

Refining AI-Generated Images by Correcting Text Hallucinations

Amaan Feroz
Dept. of CSE
PES University
Bengaluru, India

Ananya J
Dept. of CSE
PES University
Bengaluru, India

Ananya Mahishi
Dept. of CSE
PES University
Bengaluru, India

Archishman VB
Dept. of CSE
PES University
Bengaluru, India

Dr. Surabhi Narayan
Dept. of CSE
PES University
Bengaluru, India

Abstract—Recent advancements in artificial intelligence (AI) have led to the development of models capable of generating realistic images based on textual prompts. Concerns have been raised over the validity and dependability of AI-produced content due to the hallucinogenic nature of the generated material. In this work, we introduce a unique method for identifying and assessing artificial intelligence (AI) hallucinations in text-embedded images. We first describe the process of generating a diverse dataset of AI hallucinated images paired with textual prompts. Next, we present a methodology for evaluating the validity and coherence of generated information, which includes techniques such as manual annotation, human evaluation, and expert judgement. We demonstrate the efficacy of our approach through practical experiments and case studies, illustrating the obstacles and potential for detecting and minimising AI hallucinations. Finally, we will examine the consequences for AI ethics, responsible AI development, and future research objectives in this growing subject. Our work helps to advance understanding and awareness of AI-generated hallucinations, allowing for the development of reliable and trustworthy AI systems for image generation and content creation.

Index Terms—Text correction , Image processing , Text detection , Character recognition , Error correction , Large Language Models (LLMs), Generative AI, AI Hallucinations

I. INTRODUCTION

A. Introduction to AI-generated Images with Text

Recent developments in artificial intelligence (AI) have made it possible to create models that, given textual descriptions, can produce visuals in any required style. These artificial intelligence (AI)-generated graphics have enormous promise for a variety of uses, such as visual storytelling, design automation, and content production. Natural language processing (NLP) methods and deep learning architectures, including transformer models and generative adversarial networks (GANs), enable AI systems to comprehend verbal prompts and produce related visuals with impressive accuracy. One distinguishing feature of AI-generated graphics is the ability to incorporate textual content directly into the visual depiction. This combination of text and image creates new opportunities for delivering information, expressing creativity, and improving visual communication. However, the process of creating graphics with incorporated text presents distinct obstacles and considerations. The combination of textual and visual modalities necessitates careful consideration to

guarantee that the created content is coherent, relevant, and interpretable.

B. Challenges in AI-generated Content

While AI-generated images with embedded text offer exciting opportunities, they also raise important questions about the reliability and validity of the generated content. In particular, there is growing concern about the potential for AI systems to produce hallucinatory images that may mislead or deceive viewers. These "AI hallucinations" can manifest as surreal scenes, distorted objects, or nonsensical combinations of visual and textual elements, posing challenges for users and developers alike. The issue that our paper focuses on is hallucinated text in AI generated images. This can be seen in various forms which can range from simple issues such as misspellings in words to more complex issues

C. Applications of AI generated images with embedded text

AI-generated images with embedded text offer a unique combination of visual content and written information, opening doors to several exciting applications: **1. Content Creation and Design** The process of developing visual content for marketing materials, commercials, social media postings, and website designs can be automated with the use of AI-generated images that have embedded text. **2. Educational Resources and Learning Materials** Can be used by educators and instructional designers to produce visually appealing course materials, learning resources, and instructional materials. **3. Product Design and Prototyping** Can help with iterative design, improve communication with stakeholders, and speed up the development of new goods and ideas. **4. Virtual and Augmented Reality (VR/AR) Experiences** Combining AI-generated visuals with integrated text. These photos can be used as virtual objects, background elements, or informational overlays in VR/AR simulations, games, training modules, and digital environments. **5. Data Visualization and Infographics** can be used by data analysts, journalists, and communicators to graphically represent information, statistics, and data. **6. Personalization and Customization** Organisations can improve user interfaces, product recommendations, marketing campaigns, and conversion rates by dynamically producing images depending on user choices, demographics, or behaviour.

II. RELATED WORK

The study related to character-aware models in improving visual text rendering [1] introduces a benchmark, DrawText, to assess the text rendering quality of text-to-image models. The benchmark has two parts: DrawText Spelling and DrawText Creative. DrawText Spelling evaluates the spelling ability of image generation models by creating prompts with 100 words selected from each of the English WikiSpell frequency buckets and evaluating them using optical character recognition (OCR) metrics. DrawText Creative assesses end-to-end text rendering across varied visual situations by collaborating with a professional graphic designer to create 175 challenges that require text to be rendered in a variety of creative styles and circumstances. The use of fine-grained tokenizers, such as character and position tokens, improves spelling accuracy. The research also includes tests to assess the spelling ability of text-to-image generative models using the suggested DrawText benchmark. The investigations compare character-aware models to character-blind models and discover that character-aware models perform better on image-text alignment, implying that making the text encoder character-aware can help close the performance gap. Furthermore, the research investigates whether text encoders that do not require text rendering can nevertheless do well on spelling tests. The results reveal that strictly character-level models perform worse on image-text alignment, whereas a hybrid character-aware model can improve text rendering without significantly reducing overall efficiency.

Chen, J., Huang, Y., Lv, T., Cui, L., Chen, Q., and Wei, F.'s article [2] delves further into the usage of language models to improve text rendering processes. They introduced TextDiffuser, a model which focuses on generating images with visually appealing text that is coherent with backgrounds. TextDiffuser consists of two stages: first, a Transformer model generates the layout of keywords extracted from text prompts, and then diffusion models generate images conditioned on the text prompt and the generated layout [3].

Recent work based on the TextDiffuser Model [4] emphasizes the need of creating extensive assessment datasets for visual text production, as well as assessing models' ability to interpret lengthy and unusual textual parts. The paper suggests training-free techniques to improve the TextDiffuser model for creating complicated visual text, furthering the field of visual text Image production. The authors introduce the LenCom-EVAL testbed, which is meant to challenge models with complicated scenarios and extensive textual parts, ensuring robustness and adaptation for real-world applications characterized by unpredictability and complexity. It emphasizes the difficulties encountered by previous models such as TextDiffuser, such as issues with extensive text, poor layout generation resulting in overlapping text, and inadequacies in rigidly adhering to text prompts. In response to these problems, the study proposes a method called Simulated Annealing and OCR-Aware Recursive In-Painting to reduce layout overlap and rectify spelling errors in output images.

These works demonstrate the importance of improving text rendering in visual content development. They present fresh benchmarks and approaches for improving the quality and inventiveness of text-to-image models. They increase spelling accuracy and image-text alignment significantly by utilizing fine-grained tokenizers and character-aware techniques. Furthermore, the emphasis on developing large assessment datasets and dealing with issues such as layout creation and text overlap emphasizes the task's complexity. While these initiatives are primarily concerned with enhancing existing models and training approaches, our approach differs by focusing on post-generation modifications to previously created images. By building on completed images and adopting unique methodologies, we hope to advance text correction in AI-generated visual material.

III. PROPOSED APPROACH

This paper proposes a three-part pipeline framework to mitigate AI hallucinations in text-guided image generation:

A. Text Detection with Custom-Trained YOLOv8

YOLOv8 (You Only Look Once) [5] is one of the latest iterations in a popular family of object detection algorithms. It excels at identifying and locating objects within images and videos, making it a valuable tool for computer vision tasks.

Core Functionality: Like its predecessors, YOLOv8 is a single-stage object detection model. This means it predicts bounding boxes and class probabilities for objects in a single forward pass through the neural network, making it fast and efficient.

Focus on Speed and Accuracy: YOLOv8 prioritizes both speed and accuracy. It achieves high mean Average Precision (mAP) scores on benchmark datasets while maintaining real-time inference capabilities.

Framework: While the specific framework used by Ultralytics, the developers of YOLOv8, is not publicly disclosed, it's believed to be heavily influenced by PyTorch, a popular deep learning framework known for its flexibility and ease of use.

Multiple Model Variants: YOLOv8 comes in various versions offering a trade-off between speed and accuracy. Smaller versions like YOLOv8n prioritize speed for real-time applications, while larger models like YOLOv8x deliver higher accuracy for tasks demanding precise object detection.

For the particular task of image detection due to various constraints in terms of time and hardware, we have chosen to deploy the YOLOv8s (small) model as it provided a good tradeoff between speed and accuracy.

A custom-trained YOLOv8 object detection model will be employed to identify and localize text regions within the generated image. YOLOv8 is chosen for its speed and accuracy in real-time object detection tasks. The model was trained on a dataset of text-annotated images specifically designed to capture the variations in font styles, sizes, and orientations that may appear in DALL-E or other AI generated images. The model detects and puts bounding boxes on regions of the image which contain text.

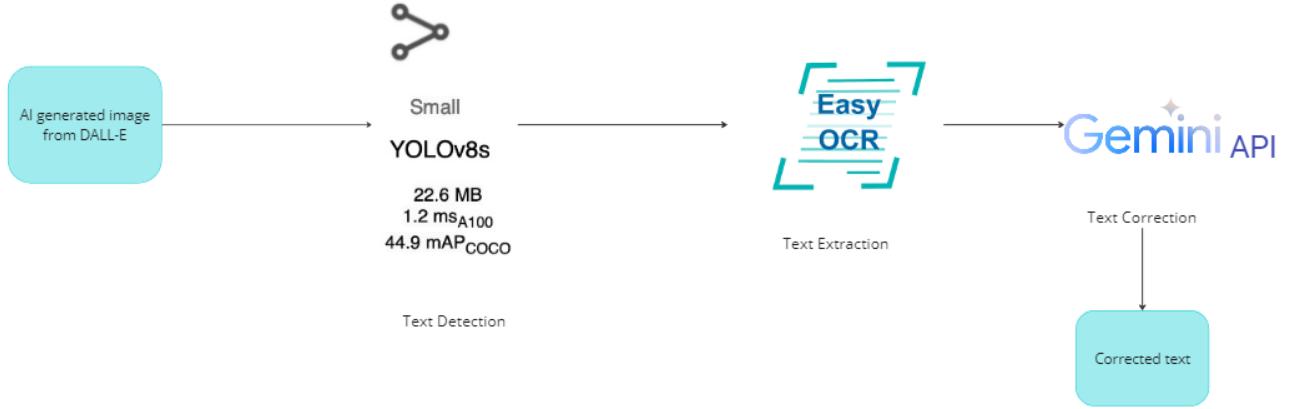


Fig. 1. A three part pipeline for detection and correction of hallucinated text in image

The equation behind the bounding box calculation is as follows;

$$\begin{aligned} b_x &= \sigma(t_x) + c_x, \\ b_y &= \sigma(t_y) + c_y, \\ b_w &= p_w e^{t_w}, \\ b_h &= p_h e^{t_h} \end{aligned} \quad (1)$$

YOLO also provides a confidence score alongside each detection that it makes to indicate the degree that its believes the detection exists , the math behind it being as follows:

$$\text{confidence} = \sigma(t_o) \quad (2)$$

The Loss function that this model makes use of is as follows:

$$\begin{aligned} \text{Loss} &= \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}^{obj} ij [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\ &+ \lambda_{\text{obj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}^{obj} ij (C_i - \hat{C}_i)^2 + \dots \end{aligned} \quad (3)$$

The Dataset used for this task was one publicly available on Robotflow which is an end-to-end computer vision platform that simplifies the process of building computer vision models.

B. Text Extraction and Recognition with EasyOCR

After the YOLOv8 model successfully detects text regions, we proceed to the text extraction and recognition phase. An open-source text recognition library called EasyOCR is essential in this situation:

The functionality of EasyOCR

EasyOCR was created especially to close the gap between text that can be read by machines and photos. It recognises the text content in an image that has text as input and outputs it.

Under the hood, EasyOCR uses deep learning models that have already been trained to recognise individual characters

in the image patch. These models are robust to variable text appearances since they can handle a wide range of font styles, sizes, and orientations. Connectivity to YOLOv8 Output:

Bounding boxes are provided by YOLOv8 around the image's recognised text sections. For EasyOCR, this information is essential. These bounding boxes are used by us to crop the particular image patches with text on them. In essence, we separate the text region of interest from the surrounding area so that EasyOCR may analyse it further. Procedure for Text Recognition:

Following receipt of the cropped picture patch, EasyOCR enters the data into its deep learning model. Pixel by pixel, the model deconstructs the image to identify individual characters, which it then puts together to form words and sentences.

Together with extra information, including confidence scores for each recognised character, EasyOCR produces the text content that has been recognised. These ratings show how confident the model is in its predictions, enabling future filtering or error correction.

Some of the advantages of EasyOCR are:

Efficiency: Real-time applications can benefit from EasyOCR's quick and lightweight processing, which is well-known. Maintaining a seamless pipeline workflow depends on this.

Multilingual Support: A large number of languages and character sets are supported by EasyOCR. The adaptability of this pipeline expands its applicability by enabling text extraction from photos that contain a variety of languages.

EasyOCR's user-friendly API facilitates seamless integration with other programmes, such as YOLOv8. This lowers the amount of coding work needed and streamlines the development process.

Essentially, EasyOCR serves as a conduit between an image's textual representation and its associated machine-readable format. This step successfully extracts the text from the created image by utilising the object detection capabilities of YOLOv8 and the text recognition expertise of EasyOCR.

This prepares the way for further mistake correction and refining.

The sequence labelling process of easyOCR occurs using the following equation:

$$h_t = \text{RNN}(h_{t-1}, x_t) \quad (4)$$

C. Text Error Correction with Gemini LLM

Having extracted the text content from the image using EasyOCR, we now turn to Gemini, a large language model (LLM), to address potential errors and inconsistencies. Gemini's capabilities elevate the pipeline by ensuring the extracted text adheres to proper language use and factual accuracy.

Gemini's Role:

Gemini acts as a multifaceted language processing tool within the pipeline. It analyzes the text extracted by EasyOCR, searching for various types of errors and inconsistencies: Spelling Errors: Gemini identifies misspelled words by comparing them to its vast vocabulary and suggesting potential corrections. This rectifies typos introduced during the image generation process. Grammatical Mistakes: Gemini assesses the text for grammatical errors, such as incorrect verb tenses, subject-verb agreement issues, or misplaced modifiers. By analyzing the sentence structure and context, it proposes corrections that enhance the overall clarity and fluency of the text. Factual Inconsistencies: Beyond basic grammar and spelling, Gemini can also identify factual inconsistencies within the extracted text. This is particularly valuable when dealing with text generated by AI models, which may not always adhere to real-world facts. Gemini leverages its knowledge base to compare the text with factual information and suggest corrections if discrepancies arise. Leveraging Gemini's Training Data:

Gemini's effectiveness hinges on its extensive training data. This data encompasses a massive corpus of text and code, allowing it to develop a comprehensive understanding of language use, grammar rules, and factual knowledge. During the text analysis process, Gemini draws upon this knowledge base to identify deviations from proper language use and suggest corrections that align with established norms and factual accuracy. Beyond Error Correction:

In addition to correcting errors, Gemini can also offer suggestions for stylistic improvements. It can analyze the text for clarity, conciseness, and overall tone, proposing refinements that enhance the quality and impact of the written content. Integration with the Pipeline:

Gemini receives the extracted text from EasyOCR and performs its analysis. The output from Gemini can take various forms, depending on the specific implementation. It can provide a corrected version of the text, highlight potential errors with suggestions, or offer confidence scores for its proposed changes. By incorporating Gemini's language processing capabilities, this stage refines the extracted text, ensuring it is not only free from errors but also adheres to proper grammatical structure, factual accuracy, and potentially stylistic considerations. This significantly improves the overall

quality and credibility of the text elements within the generated image.

IV. RESULTS

As our framework is pipelined, it must be noted that the evaluation of each stage or model in the pipeline is dependent on its preceding stages.

A. Evaluation of Text Detection

The YOLOv8 model trained on the Roboflow dataset demonstrated accuracy in detecting text from the supplied images. However, challenges arose particularly in scenarios with high variation in text sizes and highly stylized text. These limitations indicate the need for further improvement in the model's ability to handle diverse text characteristics. To address these challenges, training the model on the COCO-Text dataset is proposed. The COCO-Text dataset [6], a large-scale scene text dataset based on MSCOCO, contains images of complex everyday scenes. With 63,686 images, 145,859 text instances, and 3 fine-grained text attributes, the COCO-Text dataset provides a rich and diverse source of training data.

Here are some examples illustrating the performance of our model in Fig. 2.

B. Evaluation of OCR

The Optical Character Recognition (OCR) model was critical to our pipeline, particularly in terms of text extraction from photos. Using the YOLOv8 model's excellent bounding box detections, the OCR model extracted text from a variety of image contexts with impressive performance. Here, we provide a full review of the OCR component, demonstrating its usefulness with illustrated examples: **1. Accuracy in Text Extraction:** By correctly localising text regions based on YOLOv8 bounding box predictions, the OCR model was able to capture text with little error and distortion. **2. Robustness to Variability:** Exhibited robustness to variability in text styles, fonts, sizes, and orientations. **3. Handling Complex Layouts:** The OCR model demonstrated ability to handle complicated image layouts with many text sections and overlapping parts. **4. Error Analysis:** Although the general performance of the OCR model was very good, we found certain typical failure scenarios and areas that needed development. These included difficulties like irregular text layouts, poor contrast, text occlusions, and infrequent character recognition mistakes.

We have provided in Fig. 2 and Fig3. with qualitative examples showcasing the effectiveness of the OCR model in accurately extracting text from various image contexts.

C. Evaluation Of Text Correction

In our study, we performed a comparative analysis to assess the Gemini API's performance in correcting text generated by DALL-E against the original prompts provided. The prompts that were supplied included the exact text to be displayed on the image for the purpose of this evaluation. This analysis was performed in order to analyse the accuracy and efficacy of the Gemini model-facilitated text editing method. Here,

Given prompt	Generated Image	Text segmentation
"Poster recruiting astronauts"		
"create an advertisement for a car"		

Fig. 2. Table containing Given Prompt, Generated Image and the Image containing segmented Text Box

Text identified by EasyOCR	Corrected text using Gemini API	Similarity Score (TF-IDF)	Similarity Score (spaCy)
"Jon space exploration team"	"Join the space exploration team"	0.51	0.67
"The <u>feuri</u> dives row"	"The ferry drives now"	0.14	0.16

Fig. 3. Similarity scores of the Identified and Corrected text

we provide a full description of our findings. **1. Comparison between Prompt and Generated Text:** The texts prompts provided to DALL-E served as a baseline to measure accuracy of the corrected text that is generated. **2. Calculation of Similarity Scores:** We computed similarity ratings for the original text prompts and the modified text acquired from Gemini. This involved using similarity measurements including cosine similarity, Levenshtein distance, and semantic similarity measures.

As shown in Fig. 3., our findings showed that the Gemini model consistently produced high similarity scores between the original text prompts and the modified text, implying that it efficiently repaired errors, inaccuracies, and inconsistencies in the DALL-E-generated text.

V. CONCLUSION AND OUTLOOK

We presented a novel architecture in this paper that aims to tackle the important problem of text hallucinations in AI-generated graphics. By utilising state-of-the-art technologies,

our system blends OCR's precise text extraction with the YOLOv8 model's reliable text identification capabilities. We also included the Gemini API's robust text correction features, which led to a complete solution for reducing text hallucinations in images produced by the steady diffusion image generator DALL-E. Our work is important not only for academics but also for a wide range of enterprises, since text hallucinations in AI-generated graphics are a major problem. We seek to provide industries and stakeholders with the tools and processes required to navigate the changing world of AI-generated content by fusing innovation with usefulness.

VI. FUTURE SCOPE

Moving forward, our findings suggest various avenues for additional inquiry and improvement of the AI hallucination correction pipeline. One important part of our future effort is to broaden the breadth of our collection to incorporate more diverse and thorough sources, such as the Coco Text dataset [6]. By including a greater range of images and textual

prompts, we want to improve the robustness and generalisation capabilities of our text detection approach. Expanding our AI hallucination detection and correction system to include textual inpainting marks a big step forward in our research agenda. Textual inpainting is the process of replacing missing or incorrect text in photos with accurate and relevant text while maintaining the original image's style and arrangement. By including this component in our framework, we hope to provide a comprehensive solution for identifying, correcting, and improving AI-generated hallucinations in text-embedded images. Furthermore, we intend to examine the use of style transfer and adaptive layout generation techniques to improve the visual appeal and aesthetic quality of the inpainted text, assuring seamless integration with the overall image composition.

REFERENCES

- [1] R. Liu et al., "Character-aware models improve visual text rendering," in Proc. 61st Annu. Meet. Assoc. Comput. Linguist. (Volume 1: Long Papers), A. Rogers, J. Boyd-Graber, and N. Okazaki, Eds. Toronto, Canada: Association for Computational Linguistics, Jul. 2023, pp. 16270–16297. doi: 10.18653/v1/2023.acl-long.900.
- [2] J. Chen et al., "TextDiffuser-2: Unleashing the Power of Language Models for Text Rendering," arXiv:2311.16465 [cs.CV], 2023. [Online]. Available: <https://arxiv.org/abs/2311.16465>
- [3] J. Chen et al., "TextDiffuser: Diffusion Models as Text Painters," arXiv:2305.10855 [cs.CV], 2023. [Online]. Available: <https://arxiv.org/abs/2305.10855>
- [4] S. Lakhanpal et al., "Refining Text-to-Image Generation: Towards Accurate Training-Free Glyph-Enhanced Image Generation," arXiv:2403.16422 [cs.CV], 2024. [Online]. Available: <https://arxiv.org/abs/2403.16422>
- [5] YOLOv8. (Year). YOLOv8: A New State-of-the-Art Computer Vision Model. [2024]. Available: <https://yolov8.com/>
- [6] A. Veit et al., "COCO-Text: Dataset and Benchmark for Text Detection and Recognition in Natural Images," CoRR, vol. abs/1601.07140, 2016. [Online]. Available: <http://arxiv.org/abs/1601.07140>
- [7] H. I. Koo, B. S. Kim, Y. K. Baik and N. I. Cho, "Fast and simple text replacement algorithm for text-based augmented reality," 2016 Visual Communications and Image Processing (VCIP), Chengdu, China, 2016, pp. 1-4, doi: 10.1109/VCIP.2016.7805429.
- [8] Y. Zhang et al., "Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models," arXiv:2309.01219 [cs.CL], 2023. [Online]. Available: <https://arxiv.org/abs/2309.01219>
- [9] Hládek, D., Staš, J., Ondáš, S. et al. "Learning string distance with smoothing for OCR spelling correction". *Multimed Tools Appl* 76, 24549–24567 (2017). doi: 10.1007/s11042-016-4185-5
- [10] J. Wei, K. Chen, J. He, Z. Huang, Y. Lian and Y. Zhou, "A New Approach for Integrated Recognition and Correction of Texts from Images," 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, NSW, Australia, 2019, pp. 615-620, doi: 10.1109/ICDAR.2019.00104.
- [11] Y. Gao et al., "AIGCs Confuse AI Too: Investigating and Explaining Synthetic Image-induced Hallucinations in Large Vision-Language Models," arXiv:2403.08542 [cs.CV], 2024. [Online]. Available: <https://arxiv.org/abs/2403.08542>
- [12] Krishnan, Praveen, et al. "Textstylebrush: transfer of text aesthetics from a single example." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).
- [13] F. Zhan and S. Lu, "ESIR: End-to-end Scene Text Recognition via Iterative Image Rectification," arXiv:1812.05824 [cs.CV], 2019. [Online]. Available: <https://arxiv.org/abs/1812.05824>
- [14] Fu, Zhenxin, et al. "Style transfer in text: Exploration and evaluation" Proceedings of the AAAI conference on artificial intelligence. Vol. 32. No. 1. 2018.
- [15] Y. M. Y. Hasan and L. J. Karam, "Morphological text extraction from images," in *IEEE Transactions on Image Processing*, vol. 9, no. 11, pp. 1978-1983, Nov. 2000, doi: 10.1109/83.877220.
- [16] Koo, H. I., Kim, B. S., Baik, Y. K., & Cho, N. I. (2016). *Fast and simple text replacement algorithm for text-based augmented reality*. 2016 Visual Communications and Image Processing (VCIP). doi:10.1109/vcip.2016.7805429