TextInVision: Text and Prompt Complexity Driven Visual Text Generation Benchmark

Supplementary Material

A. TextInVision prompt set

In this section, we detail the structure of the TextInVision prompt set. As discussed previously, TextInVision separates the prompts from the textual content that needs to be embedded within images. This separation allows for independent evaluation of model capabilities and aids in pinpointing the causes of errors. Each prompt is associated with specific attributes to facilitate the selection and analysis of the results. For example:

> **Prompt:** A paper with the words "Vacation calories don't count. Right? - Unknown" written on it.
>
> **Attributes:**
>
> - **Prompt Type:** Simple
> - **ID:** s119938
> - **Text:** "Vacation calories don't count. Right? - Unknown"
> - **Character Length:** 41
> - **Word Count:** 7
> - **Oxford Category:** None
> - **Contains Rare Words:** No
> - **Contains Gibberish:** No
> - **Contains Special Characters:** Yes
> - **Contains Numbers:** No
> - **Is a Sentence:** Yes
> - **LAION Frequency:** 0
> - **COCO Frequency:** 0
> - **Text Source:** `https://www.shutterfly.com/ideas/what-to-write-in-a-holiday-card/`

Another example is as follows.

> **Prompt:** A science fair with students presenting projects, a display board titled "explore" with diagrams and data, judges evaluating.
>
> **Attributes:**
>
> - **Prompt Type:** Complex
> - **ID:** c119237
> - **Text:** "explore"
> - **Character Length:** 7
> - **Word Count:** 1
> - **Oxford Category:** B1
> - **Contains Rare Words:** No
> - **Contains Gibberish:** No
> - **Contains Special Characters:** No
> - **Contains Numbers:** No
> - **Is a Sentence:** No
> - **LAION Frequency:** 467
> - **COCO Frequency:** 8
> - **Text Source:** `https://www.oxfordlearnersdictionaries.com/wordlists/oxford3000-5000`

In these examples, the *Text Source* attribute denotes the origin of the textual content. The detailed attributes provide insights into various aspects of the text, such as length, complexity, and frequency in existing datasets, which are crucial for analyzing model performance under different conditions.

In addition to general and complex prompts that include various words, phrases, and longer texts, we have incorporated prompts derived from real-world situations that often require images containing visual text. By focusing on real-world scenarios, we aim to enable meaningful evaluations and push the boundaries of current models toward practical applications across different domains.

An example from the *Advertisements* category is:

> **Prompt:** A local fair with signs at all entrances reading the text "Free Entry Today".
>
> **Attributes:**
>
> - **Group:** Advertisements
> - **Text:** Free Entry Today
> - **Character Length:** 16
> - **Word Count:** 3
> - **ID:** r112277

To provide a clearer picture of these real-world prompts, Figure 1 illustrates their distribution across six primary categories, each further divided into subcategories that highlight their respective contributions to the overall dataset.

To showcase how the models perform on prompts from different categories, we present more examples in Figure 2. This figure includes prompts and the corresponding images generated by the seven models evaluated in this study. These examples illustrate the models' capabilities and limitations when handling various types of textual content within images.

B. OCR & edit distance score

To extract text from images, we initially utilized three OCR models: Llava Next [3], EasyOCR [1], and PaddleOCR [2].

Figure 1. Breakdown of the TextInVision dataset's real-world prompts into six primary categories and their subcategories, highlighting the diversity and proportional representation within the dataset

As our evaluation progressed, we opted to use EasyOCR for the remaining tasks due to its faster processing speed and consistent performance. While Llava Next was capable of accurate text recognition, it lacked sufficient reliability for our requirements. PaddleOCR did not offer significant advantages over EasyOCR, reinforcing our decision to rely exclusively on EasyOCR.

To comprehensively analyze the OCR results, we employed three methods, each providing a unique perspective on the accuracy and quality of the text extraction:

- **Edit Distance** [5]: Measures the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into another. This metric is useful for evaluating the degree of similarity between the OCR output and the ground truth, especially in cases of partial matches.
- **Longest Common Subsequence (LCS)**: Identifies the longest sequence of characters that appear in both the OCR result and the ground truth in the same order, but not necessarily consecutively. This method allows for gaps or misspellings and provides insight into the overall similarity between the two texts.
- **Longest Ordered Match (LOM)**: Focuses on finding the longest sequence of characters that appear in both texts while maintaining their original order. This metric offers insight into how well the OCR preserves the text structure and sequence.

For example, when comparing the ground truth "Digital Dreamscapes" with the OCR result "Dlgitoi Draseampes":

- Edit Distance
  - Edits needed:
    1. Substitute 'i' with 'l'

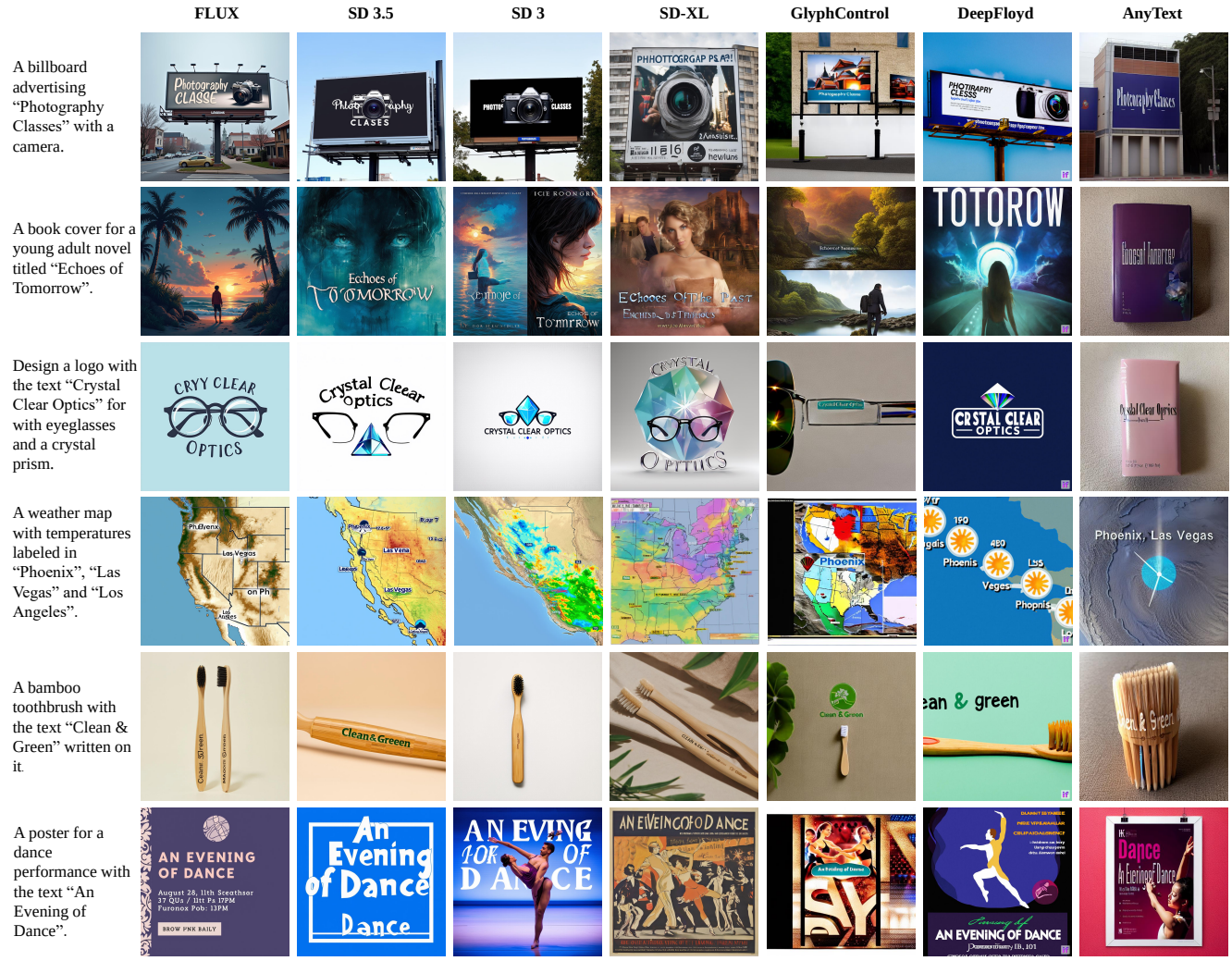


Figure 2. Examples of real-world prompts and the images generated by seven models, demonstrating their performance.

2. Substitute 'a' with 'o'
3. Substitute 'l' with 'i'
4. Substitute 'e' with 'a'
5. Substitute 'a' with 's'
6. Substitute 'm' with 'p'
7. Substitute 'c' with 'e'
8. Substitute 'e' with 's'
9. Delete the final 's'

– Total edits: 12 (A lower number signifies a closer match between the OCR result and the ground truth.)

- LCS
  - LCS: "Dgit Dreamapes"
  - Length of LCS: 13
- LOM
  - LOM: "git eam"
  - Length of LOM: 6

Given that T2I models can produce more text than desired, ensuring that the extracted text is accurate and aligns with the prompts is crucial. Therefore, we used Algorithm 1 that incorporates these three methods to analyze the OCR results and evaluate the model's performance. This algorithm systematically evaluates the OCR results by calculating the edit distance and identifying the LCS, providing a comprehensive assessment of the OCR model's performance.

In this algorithm, the variables are *ocr_text*, representing the OCR output text; *exp_text*, the expected text to compare against; *dist*, the computed edit distance; *words*, a list of words extracted from *ocr_text*; *max_lcs_len*, the maximum length of the LCS found; *best_match*, the word from *ocr_text* that best matches *exp_text*; *exp_words* and *ocr_words*, lists of words from *exp_text* and *ocr_text*, respectively; and *rem_exp* and *rem_ocr*, the remaining texts after removing common words. The functions used are SPLITWORDS(TEXT), which splits the text into a list of words;

Algorithm 1 Edit distance based score

Input: *ocr_text*, *exp_text*
Output: *dist*
Read *ocr_text*
if *exp_text* is a substring of *ocr_text* **then**
    dist $\leftarrow$ 0
else
    if *exp_text* is a single word **then**
        words $\leftarrow$ SPLITWORDS(*ocr_text*)
        max_lcs_len $\leftarrow$ 0
        best_match $\leftarrow$ empty string
        for all *w* in *words* **do**
            lcs $\leftarrow$ LCS(*w*, *exp_text*)
            if LENGTH(*lcs*) $>$ *max_lcs_len* **then**
                max_lcs_len $\leftarrow$ LENGTH(*lcs*)
                best_match $\leftarrow$ *w*
            end if
        end for
        dist $\leftarrow$ EDITDISTANCE(*best_match*, *exp_text*)
    else
        exp_words $\leftarrow$ SPLITWORDS(*exp_text*)
        ocr_words $\leftarrow$ SPLITWORDS(*ocr_text*)
        for all *w* in *exp_words* **do**
            if *w* is in *ocr_words* **then**
                Remove *w* from *exp_words*
                Remove *w* from *ocr_words*
            end if
        end for
        rem_exp $\leftarrow$ JOINWORDS(*exp_words*)
        rem_ocr $\leftarrow$ JOINWORDS(*ocr_words*)
        dist $\leftarrow$ EDITDISTANCE(*rem_ocr*, *rem_exp*)
    end if
end if
return *dist*

JOINWORDS(WORDS), which joins a list of words into a single string; LCS(A, B), which computes the longest common subsequence between strings a and b; EDITDISTANCE(A, B), which calculates the edit distance between two strings; and LENGTH(S), which returns the length of the string s.

C. Human evaluation

We conducted a human evaluation study to validate our TextInVision prompt set for assessing T2I models' ability to generate images with accurate embedded text, based on the hypothesis that current models struggle to fully reflect prompts in this aspect. The primary objective was to determine whether our edit distance-based score aligns with human judgments and confirms our comparisons of model performance, highlighting their shortcomings.

To achieve this, a total of 66 participants were recruited for this evaluation, hosted on a Hugging Face Space (see Figure 3). This accessible and user-friendly platform enabled participants to review images and submit their responses efficiently.

The evaluation dataset consisted of 1,000 images, randomly selected to ensure fairness and comprehensive coverage across all categories and models in our study. We ensured that at least 10 images from each group (combination of category and model) were included to maintain equitable representation. This stratified random sampling approach aimed to minimize selection bias and provide a balanced assessment of each model's performance.

Given the large number of images, it was not feasible for a single person to evaluate all and on average, each participant evaluated 30 images. To assess the consistency, we calculated the agreement per image and the average agreement across all images and participants. For the prompt following criterion, the average agreement was 90.06%, and for the accuracy criterion, the average agreement was 88.53%.

These high agreement rates demonstrate that, despite each image being evaluated by different subsets of participants, there was a strong overall consistency in the assessments. This consistency enhances the credibility of our human evaluation methodology and supports the reliability of our conclusions regarding the performance of the T2I models. It indicates that the participants, even when evaluating different images, applied the evaluation criteria in a similar manner, reinforcing the validity of our findings about the models' shortcomings in generating images with accurate embedded text.

D. Detailed evaluation of VAE

In this section, we provide an additional information on the evaluation of the VAE component applied to the TexInVision image set. The following subsections offer further insights into the dataset organization and the metrics used in our analysis.

TexInVision image set

The dataset is organized within a JSON file, which facilitates easy access and management of the images. Each entry in the JSON file contains essential metadata, including the source, photographer profile link, and a link to get the image. This structured format ensures efficient handling and retrieval of data for further processing. Figure 4 presents more examples of VAE reconstructed images from the TextInVision image set.

Word retention & partial accuracy calculations

Word Retention (WR) and Partial Accuracy (PA) are essential metrics for comprehensively analyzing OCR results in the evaluation of VAEs. WR measures the proportion of words that are exactly retained by the OCR process com-

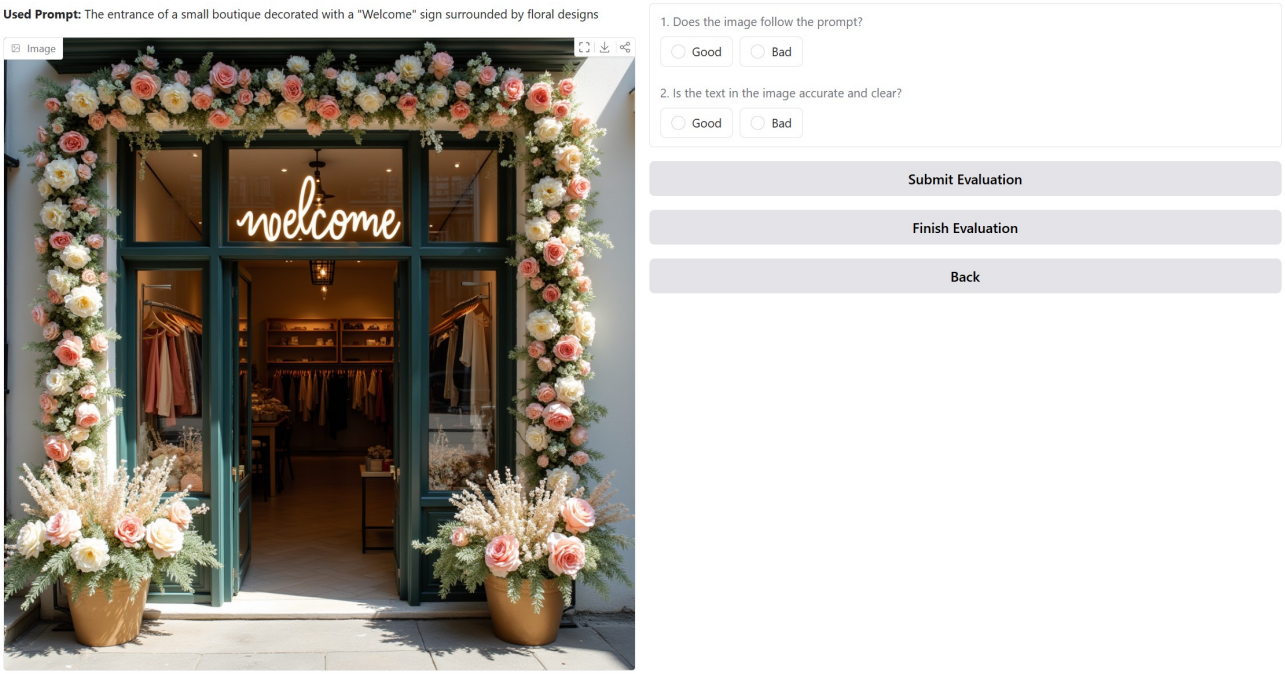
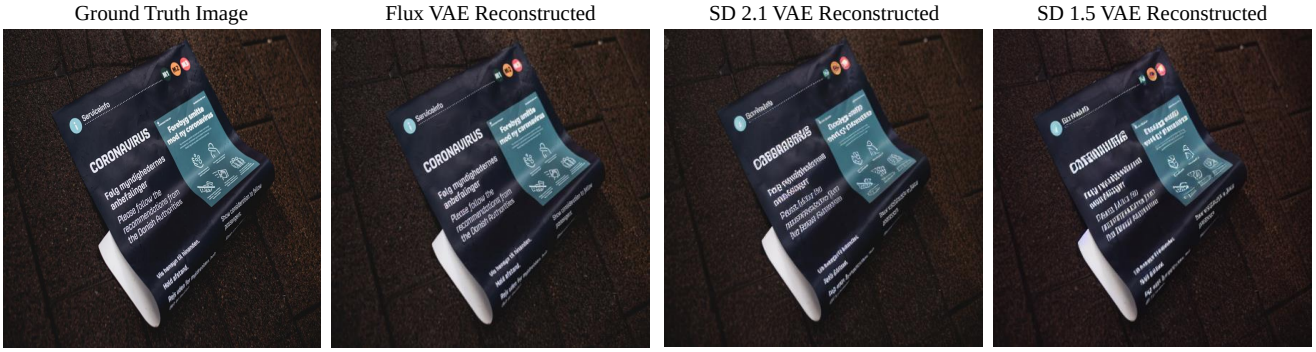


Figure 3. Examples of real-world prompts and the images generated by seven models, demonstrating their performance.



(a) Photo by Hjalte Gregersen on Unsplash [4].

(b) Photo by Mockup Free on Unsplash [4].

Figure 4. Assessing VAEs for reconstructing text-embedded images reveals noticeable detail degradation and errors in text representation.

pared to the ground truth text. It provides a straightforward metric to assess the accuracy of word-level recognition.

Let N_{correct} be the number of words exactly matched between the OCR output and the ground truth and N_{total} be the total number of words in the ground truth text. Then, WR is calculated as:

$$WR = \frac{N_{\text{correct}}}{N_{\text{total}}} \times 100\% \tag{1}$$

However, solely relying on exact word matches can overlook subtle yet significant errors within recognized words. This is where PA becomes crucial, as it assesses the similarity of partially correct words by evaluating the letter-level edit distance between the OCR output and the ground truth.

Let W_i be the i-th word in the ground truth, $\hat{W}_i$ be the corresponding OCR-recognized word, $d(W_i, \hat{W}_i)$ be the Levenshtein edit distance between W_i and $\hat{W}_i$ and $L(W_i)$ be the number of letters in W_i. Then, PA is calculated as:

$$PA = \left(1 - \frac{\sum_{i=1}^{N_{\text{total}}} d(W_i, \hat{W}_i)}{\sum_{i=1}^{N_{\text{total}}} L(W_i)}\right) \times 100\% \tag{2}$$

Together, these metrics enable a nuanced evaluation of the VAE's performance in processing and reconstructing textual data, highlighting both its strengths and areas for improvement in maintaining the integrity of the original text.

References

[1] EasyOCR. Easyocr. `https://github.com/JaidedAI/EasyOCR`, 2020.

[2] Chenxia Li, Weiwei Liu, Ruoyu Guo, Xiaoting Yin, Kaitao Jiang, Yongkun Du, Yuning Du, Lingfeng Zhu, Baohua Lai, Xiaoguang Hu, et al. Pp-ocrv3: More attempts for the improvement of ultra lightweight ocr system. *arXiv preprint arXiv:2206.03001*, 2022.

[3] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024.

[4] Unsplash. Unsplash. `https://unsplash.com/license`, 2024. Accessed: 2024-04-01.

[5] Li Yujian and Liu Bo. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095, 2007.