Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 1209–1216
Marseille, 20-25 June 2022

CAMIO: A Corpus for OCR in Multiple Languages

Michael Arrigo[1], Stephanie Strassel[1], Nolan King[2], Thao Tran[2], Lisa Mason[2]
[1]Linguistic Data Consortium, University of Pennsylvania, 3600 Market Street, Suite 810, Philadelphia, PA 19104 USA
[2]U.S. Dept. of Defense, MD, USA
micarrigo@gmail.com, strassel@ldc.upenn.edu

Abstract

CAMIO (Corpus of Annotated Multilingual Images for OCR) is a new corpus created by Linguistic Data Consortium to serve as a resource to support the development and evaluation of optical character recognition (OCR) and related technologies for 35 languages across 24 unique scripts. The corpus comprises nearly 70,000 images of machine printed text, covering a wide variety of topics and styles, document domains, attributes and scanning/capture artifacts. Most images have been exhaustively annotated for text localization, resulting in over 2.3M line-level bounding boxes. For 13 of the 35 languages, 1250 images/language have been further annotated with orthographic transcriptions of each line plus specification of reading order, yielding over 2.4M tokens of transcribed text. The resulting annotations are represented in a comprehensive XML output format defined for this corpus. The paper discusses corpus design and implementation, challenges encountered, baseline performance results obtained on the corpus for text localization and OCR decoding, and plans for corpus publication.

Keywords: OCR, multilingual, collection, text localization, reading order, transcription

1. Introduction

CAMIO (Corpus of Annotated Multilingual Images for OCR) is a new corpus developed by Linguistic Data Consortium to serve as a resource to support the development and evaluation of optical character recognition (OCR) and related technologies for 35 languages across 24 unique scripts. The corpus was designed to address gaps in language and script coverage from existing OCR corpora (Huang et al, 2021), and to support future evaluation of OCR capabilities through a systematically constructed data set.

Data consisting of machine printed text was collected for each of 35 language-script pairs, with up to 2,500 distinct documents (pages) per language. Data collection encompassed two primary techniques: harvesting found data (existing images or scanned documents) from the web, and collecting images of machine printed text from data donations by the crowd. Text localization annotation was applied to the collected images in order to produce bounding boxes for each line of printed text, along with reading order annotation. For 13 of the language-script pairs, a subset of the collected data was also subject to manual transcription. The resulting corpus contains a total of 69,440 images, labeled with over 2.34 million bounding boxes around lines of printed text as well as reading order. Across 13 of the languages, 15,724 images have been transcribed, yielding nearly 2.4 million tokens in all. The resulting corpus contains source images along with annotations in a unified XML format defined for this effort, as well as metadata about each document.

2. Data Requirements

Data was collected and annotated for 35 language-script pairs, and a subset of the data was transcribed for 13 of these language-script pairs. The languages in this corpus were selected to represent a variety of scripts, ranging from those unique to a single language to those used for many languages. Table 1 lists the language-script pairs present in the CAMIO corpus. The script for each language is shown in parentheses, and transcription languages are denoted with bold font.

As originally conceived, the corpus would comprise 2,500 images of machine printed text per language, with 1,250 images transcribed per transcription language. The planned data volume was revised after collection began to target a variable number of collected images per language, retaining the goal of 1,250 images transcribed per transcription language.

Amharic (Ge'ez)	Malayalam (Malayalam)
Arabic (Arabic)*	Maldivian (Thanna)
Armenian (Armenian)	Oriya (Oriya)
Bengali (Eastern Nagari)	Pashto (Arabic)
Burmese (Burmese)	**Russian (Cyrillic)***
Cambodian (Khmer)	Sinhalese (Sinhala)
Chinese (Simplified)*	Swahili (Latin)
Dari (Arabic)	Tagalog (Latin)
English (Latin)*	**Tamil (Tamil)***
Farsi (Arabic)*	Telugu (Telugu)
Georgian (Georgian-Mkhedruli)	**Thai (Thai)***
Greek (Greek)	Tibetan (Tibetan)
Hebrew (Hebrew)	Tigrinya (Ge'ez)
Hindi (Devanagari)*	Ukrainian (Cyrillic)
Hungarian (Latin)	**Urdu (Arabic)***
Japanese (Japanese)*	Uyghur (Arabic)
Kannada (Kannada)*	**Vietnamese (Latin)***
Korean (Hangul)*	

Table 1: CAMIO language-script pairs

2.1 Data Properties

Existing OCR corpora are limited in the number of images collected for each language-script pair, or they are collected under specific constraints. For example, the MDIW-13 multiscript document database (Ferrer et al., 2019) consists of images from local newspapers and magazines. Other popular corpora such as the Scanned Receipts OCR and Key Information Extraction (SROIE) dataset (Huang et al., 2019), the Robust Reading Challenge-Multilingual Text (RRC-MLT) dataset (Nayef et al., 2019), and the Robust Reading Challenge-Large-Scale Street View Text (RRC-

LSVT) dataset (Sun et al., 2019) were designed to support OCR research for specific domains.

In contrast, CAMIO was designed to support OCR development and evaluation across a broad set of document domains. The data come from a variety of sources, including newspapers, books, journals, web pages, and other sources. The corpus includes both scanned documents and images of printed texts and is categorized with respect to basic metadata, including language, script, genre, domain, content attributes, and document scanning/capture artifacts.

Collection genres of interest included books, cards/slides, periodicals, records, scene text and webpages. Note that the webpage genre refers to images/screenshots of webpage as displayed on a digital device, and the goal was for the corpus to include as much variety as possible in terms of screenshot type and dimensions. During the collection phase, annotators were instructed to provide screenshots taken on desktop computers, laptops, tablets and mobile devices, as well as screenshots containing dual or projected screens, provided that both screens contained data in the relevant language.

Each genre was required to have data in a variety of document domains, defined as follows:

- Text-heavy documents: The document contains paragraphs or sizeable portions of uninterrupted text; e.g. books, documents, newspapers, webpages
- Unconstrained text: The document contains text that is structured, but that appears at various parts of the document in varying orientations, sizes, and/or lengths; e.g. business cards, flyers, newspaper ads, weblogs
- Overlaid text: The document contains text that is superimposed over another element, such an image; e.g. closed captioning, Internet memes, timestamps, webpage ads
- Diagrams with text: The document contains at least one diagram, where a diagram is a simplified schematic illustration; e.g. figures, graphs, maps, PowerPoint slides
- Varied content: The document contains varied writing styles, such as technical, non-technical, and vernacular language; e.g. discussion forums, journals, PowerPoint slides

The corpus was also required to contain a variety of attributes, including the following:

- Tables: The document contains at least one table, where a table is defined as a collection of data displayed in rows and/or columns, with or without borders
- Multi-column: Any portion of the document's text is displayed in multiple columns
- Fielded text: Any portion of the document contains text that is displayed with a visual separator, such as a colon or other indicator
- Multi-script: The document contains more than one script, whether the scripts are used to write the same language or different languages
- Multilingual: The document contains more than one language, whether or the not the languages are written in the same script
- Text with images: The document contains at least one image along with the text, regardless of image size or location
- Handwriting: The document contains at least some handwriting
- Other complex layout: The layout of the document is such that the document could not easily be copied and pasted, due to the presence of such features as complex columns, multiple text orientations, etc. E.g. tweets, maps

Finally, the corpus was designed to contain variety in terms of document scanning/capture artifacts, including:

- Varied DPI/resolutions: Documents should be of varying DPI/resolutions or levels of detail, as long as the text is readable
- Color/grayscale/black & white: Documents should be of varying values with respect to color, grayscale, black and white
- Warping: The document is curved, wrinkled, or otherwise damaged, such as from scanning a book near the spine
- Text runoff: A portion of the document's text (words or characters) is cut off in some way, whether in the center of the document or at the document's physical borders
- Occluded text: A portion of the document's text is covered or hidden in some way, such as by markings are artifacts (e.g. stamps, stickers, etc.)
- Distance of document from camera: Documents should be of varying ranges from the camera, as long as the text is readable
- Perspective: Documents should be of varying angles with respect to the camera
- Lighting: Documents should be of varying lighting conditions, such as indoor/outdoor lighting, lamps, shadows, etc.
- Skew, slant, rotation: The text of the document is neither parallel nor at right angles to the document's physical boundaries
- Noise: The document contains random variations of brightness or color (e.g. graininess)

Beyond these defined feature variables, the corpus as a whole was also required to have broad topical variety including both formal and informal content, though topic domain was not manually annotated.

The collection effort attempted to ensure that there was some representation from every feature category, and ideally from every plausible combination of categories, for every language in the collection. However, we did not attempt to achieve a perfect balance or try to ensure identical distribution of features across all languages, since such a tightly controlled distribution would have been cost-prohibitive. Moreover, some combinations of categories were difficult or impossible to achieve (e.g. overlaid text + tables from the scene text genre on a formal, technical topic is an unlikely combination).

3. Collection

To support data collection, LDC relied on a combination of data scouting by trained annotators and crowdsourcing. Data scouting was carried out using a custom user interface designed by LDC, and crowdsourcing was carried out via HITs posted on Amazon Mechanical Turk. For both approaches, URLs for existing images on the web were provided along with feature labels so that overall feature variety could be monitored and used to inform decisions about subsequent collection needs. All images had to contain machine printed text in one of the 35 languages.

Once downloaded via LDC's web data collection system, the images received a unique identifier and were added to a comprehensive tracking database where metadata values were recorded, including the image's original URL, a unique source identifier (e.g. website name), image provenance (data scouting or crowd), language, and the various feature labels either assigned by data scouts or associated with the particular crowdsourcing task.

The original collection goal was to have images obtained using three different data collection techniques, with the understanding that the relative proportion of each method would vary from one language to the next. These methods included (1) harvesting existing data from the web, (2) creating machine printed images through scanning, and (3) receiving image contributions from the crowd. The expectation was that Method 1 would be the primary approach, with Methods 2 and 3 adopted to address shortfalls as needed.

Method 1 was preferred because it was the easiest to implement, relying on LDC-trained data scouts to search the web for images that not only satisfied our feature variety requirements, but were from websites whose terms of use were compatible with our intended use of the data. It was also straightforward to integrate Method 1 with LDC's existing web data collection system.

As originally conceived, Method 2 would have involved locating existing repositories of texts from libraries, universities and other sources and either scanning or photocopying the text to create data with the desired scanning/image capture artifacts. As we began testing this method it became apparent that it may not yield a sufficient number of documents for several of the project's languages, especially where collection involved US-based annotators. Since the first method allowed us to target collection of already-scanned images with a variety of artifacts, the second method became less necessary and it was ultimately abandoned.

When Method 1 proved insufficient to meet data volume requirements for a given language, we applied Method 3. As originally planned, Method 3 involved asking crowdworkers to photograph or scan examples of machine printed text that already existed in their own environment. Obtaining such image uploads from the crowd proved challenging, since the availability of fluent crowdworkers for many of the project's diverse languages was extremely limited. Instead, LDC asked the crowd to identify existing images on the web in a way that resembled Method 1. Initially, we also planned to not impose specific requirements regarding content attributes and scanning/image capture artifacts, instead preferring to sample the naturally-occurring variety from the crowd. However, it became necessary to incorporate some targeted collection with more explicit and detailed requirements in order to address specific gaps in the collection. Method 3 proved to be most fruitful for Bengali, Cambodian, Dari, Hebrew, Maldivian, Sinhalese and Uyghur, while Method 1 had the best yield for the remaining languages.

The collected images represented various file formats including JPG, PNG and TIF. For consistency, all documents were converted to a standard file format (PNG) after collection and before auditing and annotation; both the original file format and the converted file format were retained. All documents, regardless of collection method, were also assigned a normalized corpus filename. Original URL, original filename, standardized filename, and other document metadata resulting from image processing was tracked in the comprehensive corpus database.

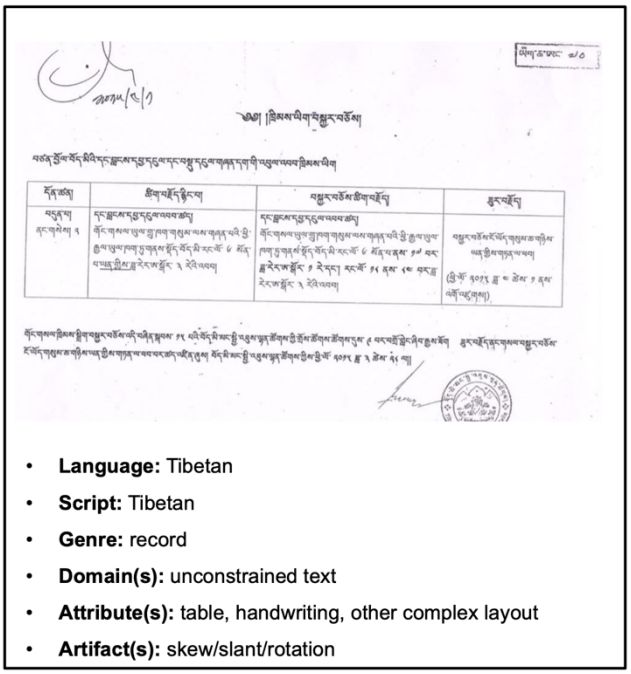
The originally planned size of the corpus was 87,500 documents in all, comprising 2,500 distinct documents per language, where each document is one page. Actual collection resulted in many more text-heavy images than anticipated, but text-heavy images are much more time consuming to annotate. To accommodate this additional annotation effort, the overall corpus size was modified to include 2,500 images per transcription language and 2,000 images per non-transcription language. Additional data volume reductions were required for nine languages where annotator retention, as well as data availability proved to be challenging. The final corpus requirements were then established as follows:

- 2500 images each: Arabic, Chinese, English, Farsi, Hindi, Japanese, Kannada, Korean, Russian, Tamil, Thai, Urdu, Vietnamese
- 2000 images each: Amharic, Armenian, Burmese, Dari, Greek, Hungarian, Malayalam, Odia, Pashto, Swahili, Tagalog, Telugu, Ukrainian
- 1500 images each: Bengali, Cambodian, Georgian
- 1000 images each: Hebrew, Tibetan, Tigrinya
- 500 images each: Maldivian, Sinhalese, Uyghur

4. Auditing

Auditing consisted of vetting the quality of each collected document and manually labeling feature metadata (genre, document domain, attributes, artifacts). While the majority of CAMIO corpus documents were scouted by native speakers, auditing was performed by specialist annotators at LDC who were skilled in metadata labeling but who were not necessarily native speakers of the document language. When a given document had not been collected by a native speaker (e.g. some documents sourced by crowdworkers using Method 3), that document was always assigned to a trained native speaker for auditing so that the auditor could also verify the language and script used. The outcome of auditing was a set of judgments for each document specifying the language, script, genre, domain, attribute and artifact, as illustrated in Figure 1. Auditing was carried out in parallel with data collection, so that auditing results could inform future data collection targets, and so that any

data scouting problems could be identified and resolved quickly.



- **Language:** Tibetan
- **Script:** Tibetan
- **Genre:** record
- **Domain(s):** unconstrained text
- **Attribute(s):** table, handwriting, other complex layout
- **Artifact(s):** skew/slant/rotation

Figure 1: Tibetan image with audit judgments

After collection and auditing, the final collection yield was 69,440 documents.

5. Ground Truth Annotation

Bounding box annotation, along with a specification of reading order for the bounding boxes created, was applied to a portion of the collected data. Where possible, the data selected for annotation were proportionally representative of the overall collection.

5.1 Text Localization

For the majority of images, bounding boxes were drawn around each line of machine printed text. The bounding box for each line consists of a unique id, its contents (e.g. text, features), and coordinates indicating the location of the line on the page.

Annotation was performed using a custom web-based user interface developed by LDC for this effort. The CAMIO user interface, created within LDC's existing Webann framework (Wright et al., 2012), allowed annotators to draw 4-point bounding boxes and to specify attributes for each box, including the presence of non-target languages, other scripts, or illegible content.

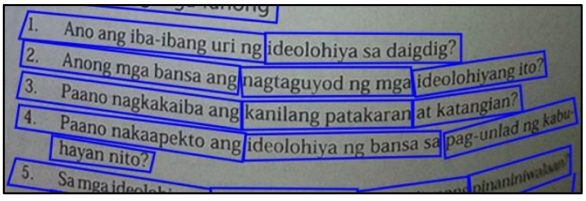


Figure 2: Multiple bounding boxes for curved lines

To satisfy text localization annotation requirements, there was sometimes the need for multiple bounding boxes over a single line of machine-printed text (e.g. when there is curvature in the line), as shown in Figure 2.

A total of 59,990 images were subject to text localization, resulting in 2,340,205 boxes for an average of 39 boxes per image.

5.2 Reading Order

Explicit reading order was indicated by applying a next_id tag to each bounding box. The value for next_id could be "NONE" when the current lineZone is the last in a sequence's reading order (i.e. the final lineZone in the document). All lineZone ids are unique within a given document.

Annotators were instructed to follow the natural, logical flow of content within each section of the document, and not to break the flow of the text by crossing between sections. In cases where a single line of text was divided among multiple bounding boxes, reading order is applied to these intra-line bounding boxes the same way it is applied to the inter-line bounding boxes.

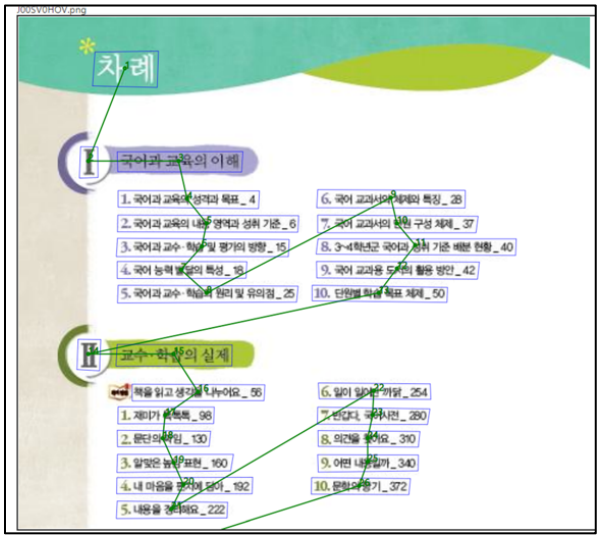


Figure 3: Reading order annotation

A CAMIO image labeled for reading order appears in Figure 3. Reading order annotation was carried out within the same CAMIO user interface that was used for text localization, and annotators could flag and correct errors in bounding boxes and/or feature labels before specifying reading order. After reading order annotation was complete we applied a number of automatic checks to ensure that all bounding boxes had been labeled, with appropriate numbering.

6. Transcription

For each of the thirteen transcription languages, a subset of 1250 annotated images was selected for orthographic transcription, with one transcript for each line of machine printed text for which a bounding box had been produced. All transcription annotators were native speakers of the document language.

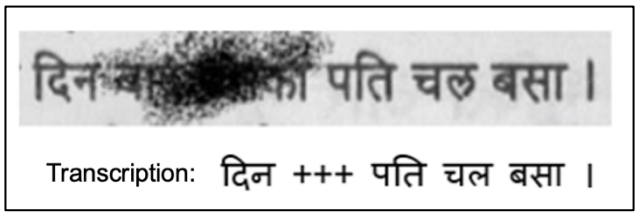
Transcription guidelines included the following guiding principles:

- Transcribe as accurately and faithfully as possible
    - Use native orthography matching the image script
    - Retain capitalization and punctuation
    - Transcribe exactly what is written and do not attempt to correct misspellings, grammatical mistakes or other errors
    - Do not transcribe stylistic features
- Only transcribe content of bounding box
- Follow the natural reading order

Within the transcript itself, annotators also used markup to specify various features, including:

- Word or symbol is not in expected script and could not be typed
- Word is in script but contains uncommon characters, diacritics, or features that could not be keyboarded
- Word is not fully readable by itself but could be understood from context
- Word is in script but is unreadable

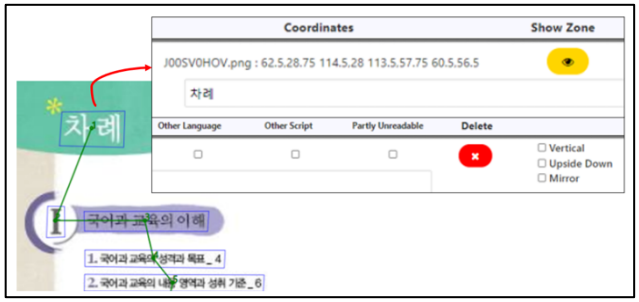
For instance, Figure 4 shows transcription markup applied to a partially legible line of Hindi text.



Transcription: दिन +++ पति चल बसा ।

Figure 4: Transcription markup for illegible text

In addition to providing the transcript for each bounding box, annotators labeled the orientation of the text during this stage of annotation. There were three orientation flags: vertical, upside down, and mirror. For the three Arabic script transcription languages (Arabic, Farsi, and Urdu), the default directionality is right-to-left. For Chinese, Japanese, and Korean, the default directionality is left-to-right when written horizontally and right-to-left when written vertically. Even though both are considered "normal," vertical text was still flagged as such during transcription for these languages (and the right-to-left directionality was captured by the reading order). The remaining seven transcription languages are written left-to-right by default.



J00SV0HOV.png : 62.5.28.75 114.5.28 113.5.57.75 60.5.56.5

차례

Figure 5: Transcription in the CAMIO user interface

Transcription and specification of text orientation were performed in the same custom user interface used for the other CAMIO annotation tasks, as illustrated in Figure 5.

In all, 15,724 documents comprising 311,619 images (boxed lines) were transcribed, yielding a total of 2,352,411 tokens.

7. Quality Control

Following initial annotation, a quality control (QC) pass was applied to identify and correct common errors. QC included exhaustive review of a portion of the annotations produced by every annotator (approximately 5%), as well as a portion of the data from all category-combinations for each language (also around 5%). Manual spot checks on the remainder of the data were also performed.

The CAMIO pipeline was also designed to allow each stage of annotation to serve as a kind of quality check on the prior stage, since all annotation tasks were performed within the same custom user interface. In particular, reading order annotation served as a full review pass on text localization and feature labeling. All reading order annotators were initially trained on text localization themselves, and they were identified as having been the strongest annotator(s) for their languages. During reading order annotation they were expected to identify and correct any errors in text localization and feature annotation before applying reading order. They also helped to catalog common mistakes in text localization, which enabled a constructive feedback loop for text localization annotators and allowed for enhancement of training methods and annotation guidelines. To maximize the benefit of this staged quality control approach, annotators were prevented from labeling documents for which they had produced annotations in a prior stage.

In addition to task-specific manual and automatic quality control, a final automatic quality pass was performed on the full set of annotations to ensure format compliance, to check that no encoding issues had been introduced at any of the various stages of the annotation pipeline, and to normalize any encoding anomalies. LDC staff external to the project team also completed an extensive sanity check protocol to validate the final corpus package with respect to a wide range of potential data integrity issues.

8. Challenges and Solutions

One of the biggest challenges in developing the CAMIO corpus resulted from the sheer size and scope of the corpus. Collecting and annotating this volume of data in 35 languages, several of them low resource, required a significant investment in annotator supervision and training. We relied heavily on project assistants to help recruit and screen potential annotators, and to maintain constant communication with contract annotators to ensure that their quality met requirements, to boost their productivity when it started to slip, and to handle compensation and administrative paperwork for hundreds of individuals. The large and distributed annotation team also meant that quality control was especially critical.

Another challenge was that for some (low resource) languages, the number of qualified native speakers who had

availability and work eligibility was extremely limited. Ultimately we were able to identify a sufficient number of annotators to meet the project's needs, relying in part on recruitment of native speaker students at Penn, but the process was laborious and time consuming.

Finally, the prevalence of text-heavy images in the corpus, as illustrated by Figure 6, significantly impacted annotation efficiency for both text localization and reading order.

Figure 6: Text-heavy document

The number of lines in text-heavy images not only required creation of more bounding boxes, but also made it more difficult to create accurate boxes because of the crowded nature of text on the page. We took a number of steps to address this challenge. First, we made some simple improvements to the user interface to boost its functionality and user-friendliness, requiring less effort from the annotator to label each image. Second, we did a light auditing round over the data to identify particularly text-heavy documents and sequestered those, removing them from the annotation (and transcription) pipelines. Finally, we conducted supplemental data scouting to identify additional documents for annotation that were less challenging with respect to text content so that we could still meet the annotation targets for languages affected by this issue.

9. Technology Baselines

The CAMIO corpus was designed to support a range of evaluation tasks including script ID, language ID, text localization, OCR decoding, keyword search and OCR end-to-end evaluations. To gauge the complexity of the images collected for CAMIO and the suitability of the data to support technology development goals, we produced baseline performance results for the text localization and OCR decoding tasks using the open-source Tesseract OCR engine (Smith et al., 2009) for a subset of transcribed images.

9.1 Baseline Experiments and Results

For the set of transcribed images, data from each of the relevant 13 languages was partitioned using a 50/10/40 train/validation/test split so that there were nominally 625 train, 125 validation, and 500 test images per language.

CAMIO Test Set	F1 Score	Precision	Recall
Arabic	27.5%	22.6%	35.3%
Chinese (Simplified)	15.6%	13.0%	19.3%
English	17.3%	15.2%	20.1%
Hindi	25.2%	22.1%	29.4%
Japanese	13.5%	11.3%	16.9%
Kannada	36.9%	30.4%	47.2%
Korean	27.0%	21.9%	35.4%
Farsi	23.8%	18.5%	33.4%
Russian	19.9%	18.2%	22.0%
Tamil	30.4%	26.2%	36.3%
Thai	16.0%	15.0%	17.1%
Urdu	27.5%	22.9%	34.5%
Vietnamese	18.3%	14.0%	26.5%

Table 2: Tesseract OCR text localization performance on CAMIO

Table 2 shows text localization results using the Tesserocr python wrapper for Tesseract 4.0.0. Precision, Recall, and F1 are standard metrics for text localization in the field of computer vision and document analysis. We provide these scores for the test partitions in each language, for all three metrics, with higher scores indicating better performance. The text localization baseline results were obtained by comparing Tesserocr SINGLE_BLOCK output with the CAMIO ground truth annotations. Precision was calculated by comparing how often Tesseract text boxes overlapped with the ground truth text shapes with an intersection over union (IoU) score of 0.5 over the total number of Tesseract boxes found. Recall shows the same number of Tesseract text boxes that meet the 0.5 IoU threshold over the number of ground truth boxes. The F1 score is the harmonic mean of precision and recall, F1 = 2* Precision * Recall / (Precision + Recall), and was used to find a balance between the two other measurements.

Table 3 show results for OCR decoding using Tesserocr. We provide the Character Error Rate (CER) for each language, which is a standard performance metric based on the edit (Levenshtein) distance between the ground truth transcript and the OCR model output. It computes the minimum number of edits it would take to modify the model output to match the ground truth divided by the number of characters in the ground truth.

CAMIO Test Set	Character Error Rate (CER)
Arabic	24.0%
Chinese (Simplified)	34.9%
English	12.1%
Hindi	16.4%
Japanese	39.8%
Kannada	12.4%
Korean	18.9%
Farsi	23.0%
Russian	15.3%
Tamil	24.5%
Thai	18.1%
Urdu	68.2%
Vietnamese	17.0%

Table 3: Tesseract OCR decoder performance on CAMIO

These scores per language reflect the accumulated number of edits needed for each line over the language dataset over the total number of characters in the ground truth for that language. In addition, normalization was performed on both the ground truth and the OCR output prior to measuring the CER, which is another standard procedure to make more meaningful comparisons. For each transcript of ground truth, the following normalization procedures were taken:

1. Conversion of characters to lower-case
2. Removal of extra spaces (down to single spaces)
3. Removal of punctuation
4. Apply Unicode compatibility decomposition, followed by canonical composition (NFKC)

CER is an error rate metric, with lower scores indicating better system performance. Considering that significant performance degradations can be noticed with CER rates as low as 5% (Bazzo et al., 2020), we observe that CAMIO presents a significant challenge across all 13 test sets with the three worst scores from Chinese (Simplified), Japanese, and Urdu. The high CERs for both the Chinese and Japanese test sets likely reflect the presence of vertical text instances (characters written right-side-up but positioned above or below each other instead of to the left or right of each other), for which OCR models have been known to struggle. The poor performance of the Urdu test set can be likely attributed to the presence of stylized fonts, including Nastaliq, which is very common in Urdu writing and also known to cause performance degradations for OCR models.

The combination of poor Tesseract performance in both the text localization and OCR decoding tasks indicates that the CAMIO corpus presents new and challenging data to support research to further improve the OCR pipeline for the kind of complex and noisy document images targeted in the corpus.

10. Data Distribution and Conclusion

The final CAMIO corpus consists of the collected image data in its original format, along with a normalized PNG image used as input to annotation. Annotation and document metadata is presented in a unified XML format defined by LDC for this effort. There is one XML file per image containing the original source URL, source and language/script info from the Collection task, document-level features from Auditing, line zone numbers from Reading Order, bounding box coordinates from Text Localization, line-level features from Reading Order, line-level orientation features from Transcription, and the transcript itself. The corpus also includes collection, auditing and annotation guidelines used for each stage of corpus development.

	Collection and Annotation	Transcription
Languages	35	13
Documents	69,440	16,246
Images Annotated or Transcribed	59,990	16,246
Line Boxes	2,340,205	323,668
Average Boxes per Image	39	20
Tokens	n/a	2,431,141

Table 4: CAMIO corpus summary

The Corpus of Annotated Multilingual Images for OCR (CAMIO) reflects an exciting new resource for research and evaluation in computer vision and document analysis. The corpus covers 35 languages across 24 unique scripts, including some for which there are no known existing OCR-related resources. It comprises nearly 70,000 images, most of which have been annotated for text localization and reading order, resulting in over 2.3M bounding boxes around lines of machine printed text. For 13 of the languages, over 16,000 images have also been transcribed, yielding over 2.4M tokens of text data. The corpus results are summarized in Table 4.

The CAMIO corpus will appear in LDC's catalog starting in 2022, with one release planned for transcribed languages and another for untranscribed languages, each with defined train/dev/test partitions. A portion of the data will remain unpublished for use as test data in future technology evaluations, and additional annotation of the CAMIO corpus is planned including document zoning, translation and additional transcription.

11. Acknowledgements

The authors gratefully acknowledge the contributions of annotation coordinators Justin Mott and Neil Kuster, technical infrastructure developers Chris Caruso, Jonathan Wright, Alex Shelmire and David Graff, and the work of hundreds of annotators who contributed to this corpus.

12. Bibliographical References

Bazzo, G.T., Lorentz, G.A., Suarez Vargas, D., Moreira, V.P. (2020). Assessing the Impact of OCR Errors in Information Retrieval. In *Advances in Information Retrieval*. ECIR 2020. Lecture Notes in Computer Science, vol 12036.

Belay, B., Habtegebrial, T., Meshesha, M., Liwicki, M., Belay, G., Stricker, D. (2020). Amharic OCR: An End-to-End Learning. Applied Sciences. 10(3): 1117.

Huang, Z., Chen, K., He, J., Bai, X., Karatzas, D., Lu, S., Jawahar, C.V. (2019). ICDAR2019 Competition on

Scanned Receipt OCR and Information Extraction. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1516-20.

Ferrer, M.A., Das, A., Diaz, M., Carmona-Duarte, C., Pal, U. (2019). MDIW-13 MultiScript Document Database.

Guyon, I., Haralick, R.M., Hull, J.J., Phillips, I.T. (1997). Data sets for OCR and document image understanding research. *Handbook of character recognition and document image analysis*, pp.779-799.

Karatzas, D., Robles, S., Gomez, L. (2014). An on-line platform for ground truthing and performance evaluation of text extraction systems. *2014 11th IAPR International Workshop on Document Analysis Systems*, pp. 242-246.

Mihov, S. *et al.* (2005). A corpus for comparative evaluation of OCR software and postcorrection techniques." *Eighth International Conference on Document Analysis and Recognition (ICDAR'05)*, pp. 162-166.

Nayef, N. *et al.* (2019). ICDAR2019 robust reading challenge on multi-lingual scene text detection and recognition—RRC-MLT-2019. *2019 International Conference on Document Analysis and Recognition (ICDAR),* pp. 1582-1587.

Smith, R., Antonova, D., Lee, D. (Jul 25, 2009) Adapting the Tesseract open source OCR engine for multilingual OCR. *Proceedings of the International Workshop on multilingual ocr*, pp. 1-8.

Sun, Y., Karatzas, D., Chan, C.S., Jin, L., Ni, Z., Chng, C., Liu, Y., Luo, C., Ng, CC., Han, J., Ding, E., Liu, J. (2019). ICDAR 2019 Competition on Large-Scale Street View Text with Partial Labeling - RRC-LSVT. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1557-62.

Wright, J., Griffitt, K., Ellis, J., Strassel, S., and Callahan, B. (2012). Annotation trees: LDC's customizable, extensible, scalable, annotation infrastructure. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation* (LREC 2012), pp. 479-485.