AfriST-VQA: Benchmarking MLLMs for Scene-Text Visual Question Answering for African Languages

Henry Gagnier
Pittsford Sutherland High School
Pittsford, NY, USA
henrygagnier9@gmail.com

Abstract

Scene-text visual question answering (ST-VQA) benchmarks have driven significant progress in multimodal large language model (MLLM) evaluation, yet existing benchmarks focus exclusively on high-resource languages, leaving African languages completely unevaluated. We introduce AfriST-VQA, the first ST-VQA benchmark for African languages, spanning eight languages across two scripts, including Swahili, Yoruba, Hausa, Igbo, Kinyarwanda, Somali, Amharic, and Tigrinya. We propose a six-phase framework for synthetic ST-VQA benchmark construction that addresses the low-resource image data, translation, and realistic scene-text rendering, and we implement a version of this framework to produce 1,024 images and 32,768 multiple-choice QA pairs across four question types: reading, reasoning, spatial, and verification. Evaluating four MLLMs on this benchmark, we find that models exhibit highly uneven performance with a 23.5-point gap between the strongest and weakest models and a 14.9-point gap between the highest and lowest scoring languages, which are Swahili and Tigrinya, respectively. Script familiarity and per-language resource availability are likely driving the performance disparity, and spatial reasoning is a challenge across all models and languages. We provide the first systematic evidence of these gaps and a reusable framework for future low-resource ST-VQA benchmark construction.

1. Introduction

The ability to read and understand text embedded in natural scenes, such as menus, street signs, storefronts, and documents, is a foundational capability for multimodal large language models (MLLMs) [9, 20]. Text visual question answering (VQA), such as OCR-VQA [24] and MTVQA [34], and scene-text VQA benchmarks, such as ST-VQA [9] and TextVQA [31], have led to significant progress in the area, requiring models to reason over image text and contents. Scene-text benchmarks are high-resource language-focused, leaving the abilities of MLLMs to reason over low-resource languages and scripts in context unevaluated.

This gap is consequential for low-resource languages and African languages in particular. Africa has over 2,000 languages spoken by more than 1.4 billion people, yet these languages are extremely under-resourced and underrepresented in NLP research [1, 3, 13]. Scene-text understanding has implications including functional MLLMs and access to information in signage, commercial goods, public health materials, and digital documents, which makes the exclusion of African languages in OCR-VQA evaluation a knowledge gap in research and an equity concern [12, 17]. Without benchmarks for low-resource ST-VQA, we cannot assess whether current MLLMs generalize to African scripts and orthographies or identify when they fail.

Constructing a synthetic benchmark for African language VQA presents substantial challenges. African languages are often low-resource, usually lacking annotated scene-text datasets, and the cost of image collection and labeling at scale is limiting [3, 26]. Prior work to evaluate MLLMs on low-resource language OCR has created simplistic synthetic benchmarks with text rendered on plain or minimally textured backgrounds, which fail to capture the complexity of authentic scene-text environments and therefore may overestimate model performance [14, 23].

In this paper, we introduce AfriST-VQA, a synthetic benchmark designed to close this gap. We develop the first ST-VQA benchmark for African languages in two scripts—Swahili, Yoruba, Hausa, Igbo, Somali, Kinyarwanda, Amharic, and Tigrinya—propose a pipeline for synthetic low-resource language ST-VQA image generation, and evaluate four MLLMs on AfriST-VQA, revealing substantial performance disparities across languages and scripts.

2. Related Work

Scene-Text VQA Benchmarks. Visual question answering in scene text is an important task to evaluate multi-

modal reasoning in images containing written language. ST-VQA [9] and TextVQA [31] introduce benchmarks combining natural scene-text images with questions requiring models to read and reason over image text. Recently, language-specific benchmarks such as ViTextVQA [37], ViSignVQA [27], KRETA [16], and MTVQA [34] have been developed from authentic scripts. While expanding scene-text VQA beyond English, these benchmarks do not encompass African languages or scripts.

Synthetic OCR Datasets. To address the scarcity of image data for low-resource languages, recent work has explored synthetic dataset construction, typically rendering translated text onto simple backgrounds [11, 14, 22, 29]. While useful for evaluating basic OCR capabilities, these benchmarks do not capture the challenges of real-world scene-text understanding and may inflate MLLM performance.

African NLP Benchmarks. NLP resources for African languages have increased significantly, with benchmarks for named entity recognition [4], sentiment analysis [25], and language identification [1, 2, 30]. Initiatives such as Masakhane [28] have helped establish evaluation benchmarks for multiple African languages. Despite these advances, multimodal benchmarks for African languages remain extremely limited. Our work introduces the first benchmark for scene-text VQA across multiple African languages.

3. A Framework for Synthetic Scene-Text VQA Benchmark Construction

We propose a framework for synthetic scene-text VQA benchmark construction decomposed into six phases (Table 1). AfriST-VQA is one implementation; the framework is intended to guide future benchmark construction for other low-resource language families.

Phase 1: Source Imagery. Source images can be drawn from existing annotated scene-text corpora such as COCO-Text [38] or TextVQA [31], collected in the field, generated via diffusion models, or replaced with plain synthetic backgrounds. Existing corpora are the most practical starting point but reflect Western visual environments. Plain synthetic backgrounds eliminate the visual context required for scene-text understanding, potentially overestimating MLLM performance.

Phase 2: Text Region Localization. Text replacement requires precisely localized text. Inherited bounding boxes are available from existing benchmarks. Off-the-shelf detectors such as CRAFT [7] and DBNet [19] produce tighter detections but may miss stylized text; oriented bounding box refinement via edge-based orientation estimation offers a practical middle ground. Manual annotation is most accurate but expensive at scale.

Phase 3: Translation. Translating source text and QA pairs can be extremely error-prone. MT systems and LLMs are scalable but may fail for low-resource languages. Cross-modal consistency must also be considered: image text strings and QA answer strings should be translated together to ensure answers match rendered images.

Phase 4: Image Synthesis. The synthesis approach determines benchmark realism. Inpaint-then-render pipelines (e.g., LaMa [33]) erase source text and re-render translated text with matched visual style. Diffusion-based text editors (e.g., AnyText [36]) offer more photorealistic edits for Latin scripts but have limited support for complex scripts. Plain compositing is trivial to implement but produces obvious artifacts.

Phase 5: QA Generation and Task Design. LLM-based QA generation is scalable but risks producing questions answerable from world knowledge without reading the image. Constraining distractors to other text strings in the same image substantially reduces this confound. A multi-type taxonomy covering reading, reasoning, spatial, and verification subtasks is preferable to a single type, as each has a distinct failure mode.

Phase 6: Quality Filtering. Synthetic pipelines accumulate errors at each stage. Image-level filters (on resolution, text region count, and word frequency) select authentic scene-text images. Translation filters (script compliance, diacritic retention, untranslated string detection) catch MT/LLM failures. QA consistency checks verify that answer strings appear in the translated text list for each image.

We advise that synthetic ST-VQA benchmark results should not be compared with traditional ST-VQA benchmarks without significant improvements. Errors in machine translation, rendering unrealism, and cultural inaccuracies mean synthetic benchmarks are not comparable with authentic images, and their limitations must be considered.

4. Benchmark Construction

We constructed AfriST-VQA using existing scene-text datasets, LLM-based QA generation and translation, and image synthesis. After construction and filtering, we produced 1,024 images and 32,768 QA pairs.

4.1. Source Images

We build AfriST-VQA from English-language scene-text datasets that supply both natural images and per-region text annotations: COCO-Text [38] and TextVQA [31]. Raw

Phase	Methods	Key Tradeoff	AfriST-VQA Choice
1. Source Imagery	Existing corpora; real-world collection; diffusion backgrounds; plain synthetic	Ecological validity vs. cost and domain mismatch	COCO-Text + TextVQA
2. Text Localization	Inherited boxes; off-the-shelf detectors; OBB refinement; manual annotation	Annotation precision vs. cost	OBB refinement of inherited boxes
3. Translation	MT systems; LLM translation; human translation	Quality and consistency vs. cost and scalability	LLM (Gemini 2.5 Flash); joint image-text + QA translation
4. Image Synthesis	Inpaint-then-render; diffusion text editing; plain compositing	Realism vs. script support and cost	LaMa inpainting + Pillow/PangoCairo rendering
5. QA Generation	LLM generation; rule-based templates; human authoring	Linguistic diversity vs. hallucination risk	LLM generation (Gemini 2.5 Flash); 4-type taxonomy
6. Quality Filtering	Image-level; translation; synthesis artifact; QA consistency; human audit	Benchmark integrity vs. dataset size	Automated filters at each stage

Table 1. Summary of the synthetic ST-VQA benchmark construction framework. Each row describes one phase, the space of methods, the core tradeoff, and the choice made in AfriST-VQA.

images are merged and subjected to uniform image-level quality filters to select images that (a) contain multiple text regions amenable to translation and (b) resemble authentic real-world scenes.

We retain only images satisfying all of the following criteria: minimum resolution of 400×400 pixels; between 3 and 10 annotated text regions; combined annotation character count of at least 15; at most 25% of image area covered by annotated text regions; average annotation length of at least 5 characters per region; at least 3 distinct text strings; a mean Zipf word-frequency score of at least 3.0 [32]; and an alphabetic-character ratio of at least 0.75 in the combined text.

For each image, we generate a natural-language scene caption using BLIP-Large [18] (`Salesforce/blip-image-captioning-large`), used to provide scene context to the translation model.

4.2. English QA Generation

For each image, we generated four multiple-choice questions using Gemini 2.5 Flash, supplied with the image and a complete list of text strings. We instructed the model to generate one question each of the reading, reasoning, spatial, and verification types (Table 2), representing different MLLM tasks and isolating model failures. Each question is a four-choice MCQ with exactly one correct answer and three distractors.

In reading and spatial categories, incorrect answers are different text strings in the same image; for reasoning, they are alternate scene or product categories; for verification, incorrect answers are drawn from {No, Yes, Maybe, Unclear} depending on the correct answer.

Type	Description
Reading	Directly read a text string from the image.
Reasoning	Infer scene type, purpose, or product category from visible text.
Spatial	Identify the relative position of one text string to another.
Verification	Confirm or deny the presence of a specific text string.

Table 2. The four question types in AfriST-VQA.

4.3. Translation to African Languages

We translate all English QA pairs and text strings to eight African languages: Swahili, Yoruba, Hausa, Igbo, Kinyarwanda, Somali, Amharic, and Tigrinya. The first six use the Latin script; Amharic and Tigrinya use the Ge'ez script. We perform translation using Gemini 2.5 Flash by requesting translations of MCQ questions and image text simultaneously, giving the model scene context to ensure consistency between questions and generated images. Amharic and Tigrinya translations are required to be in Ge'ez script (except proper nouns and brand names); Yoruba and Igbo translations must include diacritics. We filter out records where more than 50% of originally translatable strings remain untranslated, or where Amharic/Tigrinya strings contain no Ge'ez script.

Figure 1. Image generation process and sample images from AfriST-VQA.

4.4. Synthetic Scene-Text Generation

To create benchmark images, we replace every annotated English text region with its translated counterpart in a three-stage pipeline: oriented bounding box extraction, neural inpainting, and re-rendering.

Oriented Bounding Box Extraction. The original COCO-Text and TextVQA annotations provide axis-aligned bounding boxes that overestimate true text extent when text is rotated. We re-annotate each region with a tight oriented bounding box (OBB) by estimating the dominant text orientation via edge detection and Hough line estimation on a grayscale crop. Regions with insufficient edge evidence fall back to the original bounding box.

Neural Inpainting. Given the rotated bounding box corners, we build a binary inpainting mask by filling each polygon with OpenCV's `fillPoly` and dilating by 4 pixels to cover font edges. We then run LaMa inpainting [33] to produce a clean background with the original text removed.

Text Re-Rendering. Translated text is re-rendered onto the inpainted background at the correct position, angle, and visual style. For the six Latin-script languages, we use Pillow with Noto Sans/Serif fonts; font weight, condensedness, and serif style are inferred from the original image region. Text ink color is estimated by running K-means on Canny-edge pixels of the original region.

For Amharic and Tigrinya, we use PangoCairo with the Noto Sans Ethiopic typeface to ensure correct syllabic shaping of Ge'ez characters, which Pillow's FreeType back-end does not handle reliably. A light Gaussian blur and pixel-level noise are applied to the rendered text patch to reduce the pasted-on appearance.

Model	Citation
Gemini 3.0 Flash	[35]
Claude Haiku 4.5	[6]
GPT-4o-mini	[15]
Qwen3-VL-32B	[39]

Table 3. MLLMs evaluated on AfriST-VQA.

Model	Acc.	Δ	95% CI
Gemini 3.0 Flash	**71.3**	+46.3	[69.1, 73.5]
Claude Haiku 4.5	58.2	+33.2	[55.7, 60.8]
GPT-4o-mini	55.3	+30.3	[52.9, 57.6]
Qwen3-VL-32B	47.8	+22.8	[45.4, 50.1]
Chance baseline	25.0	—	—

Table 4. Overall accuracy (%) on AfriST-VQA. Δ = improvement over random chance. 95% CIs via binomial proportion.

5. Experimental Setup

5.1. Models

We evaluate four diverse MLLMs: three proprietary and one open-weight model (Table 3).

5.2. Evaluation Protocol

All models are queried at `temperature=0.0`. To minimize API costs, all four questions for a given (image, language) pair are batched into a single API call. Each call sends the synthetic benchmark image together with all four translated questions and MCQ choices, and instructs the model to respond with only the answer letters in the format `Q1: A`, `Q2: B`, etc. (`max_tokens=32`). We report accuracy as the percentage of correct answers. A random chance baseline is 25% for four-choice MCQ.

6. Results

Table 4 shows overall accuracy for each of the four MLLMs on AfriST-VQA. All four models substantially exceed the random-chance baseline, demonstrating that MLLMs possess some capacity for scene-text understanding in African languages. Gemini 3.0 Flash achieves the highest accuracy of 71.3%, while Claude Haiku 4.5 and GPT-4o-mini score 58.2% and 55.3%, respectively. Qwen3-VL-32B performs worst at 47.8%.

Table 5 shows per-language accuracy. Performance is generally greater for Latin-script languages compared to Ge'ez-script languages, with Tigrinya scoring lowest across three of the four models. Gemini 3.0 Flash is the only model to score above 60% in all eight languages. Qwen3-VL-32B performs competitively on Swahili but much lower on Yoruba and Hausa, suggesting model performance is sensi-

tive to individual language identity.

Table 6 shows accuracy by question type. Reasoning questions are easiest—Gemini 3.0 Flash achieves 87.3%—likely because they rely on world knowledge alongside visual context rather than precise OCR. Spatial questions are most challenging across all models: Gemini 3.0 Flash scores 60.4% and Qwen3-VL-32B scores 34.2%, only slightly above chance. Verification and reading fall between these extremes.

Table 7 shows per-language accuracy by question type. The gap between reasoning and spatial in Tigrinya is striking: 65.5% vs. 33.0%, suggesting unfamiliar scripts compound spatial grounding difficulty. Swahili achieves the highest average accuracy across models (65.2%) while Tigrinya is lowest (50.3%), a 14.9-point gap.

7. Discussion

All models greatly exceed the random-chance baseline, demonstrating that modern MLLMs possess scene-text understanding in African languages. Performance is highly uneven, with a 23.5-point gap between the strongest and weakest models and a 14.9-point gap between Swahili (highest) and Tigrinya (lowest). Question type is a strong factor: reasoning questions are much easier than spatial questions, with Qwen3-VL-32B scoring 34.2% on spatial and 60.6% on reasoning.

The advantage of proprietary models is consistent with IrokoBench [5], which evaluated 14 LLMs across 16 African languages and found that proprietary models substantially outperform open-weight models. Our results mirror this pattern and suggest imbalanced pretraining as a driver of performance disparity. The gap between Tigrinya and Latin-script languages is consistent with challenges for Ge'ez script OCR [21], which persist even in specialized models [8]. The difficulty of spatial questions is consistent with recent work showing spatial reasoning is a general weakness of MLLMs [10], compounded in low-resource settings.

Notably, Amharic outperforms several Latin-script languages (62.5% average) despite using the Ge'ez script, suggesting greater Amharic representation in multilingual pretraining. The sharp drop in Qwen3-VL-32B on Yoruba and Hausa relative to Swahili may indicate that Qwen's multilingual representations are highly sensitive to individual language identity rather than script.

Future work should replace source images with images collected in African contexts, extend coverage to Arabic-script African languages and tonal languages with complex diacritics, and incorporate open-ended evaluation formats to enable finer-grained analysis of model failures.

8. Conclusion

We introduce AfriST-VQA, the first scene-text VQA benchmark spanning eight African languages across two scripts, and a reusable six-phase framework for synthetic ST-VQA benchmark construction. Evaluating four MLLMs, we find all models exceed random chance but performance is highly uneven: a 23.5-point gap across models, a 14.9-point gap across languages, and consistent difficulty on spatial reasoning. Script familiarity and per-language resource availability drive the performance disparity. We hope AfriST-VQA and the accompanying framework serve as a foundation for equitable multimodal evaluation and motivate future work on low-resource African languages in NLP and OCR.

Limitations. AfriST-VQA is built from primarily Western scene-text corpora, limiting cultural validity. All translations were produced automatically; LLM translation quality varies substantially across languages and may contain errors. The rendering technique introduces artifacts not present in authentic scene-text images. Coverage is limited to eight languages and two scripts out of Africa's 2,000+. All questions are multiple choice, which introduces guessing and may allow models to exploit superficial cues, and prevents detection of script-level errors that would be exposed in open-ended evaluation. Translation quality was not verified by native speakers. For low-resource languages, LLM translation errors may corrupt ground-truth labels, meaning some reported model failures may reflect benchmark error rather than model incapability. Translation and rendering errors compound unpredictably as a model answering correctly may be exploiting a rendering artifact, and a model answering incorrectly may be penalized by a mistranslated ground truth, making reported accuracies noisy baselines rather than precise capability measurements. Finally, a limitation of our evaluation design is that Gemini 2.5 Flash was used for benchmark construction and is from the same model family as Gemini 3.0 Flash, the top-performing model. This self-evaluation bias may inflate Gemini's reported scores, and future work should construct benchmarks using translation and question generation models independent of those being evaluated.

References

[1] Ife Adebara and Muhammad Abdul-Mageed. Towards afrocentric NLP for African languages: Where we are and where we can go. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3814–3841, Dublin, Ireland, 2022. Association for Computational Linguistics. 1, 2

[2] Ife Adebara, AbdelRahim A Elmadany, and Muhammad Abdul-Mageed. Improving african language identification with multi-task learning. In *4th Workshop on African Natural Language Processing*, 2023. 2

Model	Swa	Yor	Hau	Igb	Kin	Som	Amh†	Tir†	Avg.
Gemini 3.0 Flash	**71.2**	**68.4**	**74.5**	**70.3**	**73.1**	**73.1**	**75.0**	**64.6**	**71.3**
Claude Haiku 4.5	60.5	59.1	52.8	57.1	58.2	61.2	63.3	52.1	58.0
GPT-4o-mini	63.7	50.9	56.1	59.4	56.6	58.5	53.8	42.9	55.2
Qwen3-VL-32B	65.1	40.6	35.8	49.5	45.3	45.8	58.0	42.0	47.8

Table 5. Accuracy (%) broken down by language. †Ge'ez script (Ethiopic). Swa=Swahili, Yor=Yoruba, Hau=Hausa, Igb=Igbo, Kin=Kinyarwanda, Som=Somali, Amh=Amharic, Tir=Tigrinya.

Model	Read.	Reas.	Spat.	Verif.	Avg.
Gemini 3.0 Flash	**69.1**	**87.3**	**60.4**	**68.4**	**71.3**
Claude Haiku 4.5	58.7	67.6	41.9	64.8	58.2
GPT-4o-mini	51.4	73.6	39.6	56.4	55.2
Qwen3-VL-32B	49.3	60.6	34.2	46.9	47.8

Table 6. Accuracy (%) by question type.

Lang.	Read.	Reas.	Spat.	Verif.	Avg.
Swahili	58.4	**83.7**	**50.2**	**68.4**	**65.2**
Amharic	**63.2**	80.4	42.6	63.7	62.5
Somali	55.3	73.8	42.7	66.5	59.6
Igbo	60.0	70.2	48.3	58.0	59.1
Kinyarwanda	57.2	70.2	45.2	60.6	58.3
Hausa	51.0	65.2	48.5	54.9	54.9
Yoruba	56.0	70.0	41.5	50.5	54.5
Tigrinya	55.2	65.5	33.0	47.4	50.3

Table 7. Average accuracy (%) per language and question type, aggregated across all models.

[3] David Ifeoluwa Adelani. Natural language processing for african languages, 2025. 1

[4] David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, et al. Masakhaner: Named entity recognition for african languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131, 2021. 2

[5] David Ifeoluwa Adelani, Jessica Ojo, Israel Abebe Azime, Jian Yun Zhuang, Jesujoba Alabi, Xuanli He, Millicent Ochieng, Sara Hooker, Andiswa Bukula, En-Shiun Annie Lee, et al. Irokobench: A new benchmark for african languages in the age of large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2732–2757, 2025. 5

[6] Anthropic. Claude 4.5 haiku, 2025. Large language model. 4

[7] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9365–9374, 2019. 2

[8] Birhanu Hailu Belay, Isabelle Guyon, Tadele Mengiste, Bezawork Tilahun, Marcus Liwicki, Tesfa Tegegne, and Romain Egele. A historical handwritten dataset for ethiopic ocr with baseline models and human-level performance. In *International Conference on Document Analysis and Recognition*, pages 23–38. Springer, 2024. 5

[9] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marcal Rusinol, Ernest Valveny, C.V. Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 1, 2

[10] Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465, 2024. 5

[11] Henry Gagnier, Sophie Gagnier, and Ashwin Kirubakaran. Kazakhocr: A synthetic benchmark for evaluating multimodal models in low-resource kazakh script ocr, 2026. 2

[12] Mridul Ghosh, Himadri Mukherjee, Sk Md Obaidullah, Xiao-Zhi Gao, and Kaushik Roy. Scene text understanding: recapitulating the past decade. *Artificial Intelligence Review*, 56 (12):15301–15373, 2023. 1

[13] Raymond G Gordon Jr. Ethnologue, languages of the world. *http://www. ethnologue. com/*, 2005. 1

[14] Ijazul Haq, Yingjie Zhang, and Irfan Ali Khan. Psocr: Benchmarking large multimodal models for optical character recognition in low-resource pashto language. *arXiv preprint arXiv:2505.10055*, 2025. 1, 2

[15] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 4

[16] Taebaek Hwang, Minseo Kim, Gisang Lee, Seonuk Kim, and Hyunjun Eun. KRETA: A benchmark for Korean reading and reasoning in text-rich VQA attuned to diverse visual contexts. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33421–33432, Suzhou, China, 2025. Association for Computational Linguistics. 2

[17] Zan-Xia Jin, Pinle Qin, Suzhen Lin, Jia Qin, Shuangjiao Zhai, Jianchao Zeng, and Xu-Cheng Yin. Robust scene text understanding with ocr token and word alignment for text-vqa and text-caption. *Pattern Recognition*, page 112362, 2025. 1

[18] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International*

conference on machine learning, pages 12888–12900. PMLR, 2022. 3

[19] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization, 2019. 2

[20] Xuejing Liu, Wei Tang, Xinzhe Ni, Jinghui Lu, Rui Zhao, Zechao Li, and Fei Tan. What large language models bring to text-rich vqa?, 2023. 1

[21] Ruchika Malhotra and Maru Tesfaye Addis. Handwritten amharic word recognition with additive attention mechanism. *IEEE Access*, 12:114645–114657, 2024. 5

[22] Haq Nawaz Malik. 600k-ks-ocr: a large-scale synthetic dataset for optical character recognition in kashmiri script. *arXiv preprint arXiv:2601.01088*, 2026. 2

[23] Haq Nawaz Malik, Kh Mohmad Shafi, and Tanveer Ahmad Reshi. synthocr-gen: A synthetic ocr dataset generator for low-resource languages- breaking the data barrier, 2026. 1

[24] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 947–952, 2019. 1

[25] Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Abinew Ali Ayele, Nedjma Ousidhoum, David Ifeoluwa Adelani, Seid Muhie Yimam, Ibrahim Sa'id Ahmad, Meriem Beloucif, Saif M. Mohammad, Sebastian Ruder, Oumaima Hourrane, Pavel Brazdil, Alipio Jorge, Felermino Dário Mário António Ali, Davis David, Salomey Osei, Bello Shehu Bello, Falalu Ibrahim, Tajuddeen Gwadabe, Samuel Rutunda, Tadesse Belay, Wendimu Baye Messelle, Hailu Beshada Balcha, Sisay Adugna Chala, Hagos Tesfahun Gebremichael, Bernard Opoku, and Stephen Arthur. AfriSenti: A Twitter sentiment analysis benchmark for African languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13968–13981, Singapore, 2023. Association for Computational Linguistics. 2

[26] Keziah Naggita, Julienne LaChance, and Alice Xiang. Flickr africa: Examining geo-diversity in large-scale, human-centric visual data. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 520–530, 2023. 1

[27] Hieu Minh Nguyen, Tam Le-Thanh Dang, and Kiet Van Nguyen. Towards signboard-oriented visual question answering: Visignvqa dataset, method and benchmark, 2025. 2

[28] Iroro Orife, Julia Kreutzer, Blessing Sibanda, Daniel Whitenack, Kathleen Siminyu, Laura Martinus, Jamiil Toure Ali, Jade Abbott, Vukosi Marivate, Salomon Kabongo, et al. Masakhane–machine translation for africa. *arXiv preprint arXiv:2003.11529*, 2020. 2

[29] Naresh Saini, Promodh Pinto, Aravinth Bheemaraj, Deepak Kumar, Dhiraj Daga, Saurabh Yadav, and Srihari Nagaraj. Ocr synthetic benchmark dataset for indic languages. *arXiv preprint arXiv:2205.02543*, 2022. 2

[30] Thapelo Andrew Sindane and Vukosi Marivate. From n-grams to pre-trained multilingual models for language identification. In *Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities*, pages 229–239, Miami, USA, 2024. Association for Computational Linguistics. 2

[31] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[32] Robyn Speer, Joshua Chin, Andrew Lin, Sara Jewett, and Lance Nathan. wordfreq: v3.0, 2022. 3

[33] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2149–2159, 2022. 2, 4

[34] Jingqun Tang, Qi Liu, Yongjie Ye, Jinghui Lu, Shu Wei, An-Lan Wang, Chunhui Lin, Hao Feng, Zhen Zhao, Yanjie Wang, Yuliang Liu, Hao Liu, Xiang Bai, and Can Huang. MTVQA: Benchmarking multilingual text-centric visual question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7748–7763, Vienna, Austria, 2025. Association for Computational Linguistics. 1, 2

[35] Gemini Team. Gemini: A family of highly capable multimodal models, 2025. 4

[36] Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. Anytext: Multilingual visual text generation and editing. *arXiv preprint arXiv:2311.03054*, 2023. 2

[37] Quan Van Nguyen, Dan Quang Tran, Huy Quang Pham, Thang Kien-Bao Nguyen, Nghia Hieu Nguyen, Kiet Van Nguyen, and Ngan Luu-Thuy Nguyen. Vitextvqa: A large-scale visual question answering dataset and a novel multimodal feature fusion method for vietnamese text comprehension in images. *Expert Systems with Applications*, 308: 130839, 2026. 2

[38] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images, 2016. 2

[39] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 4

A. LLM Prompts

A.1. QA Generation

System:

```
You are building an ST-VQA benchmark. Generate
↪  questions
that require reading visible text in the image to
↪  answer.
You will be given the exact list of all confirmed
↪  text
strings in the image -- only reference these,
↪  never invent
```

```
other text. Respond with valid JSON only -- no
↪  markdown,
no preamble.
```

User:

```
The following text strings are confirmed to
↪  appear in
this image:
  - "<string_1>"
  - "<string_2>"  ...

Generate exactly 4 questions, one of each type:
"reading", "reasoning", "spatial",
↪  "verification".
For EACH provide: "question", "answer" (1-5
↪  words),
"distractors" (EXACTLY 3 wrong answers).

Distractor rules:
- reading/spatial: other strings from the
↪  confirmed list
- reasoning: plausible alternative scene/product
↪  types
- verification: ["No","Maybe","Unclear"] if
↪  answer="Yes",
  ["Yes","Maybe","Unclear"] if answer="No"
- Spatial: ask about position relative to other
↪  text;
  answer must be a string from the confirmed list
- Verification: only ask about strings in the
↪  confirmed
  list

Respond ONLY with this JSON:
{"questions": [
  {"type":"reading","question":"...","answer":".
  ↪  ..",
   "distractors":["...","...","..."]},
  {"type":"reasoning", ...},
  {"type":"spatial", ...},
  {"type":"verification", ...}
]}
```

A.2. Translation

System:

```
You are a professional multilingual translator
specializing in African languages helping build
↪  an OCR
VQA benchmark.

Critical rules:
- Amharic/Tigrinya: Ge'ez/Ethiopic script ONLY.
↪  NEVER
  romanize. Proper nouns and brand names may stay
  ↪  as-is.
- Yoruba/Igbo: include ALL correct tonal
↪  diacritics.
- Text strings: translate as they appear on a
↪  real sign
  -- short and natural. Keep untranslatable terms
  ↪  as-is.
- QA consistency: translated answer must match
↪  translated
  text strings. Translate all 4 MCQ choices as a
  ↪  set.
- Return complete, valid JSON only. No markdown,
↪  no
  preamble. No trailing commas. Close all
  ↪  brackets.
```

User:

```
Scene context: "<blip_caption>"

Text strings visible in the image:
["<string_1>", "<string_2>", ...]

QA pairs:
[{"id":0,"type":"reading","question":"...",
  "answer":"...",
  "choices":{"A":"...","B":"...","C":"...","D":"
  ↪  ..."}},
 ...]

Translate into:
  swahili:     standard written Swahili, Latin
  ↪  script
  yoruba:      include ALL tonal diacritics, Latin
  ↪  script
  hausa:      standard written Hausa, Latin script
  igbo:       include all diacritics, Latin script
  kinyarwanda: standard written Kinyarwanda,
  ↪  Latin script
  somali:      standard written Somali, Latin
  ↪  script
  amharic:     Ge'ez/Ethiopic script ONLY
  tigrinya:    Ge'ez/Ethiopic script ONLY

Respond ONLY with this JSON structure:
{"swahili":{"text_strings":["..."],"questions":[
   {"id":0,"question":"...","answer":"...",
    "choices":{"A":"...","B":"...","C":"...","D"
    ↪  :"..."}}
  ]}, "yoruba":{...}, ..., "tigrinya":{...}}
```

A.3. MLLM Evaluation

```
Look at the image carefully and answer ALL 4
multiple-choice questions below.
```

```
Rules:
- Answer EVERY question.
- Respond with ONLY the answer letters in this
↪  exact
  format, nothing else:
Q1: <letter>
Q2: <letter>
Q3: <letter>
Q4: <letter>

--- Questions ---
Q1 [reading]:      <question>
  A) <choice_A>  B) <choice_B>
  C) <choice_C>  D) <choice_D>

Q2 [reasoning]:    <question>
  A) <choice_A>  B) <choice_B>
  C) <choice_C>  D) <choice_D>

Q3 [spatial]:      <question>
  A) <choice_A>  B) <choice_B>
  C) <choice_C>  D) <choice_D>

Q4 [verification]: <question>
  A) <choice_A>  B) <choice_B>
  C) <choice_C>  D) <choice_D>

Your answers:
```