ScreenAI: A Vision-Language Model for UI and Infographics Understanding

Gilles Baechler, **Srinivas Sunkara**, **Maria Wang**, **Fedir Zubach**, **Hassan Mansoor**, **Vincent Etter**, **Victor Cărbune**, **Jason Lin**, **Jindong Chen**, **Abhanshu Sharma**
Google DeepMind
jdchen@google.com

Abstract

Screen user interfaces (UIs) and infographics, sharing similar visual language and design principles, play important roles in human communication and human-machine interaction. We introduce ScreenAI, a vision-language model that specializes in UI and infographics understanding. Our model improves upon the PaLI architecture with the flexible patching strategy of pix2struct and is trained on a unique mixture of datasets. At the heart of this mixture is a novel screen annotation task in which the model has to identify the type and location of UI elements. We use these text annotations to describe screens to Large Language Models and automatically generate question-answering (QA), UI navigation, and summarization training datasets at scale. We run ablation studies to demonstrate the impact of these design choices. At only 5B parameters, ScreenAI achieves new state-of-the-art results on UI- and infographics-based tasks (Multipage DocVQA, WebSRC, and MoTIF), and new best-in-class performance on others (ChartQA, DocVQA, and InfographicVQA) compared to models of similar size. Finally, we release three new datasets: one focused on the screen annotation task and two others focused on question answering.

1 Introduction

Infographics, such as charts, diagrams, illustrations, maps, tables, and document layouts have long been a cornerstone of effective communication, thanks to their ability to distill complex data and ideas into simple illustrations through arrangement of layouts, and visual cues. In the digital era, mobile and desktop UIs, sharing similar design principles and visual languages with infographics, facilitate human communication and human-machine interface with rich and interactive user experiences.

Although the above observation suggests an opportunity for a unified model, because of their complexity, infographics and UIs present a unique challenge to building a single model that can understand, reason, and interact on top of pictorial pixels. To address this challenge, we introduce ScreenAI, a Vision-Language Model (VLM) for comprehensive UI and infographics understanding, including tasks such as question-answering (QA) on infographics (charts, illustrations, maps, etc.), and element annotation, summarization, navigation, and QA on UIs. Our model combines the PaLI [Chen *et al.*, 2023b] architecture with the flexible patching mechanism of Pix2struct [Lee *et al.*, 2023] and handles vision tasks by recasting them as (text, image)-to-text problems. Figure 1 provides a high level description of the model architecture and Section 2.1 describes its components in more detail.

The main contributions of this work are multifold and greatly advance the field of digital content understanding:

- We propose ScreenAI, a Vision-Language Model (VLM), as a holistic solution that focuses on understanding UIs and infographics, taking advantage of their common visual language and design sophistication.
- We introduce a textual representation for UIs, which we use to teach our model how to understand UIs during its pretraining phase.
- We take advantage of this new UI representation and Large Language Models (LLMs) to automatically generate training data at scale.
- We define pretraining and fine-tuning mixtures which cover a wide spectrum of tasks in UI and infographic understanding.
- We release three evaluation datasets for tasks described in Section 4.2: Screen Annotation, ScreenQA Short, and Complex ScreenQA. These datasets enable the research community to utilize our textual representation and allow for a more comprehensive benchmarking of models for screen-based question answering.

These innovations position ScreenAI as the go-to VLM for any digital content understanding task, ranging from UIs to infographics, and beyond. At a modest size of 4.6 billion parameters, dated on January 17, 2024 [1] , our model exhibits state-of-the-art (SoTA) performance on three public infographics QA benchmarks, surpassing other models 10x or more in size. In other tasks, ScreenAI exhibits best-in-class, or close-to-best performance. We show in Section 5.2 that the model performance gets better as we increase its size, suggesting that there is a strong potential for further gains in performance by scaling up the model.

[1] The full paper submission deadline of IJCAI-24.

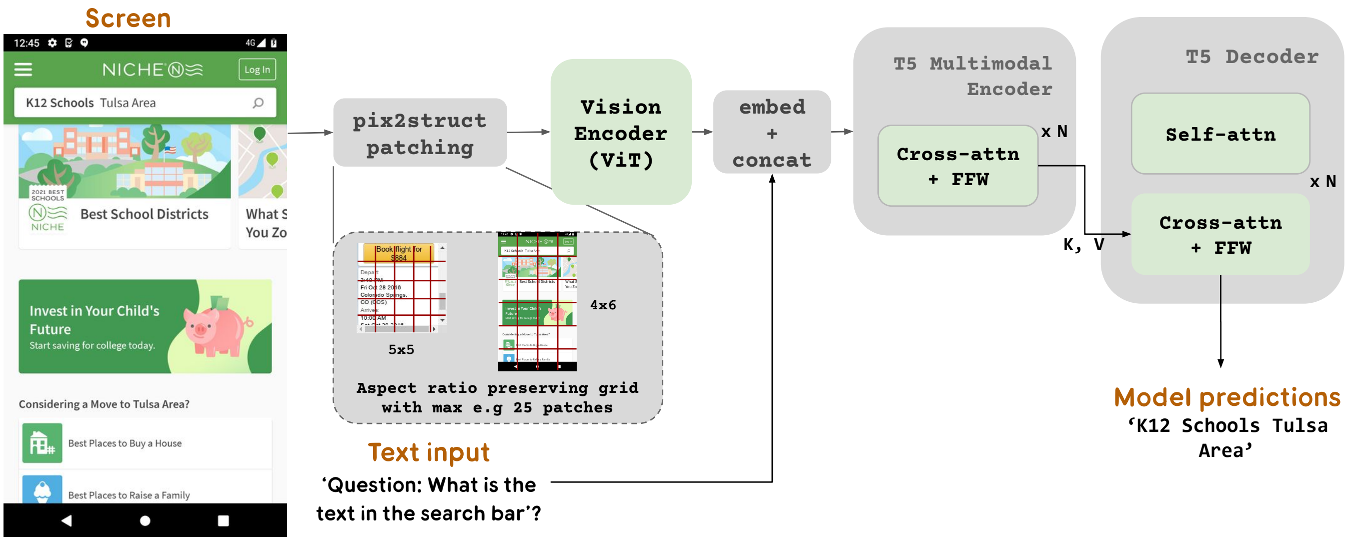


Figure 1: The overall architecture of our model. The model contains an image encoder followed by a multimodal encoder consuming embedded text and image features. The output of the multimodal encoder is fed to an autoregressive decoder to generate the final text output. This figure also illustrates pix2struct patching, where the grid size adapts to the aspect ratio and shape of the image.

1.1 Related Work

We identify three categories of closely related works.

Screen-Based UI Models. Until recently, most screen understanding efforts focused on well-defined tasks with a narrow scope. Examples include the detection of icons [Zang *et al.*, 2021] or various UI elements [Zhang *et al.*, 2021; Sunkara *et al.*, 2022; Li *et al.*, 2022a], together with their structure [Wu *et al.*, 2021]. Other notable works encompass the description of icons (widget captioning) [Li *et al.*, 2020], screen summarization [Wang *et al.*, 2021], and single-step navigation tasks [Wichers *et al.*, 2018; Li *et al.*, 2022b]. Another direction is to use LLMs to classify and describe UI elements [Gur *et al.*, 2022], or complete tasks [Nakano *et al.*, 2021; Rawles *et al.*, 2023; Deng *et al.*, 2023].

Generalist Foundation Models. The advent of large foundation models, particularly in the multimodal domain, has led to the development of versatile and unified models. These universal models excel in a broad spectrum of image understanding tasks formulated through natural language such as question-answering, image captioning, and object localization. (e.g. UniTAB [Yang *et al.*, 2022], OFA [Wang *et al.*, 2022], PaLI [Chen *et al.*, 2022; Chen *et al.*, 2023a; Chen *et al.*, 2023b], Flamingo [Alayrac *et al.*, 2022], or MaMMUT [Kuo *et al.*, 2023]). Foundational work also includes pix2seq [Chen *et al.*, 2021a], which recasts the object detection problem as a text prediction task.

Efficient Vision-Language Models. Closer to the domain of screen and document understanding, similar transformer-based [Vaswani *et al.*, 2017] architectures have been proposed for solving various document-understanding tasks (e.g. LayoutLMv3 [Huang *et al.*, 2022], Donut [Kim *et al.*, 2021], pix2struct [Lee *et al.*, 2023], MatCha [Liu *et al.*, 2022], UDOP [Tang *et al.*, 2023], or Spotlight [Li and Li, 2022]). Another example is VuT [Li *et al.*, 2021], which is made of a multimodal encoder, followed by a text decoder and a dedicated head for object detection tasks.

Other approaches like UIBert [Bai *et al.*, 2021], DocLLM [Wang *et al.*, 2023] perform screen- and document-understanding using additional textual data extracted from metadata like DOM or ancillary models like OCR.

In our paper, we introduce pre-training tasks along with a data generation schema using self-supervision and model-based annotation. Prior work with self-supervised learning tasks have typically been focused on one domain. For examples, pix2struct [Lee *et al.*, 2023], HTLM [Aghajanyan *et al.*, 2021] are focused on web-pages; ActionBert [He *et al.*, 2021], UIBert [Bai *et al.*, 2021] are focused on mobile apps, which can capture a subset of the elements like text and exclude hierarchy information. Our representation, inferred from only screen or image pixels, is applicable to a wide range of domains beyond web-pages and mobile apps, including documents, infographics, etc. Compared to prior work, our model achieves superior performance on downstream tasks. We hypothesize this is due to the positive transfer of performance when using screen, document and infographics data jointly in the pre-training mixture. Given the abundance of data in each of these domains, we believe future research in this direction can result in further improvements.

2 Methodology

2.1 Architecture

Our model architecture as shown in Figure 1 is inspired by the architecture of the PaLI family of models [Chen *et al.*, 2022; Chen *et al.*, 2023a; Chen *et al.*, 2023b], which is composed of a multimodal encoder block with a vision encoder like ViT [Dosovitskiy *et al.*, 2020] and a mT5 [Xue *et al.*, 2020; Raffel *et al.*, 2020] language encoder consuming image and text inputs, followed by an autoregressive decoder. The input

Model	ViT	Encoder-Decoder	#params
670M	B16 (92M)	mT5 base (583M)	675M
2B	H14 (653M)	mT5 Large (1.23B)	1.88B
5B	G14 (1.69B)	UL2-3B (2.93B)	4.62B

Table 1: Model variants and details of their parameter counts and split among vision and language models. The image encoders are based on ViT [Dosovitskiy *et al.*, 2020] and the text encoders are based on mT5 [Xue *et al.*, 2020] and UL2 models [Tay *et al.*, 2022].

image is transformed into a sequence of embeddings by the vision encoder and these embeddings are concatenated with the input text embeddings and fed into the mT5 language encoder. The output of this encoder is passed to the decoder to generate the text output. This generic formulation enables us to use the same model architecture to solve a variety of vision and multimodal tasks that can be recast as a text+image (input) to text (output) problem. Compared to the text input, the image embeddings constitute a significant portion of the input length to the multimodal encoder.

We further extend PaLI's encoder-decoder architecture to accept various image patching patterns. The original PaLI architecture only accepts a fixed grid pattern of patches for processing the input images. However, the data we encounter in screen-related domains spans a wide variety of resolutions and aspect ratios. To have a single model to work across all screen shapes, it is necessary to use a patching strategy which can work well with images of various shapes. To this end, we borrow a technique introduced in Pix2Struct [Lee *et al.*, 2023], which allows us to have image patches with arbitrary grid shapes based on the input image shape and a pre-defined maximum number of patches, as shown in Figure 1. This enables us to accommodate input images of various formats and aspect ratios without the need for padding or stretching the image to a fixed shape, making our model more polyvalent to handle both mobile (i.e. portrait) and desktop (i.e. landscape) image formats. In Section 5, we evaluate the impact of each of these modeling choices.

2.2 Model Configurations

We train models of 3 different sizes containing 670M, 2B and 5B parameters. For the 670M and 2B parameter models, we start from pre-trained unimodal checkpoints for the vision encoder and the encoder-decoder language models. For the 5B parameter model, we start from the multimodal pre-trained checkpoint from PaLI-3 [Chen *et al.*, 2023a], where the ViT is trained together with the UL2 [Tay *et al.*, 2022] based encoder-decoder language model. A breakdown of the parameter distribution among the vision and language models can be seen in Table 1.

Our patching strategy allows variable aspect ratios and input resolutions, as long as they fit within the allocated sequence length budget (2024 embeddings for the 670M model, 2916 embeddings for the 2B model, and 3364 embeddings for the 5B model). For square images, the corresponding maximum input resolution is 720×720 for the 670M model, 756×756 for the 2B model, and 812×812 for the 5B model.

2.3 Stages of Training

In this section, we cover the different stages of training.

Pre-Training. Starting from the checkpoints mentioned in Section 2.2, we do a first stage of training on large datasets generated from self-supervision and other models, using minimal human labeling (see Section 4.1 for a detailed description of the pre-training mixture). Contrary to the later fine-tuning stage, we train both the vision encoder and the language model. The motivation behind training the vision encoder is to incorporate the new patching strategy, and to allow the model to adapt from natural images to UI-related images. We evaluate the impact of training the vision encoder and of including LLM generated data on a variety of tasks in our ablation experiments in Section 5.

After some initial steps of pretraining, we perform additional steps with the ViT encoder frozen to further train the model while reducing the resource consumption.

Fine-Tuning. During fine-tuning, the model is trained on mixtures of tasks, most of which are labeled using human annotators. These tasks are described in details in Section 4.2. For QA-related tasks, we start by fine-tuning the model on a combination of QA-related tasks; then, additional training is performed on each individual tasks separately. For all other tasks, we fine-tune the model on each one individually.

3 Automatic Data Generation

The pretraining phase of our model's development is critically dependent on access to a vast and diverse dataset. Given the impracticality of manually annotating such an extensive dataset, our strategy focuses on automatic data generation. This approach leverages specialized smaller models, each adept at generating and labeling data both efficiently and with a high degree of accuracy.

In this section, we provide a detailed account of our data generation process, particularly highlighting how we gather and automatically annotate a diverse range of screenshots for pretraining our model. This automated approach is not only efficient and scalable compared to manual annotation but also ensures a level of data diversity and complexity.

3.1 Screen Annotation

Our initial step is to equip the model with a comprehensive understanding of textual elements, various screen components, and their overall structure and hierarchy. This foundational understanding is vital for the model's ability to interpret and interact accurately with a wide range of user interfaces.

An extensive collection of screenshots has been amassed from various devices, including desktops, mobile, and tablets, by crawling applications and web pages [Raffel *et al.*, 2020]. These screenshots are then annotated with detailed labels that describe the UI elements, their spatial relationships, and additional descriptive information.

The cornerstone of our annotation process is a layout annotator based on the DETR [Carion *et al.*, 2020] detection model. This object detector is apt at identifying and labeling a wide range of UI elements such as IMAGE, PICTOGRAM, BUTTON, TEXT, and others. This detector and the list of UI

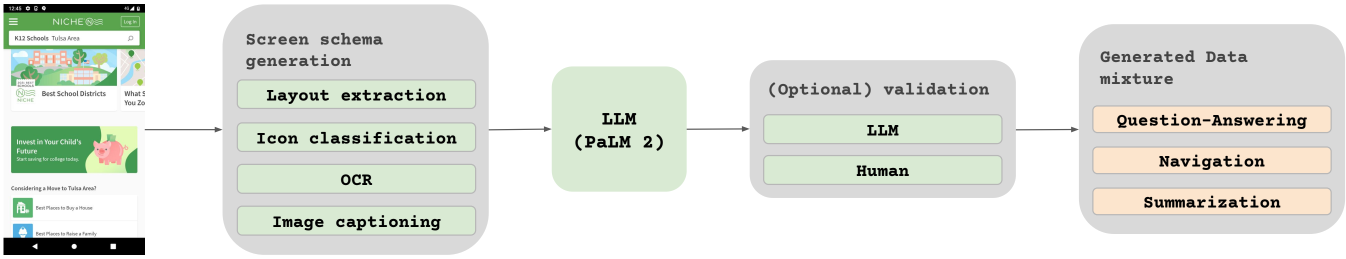


Figure 2: Task generation pipeline: 1) the screens are first annotated using various models; 2) we then use an LLMs to generate screen-related tasks at scale; 3) (optionally) we validate the data using another LLM or human raters.

elements is inspired by [Li *et al.*, 2022a]. However, the models in [Li *et al.*, 2022a] are classifiers and are provided a list of candidate bounding boxes to annotate, whereas in our case we predict the bounding boxes too.

Pictograms undergo further analysis using an icon classifier [Sunkara *et al.*, 2022] capable of distinguishing 77 different icon types. This detailed classification is essential for interpreting the subtle communication conveyed through icons. For icons that are not covered by the classifier, infographics and images, we use the PaLI image captioning model [Chen *et al.*, 2023b]. This model generates descriptive captions that provide contextual information, aiding in the comprehensive understanding of the screen's content.

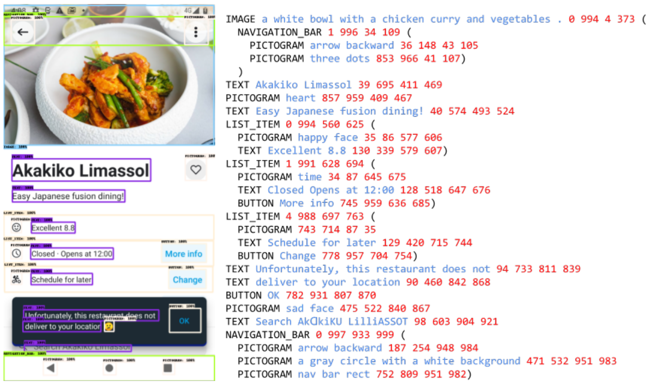
Additionally, an OCR engine extracts and annotates textual content on screen. This step is crucial for interpreting the textual information presented in various formats on interfaces. Finally, we combine the OCR text with the previous annotations to create a detailed and holistic description of each screen. The bounding box coordinates are systematically included, providing spatial context to the elements on the screen.

Figure 3 shows an example of the screen schema used in most of our pretraining tasks. Each schema contains:

1. The UI element names.
2. The OCR text (when applicable).
3. The element descriptions, e.g. captioning or icon names.
4. The bounding box coordinates, quantized and normalized between 0 and 999.

Parentheses are used to create a basic hierarchical structure between the elements, i.e. the children of a parent element are all put inside a parenthesis block. For ease of visualization, the bounding boxes from the screen schema have been overlaid on the original screenshot.

This schema plays a central role in our data generation for pretraining tasks, offering a detailed and multifaceted representation of screen content. The schema itself also serves as a pretraining task, where the model is tasked with generating a similar schema from a provided input image. This not only enhances the model's capacity to discern and interpret various UI components but also their relationships to one another. Additionally, the screen schema proves to be an invaluable natural language tool to interface with large language models (LLMs). By providing LLMs with a structured and detailed representation of screen content, we enable the creation of more intricate and contextually nuanced tasks.



```
IMAGE a white bowl with a chicken curry and vegetables . 0 994 4 373 (
  NAVIGATION_BAR 1 996 34 109 (
    PICTOGRAM arrow backward 36 148 43 105
    PICTOGRAM three dots 853 966 41 107)
  )
TEXT Akakiko Limassol 39 695 411 469
PICTOGRAM heart 857 959 409 467
TEXT Easy Japanese fusion dining! 40 574 493 524
LIST_ITEM 0 994 560 625 (
  PICTOGRAM happy face 35 86 577 606
  TEXT Excellent 8.8 130 339 579 607)
LIST_ITEM 1 991 628 694 (
  PICTOGRAM time 34 87 645 675
  TEXT Closed Opens at 12:00 128 518 647 676
  BUTTON More info 745 959 636 685)
LIST_ITEM 4 988 697 763 (
  PICTOGRAM 743 714 87 35
  TEXT Schedule for later 129 420 715 744
  BUTTON Change 778 957 704 754)
TEXT Unfortunately, this restaurant does not 94 733 811 839
TEXT deliver to your location 90 460 842 868
BUTTON OK 782 931 807 870
PICTOGRAM sad face 475 522 840 867
TEXT Search AkαkiKU LilliASSOT 98 603 904 921
NAVIGATION_BAR 0 997 933 999 (
  PICTOGRAM arrow backward 187 254 948 984
  PICTOGRAM a gray circle with a white background 471 532 951 983
  PICTOGRAM nav bar rect 752 809 951 982)
```

Figure 3: Example of our screen schema. See Appendix B for more.

3.2 LLMs to Generate Additional Tasks

To infuse greater diversity into our pretraining data, we leverage the capabilities of LLMs, in particular PaLM 2-S [Anil *et al.*, 2023b] to generate Question-Answer pairs in two stages. Initially, we generate the screen schema as previously described. Subsequently, we craft a prompt incorporating the screen schema and direct the LLM to generate synthetic data. This stage is empirical and necessitates a degree of prompt engineering. However, after several iterations, we typically identify a prompt that effectively generates the desired task. Example of such prompts are shown in Appendix C. To evaluate the quality of these generated responses, we conducted human validation on a subset of the data, ensuring that it meets a predetermined quality threshold.

This approach is described in Figure 2 and it enables us to create a variety of synthetic but realistic tasks that significantly enhance the depth and breadth of our pretraining dataset. By leveraging the natural language processing capabilities of LLMs, coupled with the structured screen schema, we can simulate a wide range of user interactions and scenarios. See Appendix D for generated examples.

4 Data Mixtures

We define two distinct sets of tasks for our model: an initial series of pretraining tasks and a subsequent set of fine-tuning tasks. The distinction primarily lies in two aspects:

1. **Source of the Groundtruth Data:** For the fine-tuning tasks, the labels are provided or verified by human raters. For the pretraining tasks, the labels are inferred using self supervised learning methods or generated using other models.
2. **Size of the Datasets:** Typically, the pretraining tasks encompass a significantly larger quantity of samples, and consequently, these tasks are used for training the model over a more extended series of steps.

4.1 Pretraining Mixture

Based on the methodology outlined in Section 3, we have selected the following tasks for pretraining our models. These tasks, each illustrated in Figure 4, are designed to cover a wide range of skills and scenarios, endowing our model with diverse real-world applications.

1. **Screen Annotation:** The model is tasked with detecting and identifying UI elements present on a screen. This includes performing OCR and image captioning to understand and interpret the textual and non-textual content. To enhance the model's contextual understanding, some text elements are intentionally masked, encouraging the model to infer information based on the surrounding context and layout.
2. **Screen Question-Answering (QA):** For this task, the model is asked to answer questions related to user interfaces and computer-generated images, such as infographics. After initial experiments, we identified certain gaps in performance on attributes like arithmetic, counting, understanding images with complex infographics. To enhance the model capabilities, we create data specifically addressing these gaps, e.g., QA involving counting, arithmetic operations, and complex data containing infographics. For these examples, we first crawl large scale webpage and infographic images, then perform prompt tuning to generate and validate relevant questions and their answers. For charts, the mix consists of 1) synthetic data [Liu *et al.*, 2023], 2) UniChart [Masry *et al.*, 2023], 3) DVQA [Kafle *et al.*, 2018], 4) TaTa [Gehrmann *et al.*, 2022], 5) Benetech [2].
3. **Screen Navigation:** This task involves interpreting navigation instructions (e.g., 'go back') and identifying the appropriate UI element to interact with. The expected output is the bounding box coordinates of the target element, bucketized between 0 and 999, demonstrating the model's ability to understand user intent and navigate through interfaces accurately.
4. **Screen Summarization:** The model is tasked to succinctly summarize the content of a screen in one or two sentences. This task assesses the model's capability to distill and caption the essence of the screen's content.

To ensure comprehensive training robust to aspect ratios, each task is made available across multiple formats (mobile and desktop) and includes several aspect ratios.

Task Name	#samples
Generated Screen Annotation	
Mobile webpages	262M
Mobile apps	54M
Mobile webpages (tall renders)	37M
Generated Screen Question-Answering	
Mobile webpages	9.8M
Mobile apps	2.0M
Mobile webpages (tall renders)	2.3M
Desktop webpages	16.4M
Infographics	6.3M
ChartQA/PlotQA	2.4M
Generated Screen Navigation	
Mobile webpages	2.6M
Mobile apps	5.9M
Mobile webpages (tall renders)	2.3M
Desktop webpages	5.1M
Generated Screen Summarization	
Mobile webpages	5.6M
Desktop webpages	7.6M
Other	
Tarzan [Xue *et al.*, 2020]	297K
VQA CC3M [Sharma *et al.*, 2018]	178K
WebLI Alt and OCR text [Kil *et al.*, 2023]	297K

Table 2: Detailed breakdown of our pretraining mixture.

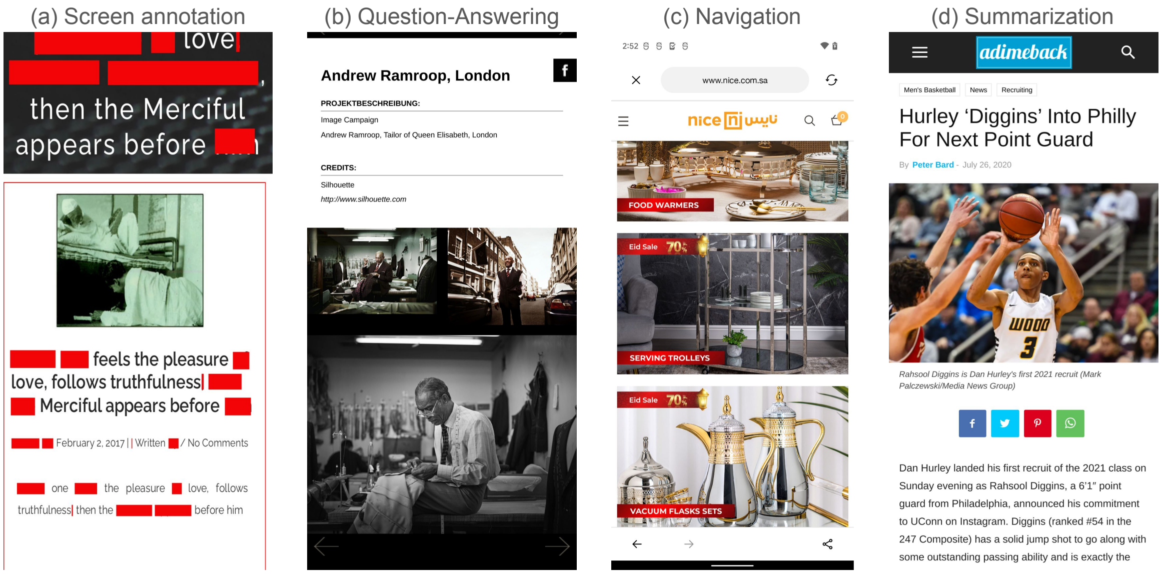
In addition to these screen-related tasks, our training regimen also incorporates a variety of other image and text data sources: Span corruption on C4 [Xue *et al.*, 2020], VQA CC3M [Sharma *et al.*, 2018], WebLI Alt and OCR text [Kil *et al.*, 2023; Chen *et al.*, 2022] and Chart-to-table translation [Liu *et al.*, 2023]. Such datasets have been instrumental in the development of PaLI models [Chen *et al.*, 2022; Chen *et al.*, 2023b], which serve as the foundational architecture for our model. Their inclusion ensures that our model not only excels in screen and infographics understanding but also maintains robust language and visual processing capabilities.

A summary of all our pretraining tasks is shown in Table 2. In the mixture, datasets are weighted proportionally to their size with a maximum allowed weight per task. Incorporating multimodal sources in our multi-task training, from language processing to visual comprehension and web content analysis, prepares our model to handle diverse scenarios effectively and enhances its overall versatility and performance.

4.2 Fine-Tuning Tasks and Benchmarks

We use a variety of tasks and benchmarks during fine-tuning to estimate the quality of our model. These benchmarks are summarized in Table 3 and include the main existing screen, infographics and document understanding benchmarks. We make the following changes to task formulations: (1) we cast RefExp [Wichers *et al.*, 2018] and Task Automation in MoTIF [Burns *et al.*, 2022] as object detection tasks, without using candidate bounding boxes and report accuracy at

[2] https://www.kaggle.com/competitions/benetech-making-graphs-accessible



love
then the Merciful appears before
feels the pleasure
love, follows truthfulness|
Merciful appears before
February 2, 2017 | | Written / No Comments
one the pleasure love, follows truthfulness| then the before him

Andrew Ramroop, London

PROJEKTBESCHREIBUNG:
Image Campaign
Andrew Ramroop, Tailor of Queen Elisabeth, London

CREDITS:
Silhouette
http://www.silhouette.com

2:52
www.nice.com.sa
FOOD WARMERS
Eid Sale 70%
SERVING TROLLEYS
Eid Sale 70%
VACUUM FLASKS SETS

adimeback
Men's Basketball News Recruiting

Hurley ‘Diggins’ Into Philly For Next Point Guard

By Peter Bard - July 26, 2020

Rahsool Diggins is Dan Hurley's first 2021 recruit (Mark Palczewski/Media News Group)

Dan Hurley landed his first recruit of the 2021 class on Sunday evening as Rahsool Diggins, a 6’1″ point guard from Philadelphia, announced his commitment to UConn on Instagram. Diggins (ranked #54 in the 247 Composite) has a solid jump shot to go along with some outstanding passing ability and is exactly the

Text input: Describe this screenshot.
Target: IMAGE pleasure or love follows truthfulness then the merciful appears before him 0 993 0 261 (TEXT pleasure of love, follows truthfulness, then the Merciful appears before him 3 991 0 248), IMAGE a ma...

Text input: What is the name of the tailor?
Target: Andrew Ramroop

Text input: Select the first item in the list.
Target: click 15 983 199 359

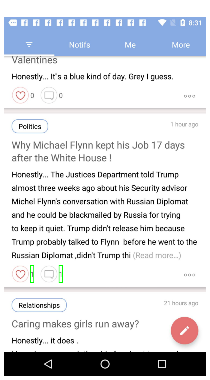
Text input: Summarize this screenshot.
Target: The screenshot shows a news article about UConn men's basketball recruiting. The article is about Dan Hurley's first recruit of the 2021 class, Rahsool Diggins, a 6'1" point guard from Philadelphia.

Figure 4: Sample of tasks that we are using in our pretraining mixture: (a) Screen annotation, with masking; (b) Question-Answering; (c) Navigation; (d) Summarization. The last three have been generated using our screen annotation model, coupled with PaLM-2-S.

IoU=0.1[3] considering only one box predicted; (2) for MoTIF, we report the number for the app-unseen split of the test set in Table 4, and other split results in Appendix E.

We supplement the tasks mentioned above with three new benchmarks that we release:

- **Screen Annotation (SA):**[4] To evaluate our model's layout annotation and spatial understanding capabilities, we create a dedicated benchmark consisting of 4.2K screenshots from the Rico dataset [Deka *et al.*, 2017]. Each UI element has been annotated by human raters, and the annotations comprise a bounding box and a UI class from the list described in 3.1. We evaluate the model's predictions using object detection metrics, including F1 score, precision and recall values computed at IoU=0.1.
- **ScreenQA Short (SQA Short):**[5] ScreenQA [Hsiao *et al.*, 2022], a benchmark for screen understanding, contains UI elements and full-sentence answers as ground truth. To align the output format with other question answering tasks, we generate a new ground truth, a list of alternative short answers, for each of the questions. We use the maximum F1 score across all the candidate answers as the metric. See Figure 5 and Appendix F for more details.



Valentines
Honestly... It"s a blue kind of day. Grey I guess.
0 0

Politics 1 hour ago
Why Michael Flynn kept his Job 17 days after the White House !
Honestly... The Justices Department told Trump almost three weeks ago about his Security advisor Michel Flynn's conversation with Russian Diplomat and he could be blackmailed by Russia for trying to keep it quiet. Trump didn't release him because Trump probably talked to Flynn before he went to the Russian Diplomat ,didn't Trump thi (Read more...)
1 1

Relationships 21 hours ago
Caring makes girls run away?
Honestly... it does .

Question: How many links and comments are there of the post "Why Michael Flynn kept his Job 17 days after the White House!" ?

Full sentence answers:

- There is 1 like and 1 comment on the post "Why Michael Flynn kept his job 17 days after the White House!".
- There is 1 like and 1 comment on the "Why Michael Flynn kept his Job 17 days after the White House!" post.
- There is 1 like and 1 comment.

List of short answers:

- one and one
- 1 and 1
- one, one
- 1, 1
- 1 like, 1 comment
- 1 like and 1 comment

Figure 5: Examples of questions and answers from the ScreenQA dataset, together with their LLM-generated short answers.

- **Complex ScreenQA (Cplx SQA):**[6] To complement SQA Short, we introduce Complex ScreenQA, which includes more difficult questions (counting, arithmetic, comparison, and non-answerable questions) and contains screens with various aspect ratios. See Figures 6 and 7 for examples and Appendix G for more details.

We also provide a few additional details on how we handle Multipage DocVQA and ChartQA.

[3] Intersection over union at threshold 0.1

[4] https://github.com/google-research-datasets/screen_annotation

[5] https://github.com/google-research-datasets/screen_qa?tab=readme-ov-file#screenqa-short

[6] https://github.com/google-research-datasets/screen_qa?tab=readme-ov-file#complexqa

Task Name/Benchmark	Metric
Screen Analysis	
Screen Annotation [Ours, Sec. 4.2]	F1@IoU=0.1
Widget Captioning [Li *et al.*, 2020]	CIDEr
Screen Question-Answering	
ScreenQA Short [Ours, Sec. 4.2]	SQuAD F1
Complex ScreenQA [Ours, Sec. 4.2]	SQuAD F1
WebSRC [Chen *et al.*, 2021b]	SQuAD F1
Screen Navigation	
RefExp [Bai *et al.*, 2021]	Acc@IoU=0.1
MoTIF-Automation [Burns *et al.*, 2022]	Acc@IoU=0.1
Screen Summarization	
Screen2Words [Wang *et al.*, 2021]	CIDEr
Infographics/Doc Visual QAs	
ChartQA [Masry *et al.*, 2022]	Relaxed Acc.
DocVQA [Mathew *et al.*, 2021]	ANLS
Multipage DocVQA [Tito *et al.*, 2023]	ANLS
InfographicVQA [Mathew *et al.*, 2022]	ANLS
OCR-VQA-200K [Mishra *et al.*, 2019]	Exact Match

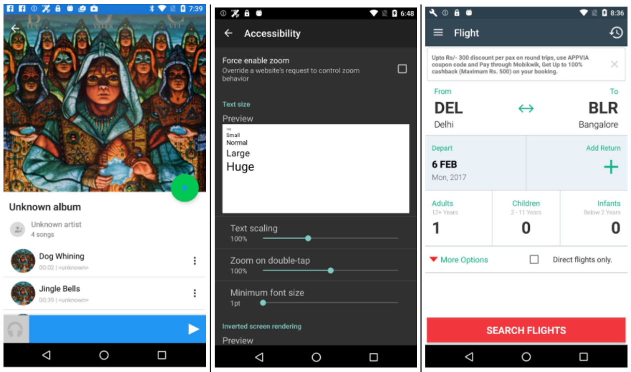
Table 3: Detailed breakdown of our fine-tuning mixture and their associated metrics. We assume readers are familiar with these metrics, but include descriptions and citations in Appendix A for reference.

Multipage DocVQA. The standard fine-tuning task for Multipage DocVQA [Tito *et al.*, 2023] can be transformed into a single-page DocVQA task by pairing the same question with each page of the document and choosing the answer with the highest score among all pages. In this formulation, we modify the training set by splitting a question, answer and multipage document into a positive pair (with the actual answer for the page containing the answer) and multiple negative pairs (with "no answer" for pages which do not contain the answer). The negative pairs are subsampled to avoid overfitting on not predicting an answer and the original DocVQA task [Mathew *et al.*, 2021] is added to the fine-tuning mixture.

ChartQA. Concurrent work in [Carbune *et al.*, 2024] showed that the original fine-tuning dataset [Masry *et al.*, 2022] is insufficiently rich for learning solving complex reasoning tasks. There, they overcome this limitation through synthetic examples and rationales, paired with training loss changes. Here, we leverage the synthetic examples, but without modifying the training loss or incorporating rationales. We therefore maintain parity how we fine-tune for the rest of the tasks. We report similar performance with or without OCR, hinting that the scale of the dataset contributes more than the input features. Our results otherwise further strengthen the contribution of the pre-training and architecture changes with pix2struct to better leverage the same synthetic examples and not needing to rely on rationales.

5 Experiments and Results

In this section, we present the setup we used to conduct our experiments and analyze our findings. First, we compare the best performing ScreenAI model to the SoTA on a variety of Screen and Infographics related tasks. Next, we report the impact of model size on overall performance. Finally, we report results on ablation studies to validate the design choices made for the models.



Unknown album
Unknown artist
4 songs
Dog Whining
00:02 | <unknown>
Jingle Bells
00:39 | <unknown>

Accessibility
Force enable zoom
Override a website's request to control zoom behavior □
Text size
Preview
Small
Normal
Large
Huge
Text scaling 100%
Zoom on double-tap 100%
Minimum font size 1pt
Inverted screen rendering
Preview

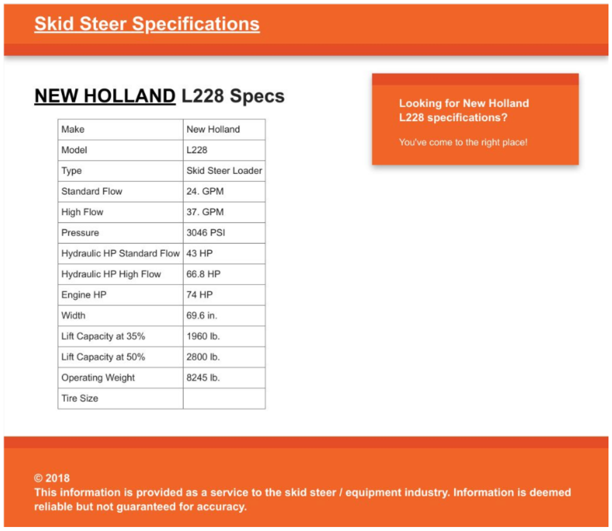
Flight
Upto Rs/- 300 discount per pax on round trips, use APPVIA coupon code and Pay through Mobikwik, Get Up to 100% cashback (Maximum Rs. 500) on your booking.
From DEL Delhi ↔ To BLR Bangalore
Depart 6 FEB Mon, 2017
Add Return +
Adults 12+ Years 1
Children 2 - 11 Years 0
Infants Below 2 Years 0
More Options
□ Direct flights only.
SEARCH FLIGHTS

Question: How many songs have a duration of less than 30 seconds? **Answer:** 1

Question: How many text size options are there? **Answer:** 5

Question: How many days are between the departure and return dates? **Answer:** There is no answer on the screen.

Figure 6: Examples of mobile screen in Complex QA dataset.



Skid Steer Specifications

NEW HOLLAND L228 Specs

Make	New Holland
Model	L228
Type	Skid Steer Loader
Standard Flow	24. GPM
High Flow	37. GPM
Pressure	3046 PSI
Hydraulic HP Standard Flow	43 HP
Hydraulic HP High Flow	66.8 HP
Engine HP	74 HP
Width	69.6 in.
Lift Capacity at 35%	1960 lb.
Lift Capacity at 50%	2800 lb.
Operating Weight	8245 lb.
Tire Size	

Looking for New Holland L228 specifications?

You've come to the right place!

© 2018
This information is provided as a service to the skid steer / equipment industry. Information is deemed reliable but not guaranteed for accuracy.

Question: What is the lift capacity at 35%? **Answer:** 1960 lb.

Figure 7: An example of desktop screen in Complex QA dataset.

5.1 Experiments Setup

In the fine-tuning phase, we hold the ViT encoder frozen and fine-tune the language model only. We use 512 as our batch size for fine-tuning. Our text input sequence length is 128 and output sequence length varies depending on individual tasks. When fine-tuning with OCR as additional input, we increase the input sequence length accordingly. We generally find that the model converges within 30k steps. Unless specified otherwise, all experiments are run on the 5B model.

5.2 Results

Table 4 shows the performance of our models and compares them with state-of-the-art (SoTA) results on a variety of screen- and infographics-related tasks. We also include

	SA	Ref Exp	SQA Short	Cplx SQA	MoTIF	Screen2 Words	Widget Capt.	Chart QA	Doc VQA	MPDoc VQA	Info VQA	OCR VQA	Web SRC
SoTA	-	-	-	-	67.6^a	**130.7**^{b}	**159.8**^{b}	**80.8**^{h}	**90.9**^{h}	61.8^d	**80.3**^{h}	**77.8**^{b}	85.0^f
Without OCR													
SoTA≤5B	-	-	-	-	67.6^a	130.7^b	159.8^b	$\underline{77.3}^i$	$\underline{87.8}^c$	-	57.8^b	$\underline{76.7}^b$	77.8^g
ScreenAI	**86.2**	**86.3**	**94.6**	**42.4**	**87.4**	120.8	156.4	76.6	87.5	**72.9**	$\underline{61.4}$	75.0	**87.2**
With OCR													
SoTA≤5B	-	-	-	-	-	-	-	70.4^c	89.3^c	61.8^d	62.4^b	$\underline{77.8}^b$	85.0^f
ScreenAI	-	-	**94.8**	**43.5**	-	123.7	-	$\underline{76.7}$	$\underline{89.9}$	**77.1**	$\underline{65.9}$	76.2	-

Table 4: Comparison of ScreenAI with various SoTA models: (a) MoTIF [Burns *et al.*, 2022], (b) PaLI-3 [Chen *et al.*, 2023b], (c) SmoLA PaLI-X [Wu *et al.*, 2023a], (d) Hi-VT5 [Tito *et al.*, 2023], (e) TILT [Powalski *et al.*, 2021], (f) DocPrompt [Wu *et al.*, 2023b], (g) DUBLIN [Aggarwal *et al.*, 2023], (h) Gemini [Anil *et al.*, 2023a], (i) ChartPaLI-5B [Carbune *et al.*, 2024]. Bold font highlights SoTA score, and underscore represents best-in-class score. See Table 3 for details about the tasks and their associated metrics.

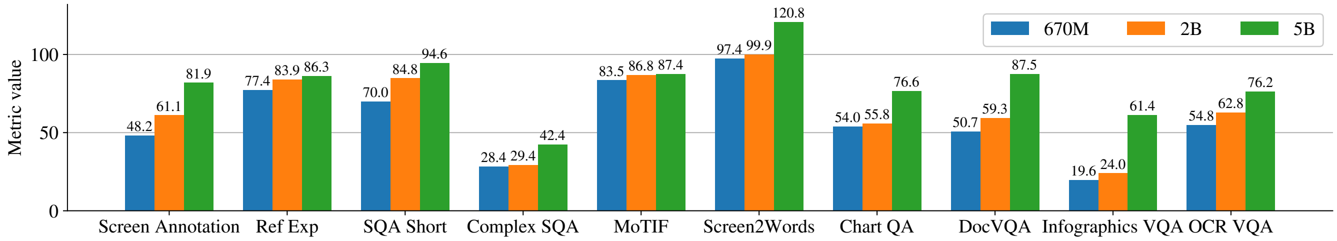


Figure 8: Performance of different model sizes on fine-tuning tasks. The metrics improve consistently as the model size increases.

the best results for models of similar size (SoTA<5B). We report new SoTA results on MoTIF, MPDocVQA, and WebSRC; and new best-in-class results in ChartQA, DocVQA and InfographicVQA (InfoVQA). We report same or competitive performance on Screen2Words, Widget Captioning, and OCR-VQA. We also report our results on the benchmarks introduced in Section 4.2 (Screen Annotations, Referring Expressions, ScreenQA Short and Complex ScreenQA).

Adding OCR as Additional Input. We analyze the impact of adding OCR[7] to the model input by conducting experiments with and without OCR. This is inspired by fine-tuning experiments in PaLI [Chen *et al.*, 2023b], where across all screen- and document-related tasks, passing OCR texts as additional input improves task performance. In Table 4 we present our single task fine-tuning results using OCR data. For QA tasks, OCR input provides a boost in performance (e.g. up to 4.5% on Complex ScreenQA, MPDocVQA and InfoVQA). However, using OCR imposes a slightly larger input length and hence results in slower overall training. It also requires having OCR results available at inference time.

Model Size. We conducted single task experiments with the following model sizes: 670M, 2B and 5B. We use benchmarks for screen tasks as well as other public tasks. In Figure 8, we observe that across all tasks, increasing the model size improves performances and the improvements have not saturated at the largest size. We observe that for tasks that require more complex visual-text and arithmetic reasoning e.g. InfoVQA, ChartQA, and Complex ScreenQA, the improvement between 2B and 5B models is significantly larger than between 670M and 2B models.

5.3 Ablation Studies

In this section, we perform ablation studies evaluating (1) the impact of pix2struct patching and (2) using LLM generated data for pre-training. All ablation studies are performed on the 670M parameter variant.

Impact of Pix2struct Patching. For this study, we compare a 670M model using pix2struct patching with another using fixed-grid patching. After pre-training, both models are fine-tuned on all tasks in Table 3. We split each dataset into subsets based on the image aspect ratio and compute the respective metric on these subsets. To compare fixed-grid patching to a variable pix2struct patching, we compute an *aggregate score*, by first dividing the score of each task subset using fixed-grid patching by the score of the model using pix2struct on the entire task, and finally compute the geometric mean across all tasks. Figure 9 shows that for images with aspect ratio > 1.0 (landscape mode images), the pix2struct patching strategy is significantly better than the fixed grid patching. For portrait mode images, the trend is reversed, but fixed grid patching is only marginally better. Given that we want the ScreenAI model to be used across images of different aspect ratios, we choose to use pix2struct patching.

Impact of LLM Generated Data. For this experiment, we compare a 670M ScreenAI model pre-trained using all the

[7] We use a proprietary OCR system similar to GCP Vision API to produce additional OCR input for each image.

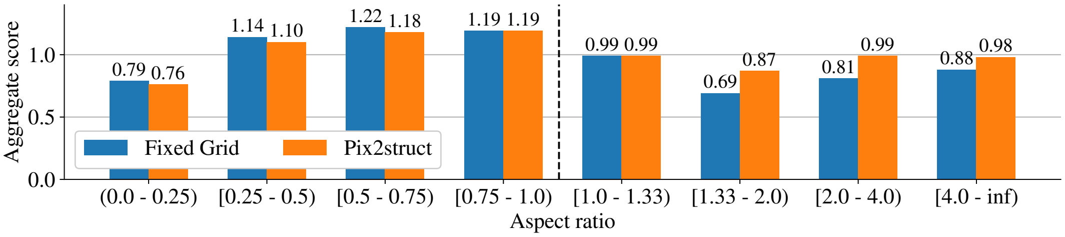


Figure 9: Ablation study for Pix2Struct vs. fixed-grid patching; the numbers represent the aggregated scores across all fine-tuned tasks. For aspect ratio > 1.0, using Pix2Struct patching significantly outperforms a fixed grid patching, whereas for aspect ratio < 1.0, a fixed grid patching outperforms Pix2Struct by a smaller margin.

datasets mentioned in Section 4.1 against a model pre-trained on a mixture excluding any LLM generated pre-training data. After pre-training, both models are fine-tuned on all tasks mentioned in Table 3 and an aggregate score is computed. We observe that adding LLM generated data to the mixture improves the aggregate score by 4.6 percentage points.

6 Conclusions

In this work, we introduce the ScreenAI model along with a new unified schema for representing complex data and visual information, compatible with infographics, document images, and various UIs. This unified representation enables the design of a mixture of self-supervised learning tasks, leveraging data from all these domains. We show that training on this mixture results in a positive transfer to screen-related tasks as well as infographics and document-related tasks. We also illustrate the impact of data generation using LLMs and justify our model design choices with ablation studies. We apply these techniques to train a model that performs competitively and achieves SoTA on a number of public benchmarks. While our model is best-in-class, we note that, on some tasks, further research is needed to bridge the gap with models like GPT-4 and Gemini, which are orders of magnitude larger. To encourage further research, we release a dataset with this unified representation, as well as two other datasets to enable more comprehensive benchmarking of models on screen-related tasks.

A Appendix

See extended edition at https://arxiv.org/abs/2402.04615 for all appendices.

Acknowledgements

We would like to thank team alumni Yo Hsiao and Zixian Ma for their contribution to the project, Fangyu Liu, Xi Chen, Efi Kokiopoulou, Jesse Berent, Gabriel Barcik, Lukas Zilka, Oriana Riva, Gang Li, Yang Li, Radu Soricut and Tania Bedrax-Weiss for their insightful feedbacks and fruitfull discussions, Rahul Aralikatte, Hao Cheng and Daniel Kim for their whole-hearted and tireless support in data preparation, and Jay Yagnik, Blaise Aguera y Arcas, Ewa Dominowska, David Petrou, and Matt Sharifi for their vision and support in leadership.

Contribution Statement

First Authors with Equal Contributions: Gilles Baechler, Srinivas Sunkara, Maria Wang, Jindong Chen.

Project Leads: Jindong Chen, Abhanshu Sharma

References

[Aggarwal *et al.*, 2023] Kriti Aggarwal, Aditi Khandelwal, Kumar Tanmay, Owais Mohammed Khan, Qiang Liu, Monojit Choudhury, Subhojit Som, Vishrav Chaudhary, and Saurabh Tiwary. DUBLIN–document understanding by language-image network. *arXiv preprint arXiv:2305.14218*, 2023.

[Aghajanyan *et al.*, 2021] Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. HTLM: Hyper-text pre-training and prompting of language models, 2021.

[Alayrac *et al.*, 2022] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

[Anil *et al.*, 2023a] Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[Anil *et al.*, 2023b] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. PaLM 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[Bai *et al.*, 2021] Chongyang Bai, Xiaoxue Zang, Ying Xu, Srinivas Sunkara, Abhinav Rastogi, Jindong Chen, and Blaise Aguera y Arcas. UIBert: Learning generic multimodal representations for UI understanding, 2021.

[Burns *et al.*, 2022] Andrea Burns, Deniz Arsan, Sanjna Agrawal, Ranjitha Kumar, Kate Saenko, and Bryan A. Plummer. A dataset for interactive vision language navigation with unknown command feasibility. In *European Conference on Computer Vision (ECCV)*, 2022.

[Carbune *et al.*, 2024] Victor Carbune, Hassan Mansoor, Fangyu Liu, Rahul Aralikatte, Gilles Baechler, Jindong Chen, and Abhanshu Sharma. Chart-based reasoning: Transferring capabilities from llms to vlms, 2024.

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey

Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[Chen *et al.*, 2021a] Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021.

[Chen *et al.*, 2021b] Xingyu Chen, Zihan Zhao, Lu Chen, Danyang Zhang, Jiabao Ji, Ao Luo, Yuxuan Xiong, and Kai Yu. WebSRC: A dataset for web-based structural reading comprehension, 2021.

[Chen *et al.*, 2022] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. PaLi: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

[Chen *et al.*, 2023a] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. PaLI-X: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023.

[Chen *et al.*, 2023b] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. PaLI-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*, 2023.

[Deka *et al.*, 2017] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*, pages 845–854, 2017.

[Deng *et al.*, 2023] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *arXiv preprint arXiv:2306.06070*, 2023.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[Gehrmann *et al.*, 2022] Sebastian Gehrmann, Sebastian Ruder, Vitaly Nikolaev, Jan A. Botha, Michael Chavinda, Ankur Parikh, and Clara Rivera. Tata: A multilingual table-to-text dataset for african languages, 2022.

[Gur *et al.*, 2022] Izzeddin Gur, Ofir Nachum, Yingjie Miao, Mustafa Safdari, Austin Huang, Aakanksha Chowdhery, Sharan Narang, Noah Fiedel, and Aleksandra Faust. Understanding HTML with large language models. *arXiv preprint arXiv:2210.03945*, 2022.

[He *et al.*, 2021] Zecheng He, Srinivas Sunkara, Xiaoxue Zang, Ying Xu, Lijuan Liu, Nevan Wichers, Gabriel Schubiner, Ruby Lee, Jindong Chen, and Blaise Agüera y Arcas. ActionBert: Leveraging user actions for semantic understanding of user interfaces, 2021.

[Hsiao *et al.*, 2022] Yu-Chung Hsiao, Fedir Zubach, Maria Wang, et al. ScreenQA: Large-scale question-answer pairs over mobile app screenshots. *arXiv preprint arXiv:2209.08199*, 2022.

[Huang *et al.*, 2022] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. LayoutLMv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4083–4091, 2022.

[Kafle *et al.*, 2018] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering, 2018.

[Kil *et al.*, 2023] Jihyung Kil, Soravit Changpinyo, Xi Chen, Hexiang Hu, Sebastian Goodman, Wei-Lun Chao, and Radu Soricut. PreSTU: Pre-training for scene-text understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15270–15280, 2023.

[Kim *et al.*, 2021] Geewook Kim, Teakgyu Hong, Moonbin Yim, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Donut: Document understanding transformer without OCR. *arXiv preprint arXiv:2111.15664*, 7:15, 2021.

[Kuo *et al.*, 2023] Weicheng Kuo, AJ Piergiovanni, Dahun Kim, Xiyang Luo, Ben Caine, Wei Li, Abhijit Ogale, Luowei Zhou, Andrew Dai, Zhifeng Chen, et al. MaMMUT: A simple architecture for joint learning for multimodal tasks. *arXiv preprint arXiv:2303.16839*, 2023.

[Lee *et al.*, 2023] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR, 2023.

[Li and Li, 2022] Gang Li and Yang Li. Spotlight: Mobile UI understanding using vision-language models with a focus. *arXiv preprint arXiv:2209.14927*, 2022.

[Li *et al.*, 2020] Yang Li, Gang Li, Luheng He, Jingjie Zheng, Hong Li, and Zhiwei Guan. Widget captioning: Generating natural language description for mobile user interface elements, 2020.

[Li *et al.*, 2021] Yang Li, Gang Li, Xin Zhou, Mostafa Dehghani, and Alexey Gritsenko. VUT: Versatile ui transformer for multi-modal multi-task user interface modeling. *arXiv preprint arXiv:2112.05692*, 2021.

[Li *et al.*, 2022a] Gang Li, Gilles Baechler, Manuel Tragut, and Yang Li. Learning to denoise raw mobile UI layouts for improving datasets at scale. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2022.

[Li *et al.*, 2022b] Tao Li, Gang Li, Jingjie Zheng, Purple Wang, and Yang Li. MUG: Interactive multimodal grounding on user interfaces, 2022.

[Liu *et al.*, 2022] Fangyu Liu, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Yasemin Altun, Nigel Collier, and Julian Martin Eisenschlos. MatCha: Enhancing visual language pretraining with math reasoning and chart derendering. *arXiv preprint arXiv:2212.09662*, 2022.

[Liu *et al.*, 2023] Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. DePlot: One-shot visual language reasoning by plot-to-table translation, 2023.

[Masry *et al.*, 2022] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

[Masry *et al.*, 2023] Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. Unichart: A universal vision-language pretrained model for chart comprehension and reasoning, 2023.

[Mathew *et al.*, 2021] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. DocVQA: A dataset for VQA on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209, 2021.

[Mathew *et al.*, 2022] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. InfographicVQA. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022.

[Mishra *et al.*, 2019] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. OCR-VQA: Visual question answering by reading text in images. In *ICDAR*, 2019.

[Nakano *et al.*, 2021] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

[Powalski *et al.*, 2021] Rafał Powalski, Łukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michał Pietruszka, and Gabriela Pałka. Going full-tilt boogie on document understanding with text-image-layout transformer, 2021.

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[Rawles *et al.*, 2023] Christopher Rawles, Alice Li, Daniel Rodriguez, Oriana Riva, and Timothy Lillicrap. Android in the wild: A large-scale dataset for android device control. *arXiv preprint arXiv:2307.10088*, 2023.

[Sharma *et al.*, 2018] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[Sunkara *et al.*, 2022] Srinivas Sunkara, Maria Wang, Lijuan Liu, Gilles Baechler, Yu-Chung Hsiao, Abhanshu Sharma, James Stout, et al. Towards better semantic understanding of mobile interfaces. *arXiv preprint arXiv:2210.02663*, 2022.

[Tang *et al.*, 2023] Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. Unifying vision, text, and layout for universal document processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19254–19264, 2023.

[Tay *et al.*, 2022] Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, et al. UL2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*, 2022.

[Tito *et al.*, 2023] Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multipage DocVQA. *Pattern Recognition*, 144:109834, 2023.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wang *et al.*, 2021] Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. Screen2words: Automatic mobile ui summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 498–510, 2021.

[Wang *et al.*, 2022] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *International Conference on Machine Learning*, pages 23318–23340. PMLR, 2022.

[Wang *et al.*, 2023] Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. DocLLM: A layout-aware generative language model for multimodal document understanding. *arXiv preprint arXiv:2401.00908*, 2023.

[Wichers *et al.*, 2018] Nevan Wichers, Dilek Hakkani-Tür, and Jindong Chen. Resolving referring expressions in images with labeled elements. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 800–806. IEEE, 2018.

[Wu *et al.*, 2021] Jason Wu, Xiaoyi Zhang, Jeff Nichols, and Jeffrey P Bigham. Screen parsing: Towards reverse engineering of ui models from screenshots. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 470–483, 2021.

[Wu *et al.*, 2023a] Jialin Wu, Xia Hu, Yaqing Wang, Bo Pang, and Radu Soricut. Omni-SMoLA: Boosting generalist multimodal models with soft mixture of low-rank experts, 2023.

[Wu *et al.*, 2023b] Sijin Wu, Dan Zhang, Teng Hu, and Shikun Feng. DocPrompt: Large-scale continue pretrain for zero-shot and few-shot document question answering, 2023.

[Xue *et al.*, 2020] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*, 2020.

[Yang *et al.*, 2022] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. UniTAB: Unifying text and box outputs for grounded vision-language modeling. In *European Conference on Computer Vision*, pages 521–539. Springer, 2022.

[Zang *et al.*, 2021] Xiaoxue Zang, Ying Xu, and Jindong Chen. Multimodal icon annotation for mobile applications. In *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction*, pages 1–11, 2021.

[Zhang *et al.*, 2021] Xiaoyi Zhang, Lilian de Greef, Amanda Swearngin, Samuel White, Kyle Murray, Lisa Yu, Qi Shan, Jeffrey Nichols, Jason Wu, Chris Fleizach, et al. Screen recognition: Creating accessibility metadata for mobile applications from pixels. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15, 2021.