', 'ѐ', 'ё', 'є', 'і',
'ї', 'Ӓ', 'Ӧ', 'Ո', '՚', 'բ', 'մ', 'ւ', 'ى' ,'ں' ,'ک' ,'ن' ,'م' ,'س' ,'ر' ,'ا', 'ं', 'अ', 'आ', 'इ', 'ई', 'उ', 'ए', 'ऐ',
'ऑ', 'ओ', 'औ', 'क', 'ख', 'ग', 'घ', 'च', 'छ', 'ज', 'ञ', 'ट', 'ठ', 'ड', 'ढ', 'ण', 'त', 'थ', 'द', 'ध', 'न', 'प', 'फ',
'ब', 'भ', 'म', 'य', 'र', 'ल', 'व', 'श', 'ष', 'स', 'ह', '़', 'ा', 'ि', 'ी', 'ु', 'ू', 'े', 'ै', 'ो', 'ौ', '्', '।', '॥',
'१', '४', '५', '८', '९', '൭', 'ഖ', 'ನ', 'ಫ', 'ಠ', 'ೆ', 'ೌ', 'ƨ', 'ɐ', 'ᑕ', '□', 'ↄ', 'ᵽ', 'ˏ', 'ᴰ', 'ᴴ', 'ᴹ', 'ᴺ', 'ᵀ',
'ᵃ', 'ᵇ', 'ᵈ', 'ᵉ', 'ᵗ', 'ᶜ', 'ḅ', 'Ṫ', 'Ṽ', 'Ạ', 'ạ', 'Ả', 'Ấ', 'Ẳ', 'Ẹ', 'ẹ', 'ẻ', 'Ẽ', 'ẽ', 'Ệ', 'ệ', 'Ỉ', 'Ọ', 'ọ',
'Ộ', 'ộ', 'Ụ', 'Ủ', 'ủ', 'Ứ', '\u200b', '\u200e', '‐', '–', '—', '―', '‘', '’', '“', '”', '„', '‟', '†', '‡', '•', '…',
'‧', '′', '‹', '›', '⁰', '⁴', '⁵', '⁶', '⁷', '⁸', '⁹', '⁺', '⁻', 'ⁿ', '₁', '₂', '₃', '₄', '₅', '₆', '₇', '₈', '₉', '₋',
'ₐ', 'ₕ', 'ₙ', 'ₜ', '₤', '€', '₹', '₺', '⃞', '℃', '№', '℗', '℠', '™', 'Ω', '℮', 'ℯ', 'ℰ', '⅓', '⅔', '⅛', '⅜', '⅝', '⅞',
'⅟', 'Ⅰ', 'Ⅱ', 'Ⅲ', 'Ⅳ', 'Ⅴ', 'Ⅵ', 'Ⅶ', 'Ⅷ', 'Ⅸ', 'Ⅹ', 'Ⅺ', 'Ⅻ', 'Ⅼ', 'ⅰ', 'ⅱ', 'ⅳ', 'ⅴ', 'ⅵ', 'ⅷ', 'ⅸ', 'ⅹ',
'ⅻ', '←', '↑', '→', '↓', '⇦', '⇧', '⇨', '⇩', '⇾', '∂', '∅', '∆', '∇', '∈', '∑', '∔', '∖', '∘', '∙', '√', '∝', '∞',
'∣', '∫', '∴', '∵', '∼', '∽', '∿', '≈', '≥', '≪', '≫', '⊕', '⊝', '⊼', '⋅', '⋆', '⋕', '⋙', '⋲', '⌀', '⌃', '⍛', '⍜',
'⎺', '⏏', '⏤', '⏩', '⏪', '⏯', '◂', '▸', '▴', '▾', '①', '②', '③', '④', '⑤', '⑥', '⑦', '⑧', '⑨', '⑩', '⑪', '⑫',
'⑬', '⑭', '⑮', '⑯', '⑰', '⑱', '⑲', '⑳', 'Ⓐ', 'Ⓑ', 'Ⓒ', 'Ⓓ', 'Ⓔ', 'Ⓕ', 'Ⓖ', 'Ⓗ', 'Ⓘ', 'Ⓙ', 'Ⓚ', 'Ⓛ', 'Ⓜ', 'Ⓟ',
'Ⓡ', 'Ⓢ', 'Ⓣ', 'Ⓤ', 'Ⓦ', 'Ⓨ', 'ⓒ', 'ⓘ', 'ⓚ', 'ⓞ', 'ⓟ', 'ⓣ', 'ⓤ', 'ⓥ', '⓵', '─', '╻', '▲', '▼', '●', '◦', '■',
'■', '★', '☆', '☐', '☓', '☥', '☰', '♀', '♂', '♡', '♦', '♬', '⚡', '⚪', '⛔', '✓', '✩', '❤', '➀', '➁', '➂', '➃', '➝',
'⟶', '⤉', '⬅', '⬇', '、', '。', '〔', '〕', 'お', 'く', 'け', 'に', 'の', 'る', 'ゥ', 'キ', 'ク', 'コ', 'ス', 'ッ', 'ト',
'ヘ', 'ャ', 'ラ', 'リ', 'ル', '・', 'ー', 'ㄱ', 'ㅇ', '㇐', '一', '万', '们', '刻', '十', '因', '国', '图', '地', '大', '川',
'情', '房', '数', '日', '物', '狙', '独', '理', '祭', '置', '芸', '血', '術', '超', '野', '长', '간', '갈', '객', '게', '겠',
'격', '고', '공', '국', '기', '까', '나', '남', '내', '네', '뉴', '는', '니', '님', '다', '단', '담', '도', '동', '되', '됩', '라',
'러', '로', '르', '를', '리', '말', '메', '멕', '면', '문', '미', '및', '번', '브', '비', '상', '서', '수', '스', '습', '시', '아',
'어', '오', '요', '원', '으', '의', '이', '인', '입', '있', '자', '작', '전', '제', '존', '지', '진', '질', '청', '카', '캐', '코',
'투', '티', '페', '편', '포', '프', '하', '할', '헨', '호', '화', '후', '', '𝄞', '𝜋', '🄹', '', ''
```

Fig. 5. Character set in the training split.

2.3 Evaluation and Competition Website

We host the competition on the widely recognized Robust Reading Competition (RRC) website[3] and set up our own competition page. The RRC website has been the hub of scene text and document understanding research for a long time and hosted numerous prestigious competitions. It provides easy-to-use infrastructure to set up competition, tasks, and carry out evaluation. It also supports running the competition continuously, making it an ideal candidate.

[3] https://rrc.cvc.uab.es/.

2.4 Competition Schedule

We propose and execute the following competition schedule, in accordance with the conference timeline:

- **January 2nd, 2023**: Start of the competition; submissions of results were enabled on the website.
- **April 1st, 2023**: Deadline for competition submissions.
- **April 15th, 2023**: Announcement of results.

2.5 Other Competition Rules

In addition to the aforementioned competition specifications, we also apply the following rules:

- **Regarding the usage of other publicly available datasets**: HierText is the only allowed annotated OCR dataset. However, participants are also allowed to do self-labeling on other public OCR datasets as long as they don't use their ground-truth labels. In other words, they can use the images of other public datasets, but not their labels. They can also use non-OCR datasets, whether labeled or not, to pretrain their models. We believe they are important techniques that can benefit this field.
- **Usage of synthetic datasets** Synthetic data has been an important part of OCR recently [21–25]. Participants can use any synthetic datasets, whether they are public or private, but are expected to reveal how they are synthesized and some basic statistics of the synthetic datasets if they are private.
- Participants should not use the validation split in training their models.
- Participants can make as many submissions as desired before the deadline, but we only archive the latest one submission of each participant in the final competition ranking.

2.6 Organizer Profiles

Authors are all members of the OCR team at Google Research. In addition to academic publications, authors have years of experience in building industrial OCR systems that are accurate and efficient for a diversity of image types and computation platforms.

3 Competition Results

In total, the competition received 30 submissions in Task 1 and 20 submissions in Task 2. Note that, we encourage participants to submit multiple entries using different methods, for example, to understand the effect of applying different techniques such as pretraining and synthetic data. To produce the final leaderboard in compliance with the ICDAR competition protocols, we only keep the latest 1 submission from each participants. The final deduplicated competition results are summarized in Table 1/Fig. 6 and Table 2/Fig. 7. In total, the competition received 11 unique submissions in Task 1 and 7 in Task 2.

Table 1. Results for Task 1. F/P/R/T/PQ stand for *F1-score*, *Precision*, *Recall*, *Tightness*, and *Panoptic Quality* respectively. The submissions are ranked by the *H-PQ* score. H-PQ can be interpreted as *Hierarchical-PQ* or *Harmonic-PQ*. H-PQ is calculated as the harmonic means of the PQ scores of the 3 hierarchies: word, line, and paragraph. It represents the comprehensive ability of a method to detect the text hierarchy in image. We omit the % for all these numbers for simplicity.

User	Method	Rank	Task 1 metric	Word					Line					Paragraph				
			H-PQ	PQ	F	P	R	T	PQ	F	P	R	T	PQ	F	P	R	T
YunSu Kim	Upstage KR	1	76.85	79.80	91.88	94.73	89.20	86.85	76.40	88.34	91.32	85.56	86.48	74.54	86.15	87.40	84.94	86.52
DeepSE x Upstage	DeepSE hierarchical detection model	2	70.96	75.30	88.49	93.50	83.99	85.10	69.43	82.43	82.65	82.21	84.23	68.51	81.39	81.69	81.10	84.17
zhm	hiertext_submit_0401 curve_199_v2	3	70.31	76.71	88.18	92.71	84.08	86.99	71.43	83.32	89.32	78.07	85.73	63.97	74.83	81.25	69.35	85.48
Mike Ranzinger	NVTextSpotter	4	68.82	73.69	87.07	95.10	80.29	84.63	67.76	80.42	93.87	70.35	84.25	65.51	78.04	81.82	74.60	83.94
ssm	Ensemble of three task-specific Clova DEER detection	5	68.72	71.54	92.03	93.82	90.31	77.74	69.64	89.04	91.75	86.49	78.21	65.29	83.70	84.17	83.23	78.01
xswl	Global and local instance segmentations for hierarchical text detection	6	68.62	76.16	90.72	93.45	88.16	83.95	68.50	82.22	80.24	84.31	83.31	62.55	75.11	74.00	76.25	83.28
Asaf Gendler	Hierarchical Transformers for Text Detection	7	67.59	70.44	86.09	88.47	83.83	81.82	69.30	85.23	87.83	82.78	81.31	63.46	78.40	77.84	78.97	80.94
JiangQing	SCUT-HUAWEI	8	62.68	70.08	89.58	89.79	89.37	78.23	67.70	86.20	90.46	82.33	78.53	53.14	69.06	74.03	64.72	76.96
Jiawei Wang	DQ-DETR	9	27.81	61.01	77.27	80.64	74.17	78.96	26.96	35.91	26.81	54.39	75.07	18.38	24.72	15.99	54.41	74.36
ZiqianShao	test	10	21.94	27.45	41.75	51.82	34.95	65.76	25.61	39.04	51.50	31.43	65.59	16.32	24.52	35.61	18.70	66.57
Yichuan Cheng	a	11	0.00	0.00	0.00	0.24	0.00	53.62	0.01	0.01	0.25	0.01	51.29	0.01	0.02	0.21	0.01	50.89

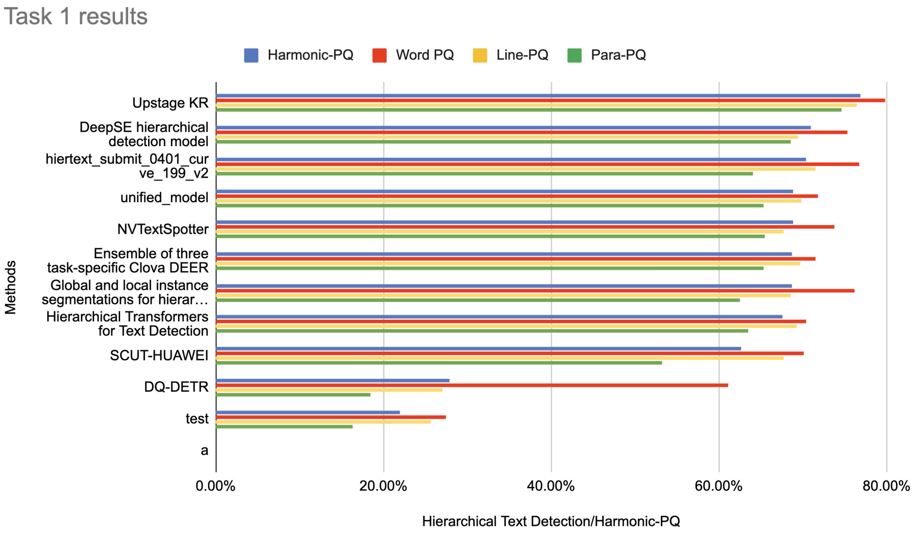


Fig. 6. Figure for the results of task 1.

Table 2. Results for Task 2. F/P/R/T/PQ stand for *F1-score*, *Precision*, *Recall*, *Tightness*, and *Panoptic Quality* respectively. The submissions are ranked by the F1 score. We omit the % for all these numbers for simplicity.

User	Method	Rank	Word				
			PQ	F	P	R	T
YunSu Kim	Upstage KR	1	70.00	79.58	82.05	77.25	87.97
DeepSE x Upstage	DeepSE End-to-End Text Detection and Recognition Model	2	67.46	77.93	88.05	69.89	86.57
ssm	Ensemble of three task-specific Clova DEER	3	59.84	76.15	77.63	74.73	78.59
Mike Ranzinger	NVTextSpotter	4	63.57	74.10	80.94	68.34	85.78
JiangQing	SCUT-HUAWEI	5	58.12	73.41	74.38	72.46	79.17
kuli.cyd	DBNet++ and SATRN	6	51.62	71.64	82.76	63.15	72.06
LGS	keba	7	44.87	54.30	68.37	45.03	82.64

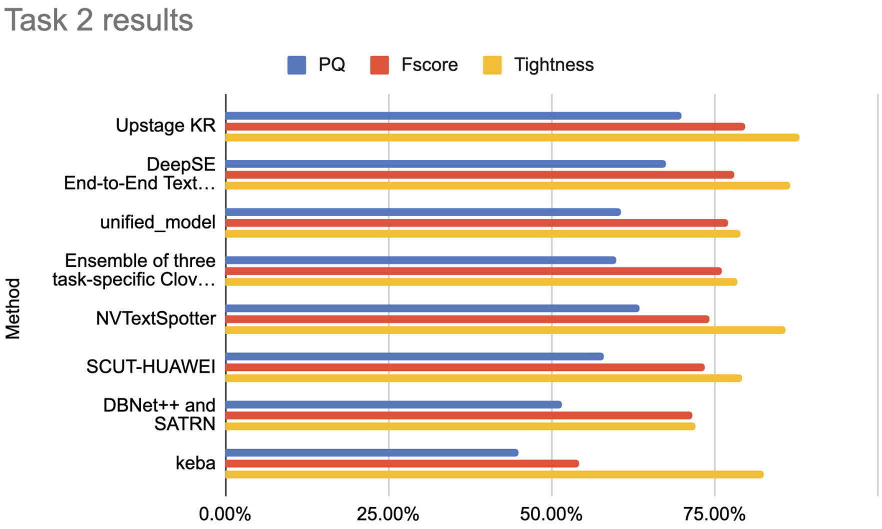


Fig. 7. Figure for the results of task 2.

3.1 Submission Validation

In the final leaderboard, each participant is only allowed to have one submission. We validate each submission and examine the number of submissions from each team. If a team has more than one submission, we keep the latest one and remove the rest from the leaderboard. Note that these removed submissions will remain on the RRC portal for reference, since they also provide important aspects into this research field. We adopt the following rules to determine the authorship of each submission:

- **user_id**: If two submissions have the same user_id field, it means they are submitted by the same RRC user account and thus should be from the same team.
- **method description**: Participants are asked to provide descriptive information of their submissions, including authors, method details, etc. If two submissions have strictly almost identical author list and method description, we consider them to be from the same team.

3.2 Task 1 Methodology

Task 1 in our competition, i.e. Hierarchical Text Detection, is a novel task in the research field. There are no existing methods that participants can refer to. Even the previous work Unified Detector [19] can only produce line and paragraph outputs but no word-level results. Among the 8 submissions in Task 1 which have disclosed their methods, we observed that 5 of them develop '*multi-head plus postprocessing*' systems. These methods treat words, lines, and paragraphs as generic objects, and train detection or segmentation models to localize these three levels of text entities in parallel with separate prediction branches for each level. In the post-processing, they use IoU-based rules to build the hierarchy in the post-processing step, i.e. assigning words to lines and lines to paragraphs. The most of the top ranking solutions belong to this type of methods. One submission (from the SCUT-HUAWEI team) adopts a cascade pipeline, by first detecting words and then applying LayoutLMv3 [26] to cluster words into lines and paragraphs. The *Hierarchical Transformers for Text Detection* method develops a unified detector similar to [19] for line detection and paragraph grouping and also a line-to-word detection model that produces bounding boxes for words. Here we briefly introduce the top 2 methods in this task:

Upstage KR team ranks 1st place in Task 1, achieving an H-PQ metric of 76.85%. It beats the second place by almost 6% in the H-PQ metric. They implemented a two-step approach to address hierarchical text detection. First, they performed multi-class semantic segmentation where classes were word, line, and paragraph regions. Then, they used the predicted probability map to extract and organize these entities hierarchically. Specifically, an ensemble of UNets with ImageNet-pretrained EfficientNetB7 [8]/MitB4 [7] backbones was utilized to extract class masks. Connected components were identified in the predicted mask to separate words from each other, same for lines and paragraphs. Then, a word was assigned as a child of a line if the line had the highest IoU with the word compared to all other lines. This process was similarly applied to lines and paragraphs. For training, they eroded target entities and dilated predicted entities. Also, they ensured that target entities maintained a gap between them. They used symmetric Lovasz loss [9] and pre-trained their models on the SynthText dataset [24].

DeepSE X Upstage HK team ranks 2nd in the leaderboard. They fundamentally used DBNet [6] as the scene text detector, and leveraged the oCLIP [5] pretrained Swin Transformer-Base [4] model as the backbone to make direct predictions at three different levels. Following DBNet, they employed Balanced

Cross-Entropy for binary map and L1 loss for threshold map. The authors also further fine-tuned the model with lovasz loss [9] for finer localization.

3.3 Task 2 Methodology

Task 2, i.e. Word-Level End-to-End Text Detection and Recognition, is a more widely studied task. Recent research [2,15] focuses on building end-to-end trainable OCR models, as opposed to separately trained detection and recognition models. It's widely believed that end-to-end models enjoy shared feature extraction which leads to better accuracy. However, the results of our competition say otherwise. The top 2 methods by the **Upstage KR team** and **DeepSE End-to-End Text Detection and Recognition Model team** are all separately trained models. There are two end-to-end submissions. The **unified_model team** applies a deformable attention decoder based text recognizer and ranks 3th place. Here we briefly introduce the top 2 methods in this task:
Upstage KR team uses the same task 1 method for detecting words. For word-level text recognition, they use the ParSeq [1] model but replace the visual feature extractor with SwinV2 [3]. The text recognizer is pretrained with synthetic data before fine-tuning it on the HierText dataset. They use an in-house synthetic data generator derived from the open source SynthTiger [25] to generate word images using English and Korean corpus. Notably, they generate 5M English/Korean word images with vertical layout, in addition to 10M English/Korean word images with horizontal layout. For the final submission, they use an ensemble of three text recognizers for strong and stable performance.
DeepSE End-to-End Text Detection and Recognition Model team also uses the ParSeq [1] model as their recognizer. They point out that, in order to make the data domain consistent between the training and inference stages, they run their detector on training data, and then crop words using detected boxes. This step is important int adapting the training domain to the inference domain. This trick essentially improves their model's performance.

4 Discussion

In the Hierarchical Text Detection task, the original Unified Detector [19] can only achieve PQ scores of 48.21%, 62.23%, 53.60% on the words, lines, and paragraphs respectively. The H-PQ score for Unified Detector is only 54.08%, ranking at 10th place if put in the competition leaderboard. The winning solution exceeds Unified Detector by more than 20%. These submissions greatly push the envelope of state-of-the-art Hierarchical Text Detection method. However, current methods are still not satisfactory. As shown in Fig. 6, we can easily notice that for all methods, word PQ scores are much higher than line PQ scores, and line PQ scores are again much higher than paragraph PQ scores. It indicates that, line and paragraph level detections are still more difficult than word detection. Additionally, Fig. 8 shows that layout analysis performance is only marginally correlated with word detection performance, especially when outliers are ignored.

We believe there's still hidden challenges and chances for improvement in layout analysis. Furthermore, winning solutions in our competition rely on postprocessing which can be potentially complicated and error-prone. It's also important to improve end-to-end methods.

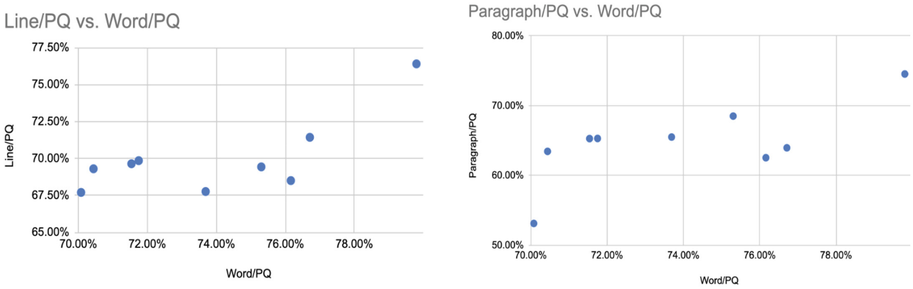


Fig. 8. Correlation between text levels. Each dot is a submission in the Task 1. **Left**: Correlation between word PQ and line PQ. **Right**: Correlation between word PQ and paragraph PQ.

The task 2 of our challenge is a standard yet unique end-to-end detection and recognition task. While it inherits the basic setting of an end-to-end task, it is based on a diversity of images which has high word density, and it has an unlimited character set. For this task, we see most of the submissions are two-stage methods, where the detection and recognition models are trained separately, and there's no feature sharing. These two-stage methods achieve much better performances than end-to-end submissions. This contrasts with the trend in research paper that favors end-to-end trainable approaches with feature sharing between the two stage. Therefore, we believe the HierText dataset can be a very useful benchmark in end-to-end OCR research. Another interesting observation for Task 2 is that, while most submissions achieve a tightness score of around 80%, the correlation between tightness scores and F1 scores and very low, with a correlation coefficient of 0.06. It could indicate that recognition is less sensitive to the accuracy of bounding boxes after it surpasses some threshold. This would mean that the mainstream training objective of maximizing bounding box IoU might not be the optimal target. For example, a slightly oversized bounding box is better than a small one which might miss some characters. With that said, a precise bounding box is still useful itself, which indicates the localization. Another potential reason is that bounding box annotation is not always accurate – it's always oversized because text are not strictly rectangular.

5 Conclusion

This paper summarizes the organization and results of ICDAR 2023 Competition on Hierarchical Text Detection and Recognition. We share details of competition motivation, dataset collection, competition organization, and result analysis. In total, we have 18 valid and unique competition entries, showing great interest from both research communities and industries. We keep the competition submission site open to promote research into this field. We also plan to extend and improve this competition, for example, adding multilingual data.

References

1. Bautista, D., Atienza, R.: Scene text recognition with permuted autoregressive sequence models. In: Computer Vision-ECCV,: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022, Proceedings, Part XXVIII, p. 2022. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19815-1_11
2. Ye, M., et al.: DeepSolo: let transformer decoder with explicit points solo for text spotting. arXiv preprint arXiv:2211.10772 (2022)
3. Liu, Z., et al.: Swin transformer v2: scaling up capacity and resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
4. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
5. Xue, C.: Language matters: a weakly supervised vision-language pre-training approach for scene text detection and spotting. In: Computer Vision-ECCV, et al.: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022, Proceedings, Part XXVIII, p. 2022. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19815-1_17
6. Liao, M., et al.: Real-time scene text detection with differentiable binarization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 07 (2020)
7. Xie, E., et al.: SegFormer: simple and efficient design for semantic segmentation with transformers. Adv. Neural Inf. Process. Syst. **34**, 12077–12090 (2021)
8. Tan, M., Le, Q.: Efficientnet: rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning, PMLR (2019)
9. Berman, M., Amal, R.T., Matthew, B.B.: The lovász-softmax loss: a tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
10. Long, S., He, X., Yao, C.: Scene text detection and recognition: the deep learning era. Int. J. Comput. Vis. **129**(1), 161–184 (2021)
11. Lee, J., et al.: Page segmentation using a convolutional neural network with trainable co-occurrence features. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), IEEE (2019)
12. Yang, X., et al.: Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
13. Ronen, R., et al.: GLASS: global to local attention for scene-text spotting. arXiv preprint arXiv:2208.03364 (2022)
14. Long, S., et al.: Textsnake: a flexible representation for detecting text of arbitrary shapes. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)

15. Qin, S., et al.: Towards unconstrained end-to-end text spotting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
16. Kittenplon, Y., et al.: Towards weakly-supervised text spotting using a multi-task transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
17. Liu, S., et al.: Unified line and paragraph detection by graph convolutional networks. In: Uchida, S., Barney, E., Eglin, V. (eds.) Document Analysis Systems. DAS 2022. LNCS, vol. 13237, pp. 33–47. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-06555-2_3
18. Wang, R., Yasuhisa, F., Ashok, C.P.: Post-ocr paragraph recognition by graph convolutional networks. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (2022)
19. Long, S., et al.: Towards end-to-end unified scene text detection and layout analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2022)
20. Li, C., et al.: StructuralLM: structural pre-training for form understanding. arXiv preprint arXiv:2105.11210 (2021)
21. Long, S., Cong, Y.: Unrealtext: synthesizing realistic scene text images from the unreal world. arXiv preprint arXiv:2003.10608 (2020)
22. Liao, M., et al.: SynthText3D: synthesizing scene text images from 3D virtual worlds. Sci. China Inf. Sci. **63**, 1–14 (2020)
23. Jaderberg, M., et al.: Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227 (2014)
24. Gupta, A., Andrea, V., Andrew, Z.: Synthetic data for text localisation in natural images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
25. Yim, M., Kim, Y., Cho, H.-C., Park, S.: SynthTIGER: synthetic text image GEneratoR towards better text recognition models. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) ICDAR 2021, Part IV. LNCS, vol. 12824, pp. 109–124. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86337-1_8
26. Huang, Y., et al.: LayoutLMv3: pre-training for document AI with unified text and image masking. arXiv preprint arXiv:2204.08387 (2022)
27. Kuznetsova, A., et al.: The open images dataset v4. Int. J. Comput. Vis. **128**(7), 1956–1981 (2020)
28. Singh, A., et al.: TextOCR: towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)
29. Krylov, I., Sergei, N., Vladislav, S.: Open images v5 text annotation and yet another mask text spotter. In: Asian Conference on Machine Learning, PMLR (2021)
30. Kirillov, A., et al.: Panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2019)