Imperial College
London

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Automated Text Rectification in AI-Generated Visual Content

Author:
Prajit Sengupta

Supervisor:
Dr Thomas Lancaster

Submitted in partial fulfillment of the requirements for the MSc degree in MSc Advanced Computing of Imperial College London

February 2025

Abstract

In a world where images with text are everywhere—on social media, advertisements, scanned documents, and AI-generated content—there's a growing problem. Many of these images contain text that's misspelled, grammatically incorrect, or just doesn't make sense. These errors not only reduce the credibility of the content but also make it less effective and harder to use. What makes this issue even trickier is that simply fixing the text isn't enough. The corrected text needs to seamlessly match the original style—its font, color, size, and alignment—so it blends naturally into the image. Without this, the fix can stand out awkwardly, defeating the purpose.

This challenge is more than just correcting words; it's about preserving the harmony between text and visuals. By tackling this issue, we aim to improve the quality of digital content. This paper introduces **GenFix** which is the first open-source, fully automated model designed to detect, correct, and smoothly integrate text in images while maintaining style consistency. Unlike existing approaches that require manual intervention, GenFix combines vision-language models with inpainting techniques to achieve accurate, context-aware corrections. It not only ensures textual accuracy, with a perfect Word Error Rate (WER) and Character Error Rate (CER) of 1, but also preserves the visual integrity of the image, achieving an impressive Structural Similarity Index (SSIM) of 0.9555, thereby ensuring high structural preservation.

To support the evaluation of this problem of incorrect text in images, we introduced **TextSynth-100**, a benchmark dataset of AI-generated image-text pairs specifically designed to work with text correction models. Whether it's for creating accessible materials, improving AI-generated media, or improving everyday content, solving this problem has far-reaching potential. This work is about bridging the gap between what we see and what we understand, by making text in images both accurate and visually perfect.

Acknowledgments

Firstly, I would like to sincerely thank Dr. Thomas Lancaster for his invaluable guidance and support throughout this project. From the very beginning, he introduced me to the fascinating challenge of correcting text in images while preserving style consistency—a problem I might never have considered on my own. His innovative perspective and thoughtful feedback helped me shape this work into something meaningful. Dr. Lancaster's encouragement to explore new ideas and his ability to ask the right questions have been truly inspiring and instrumental in my learning journey.

Finally, I also want to extend my gratitude to the Department of Computing for providing me this opportunity to work on this research project as part of my Individual Study Option (ISO). Having the platform to dive into such an exciting and complex area of research has been an incredible experience, and I deeply appreciate the support and resources that made this possible. This project has been a rewarding journey, and I'm grateful to everyone who contributed to making it happen.

Contents

1 **Introduction** **1**
- 1.1 Research Motivation 1
- 1.2 Research Approach and Outline 3
- 1.3 Image Generation Methods 4

2 **Background and Related Work** **7**
- 2.1 Challenges in Image-Text Generation by AI-Models 7
- 2.2 OCR and Image Text Extraction 8
- 2.3 Text Rendering and Correction 9
- 2.4 Image Inpainting and Style Preservation 10
- 2.5 Evaluation Metrics 11

3 **Design and Implementation** **12**
- 3.1 Proposed Pipeline 12
  - 3.1.1 OCR-Based Text Detection 12
  - 3.1.2 Caption Generation for Contextual Understanding 12
  - 3.1.3 Graph-Based Text Alignment 13
  - 3.1.4 Energy-Based Geometric Optimization 14
  - 3.1.5 Text Correction with Vision-Language Models 15
  - 3.1.6 Mask Generation for Inpainting 15
  - 3.1.7 Text Inpainting Using Stable Diffusion 15
  - 3.1.8 Final Output 15

4 **Experimental Results** **16**
- 4.1 Dataset 16
- 4.2 Experimentation on different Datasets 16
  - 4.2.1 Images with spelling errors 17
  - 4.2.2 Images with contextually incorrect text 17
  - 4.2.3 Scene-Aware Text Generation 18
- 4.3 Model Evaluation 18
  - 4.3.1 Text Accuracy Metrics 18
  - 4.3.2 Image Quality Metrics 18
  - 4.3.3 Discussion 19

5 **Conclusion and Future Work** **20**

6 **Declaration** **27**

Appendix **28**

Chapter 1

Introduction

Nowadays AI can be easily seen acing competitive exams, writing code for company workflows, or defeating e-sports world champions. For the last decade, Generative AI applications have been experiencing rapid growth. Applications like ChatGPT (1), SORA (2), Dall-E (3) and Emu (4) have been making headlines for generating innovative text to images and video content. However, while these models do demonstrate great capabilities through their creativity and precision in generating varied visual and textual content, they do often fail when it comes to generating accurate meaningful text within images. (5)

Most of the times, the text generated within these images appears to be gibberish or some nonsensical sequence of characters. Such limitation arises due to the challenges of integrating LLMs (Large Language Models) with Image Generation Models, particularly in capturing the precise details of characters within an image (6). Correcting such flawed text in generated images is essential for usability purposes and for the adoption of AI-generated visual content across various industries.

This problem has a significant impact in domains like advertising media, digital art, articles and automated content creation, where text within images often plays a crucial role in delivering clear and meaningful messages. Addressing this problem would require a multidisciplinary approach, combining concepts from computer vision, natural language processing, and certain AI techniques which we will discuss later in detail.

1.1 Research Motivation

Though AI has made such great strides in most of the fields, yet it faces difficulty in generating meaningful words and sentences in AI Generated Images. When tasked with generating a highway billboard for a specific city, AI models often produce place names that do not exist or contain spelling errors. Additionally, AI-generated images may include blurred or gibberish text which does not fall under any particular language known to humans. (5)

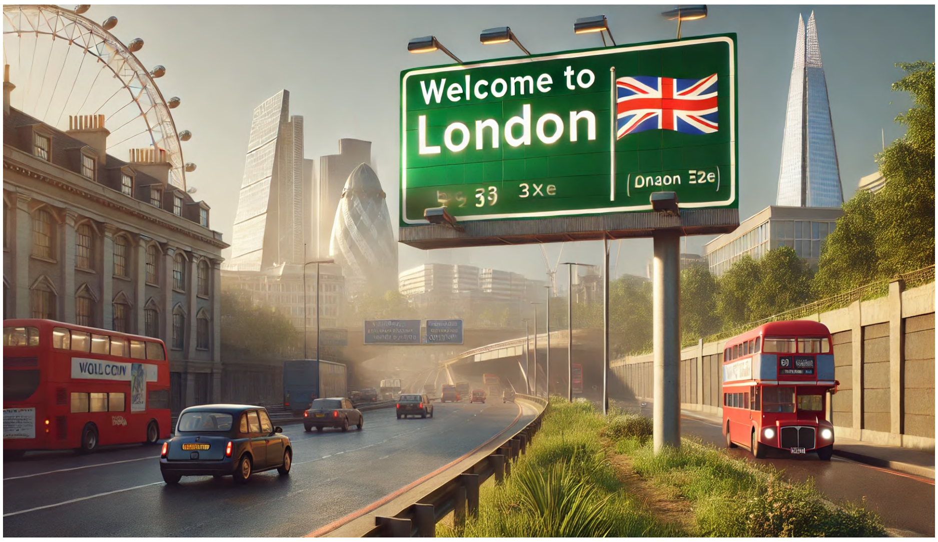

Figure 1.1: Image Generated from Dall-E3 (OpenAI) (1)

Reasons Generative AI Models struggle to generate perfect text within images:

- Currently Generative AI model's primary focus is on Visual Elements rather than Textual content. Their focus on visual elements makes it challenging to prioritize and generate fine details such as individual text characters.
- Language is inherently complex, with context playing a critical role. The generative AI models need to understand not just the words but also their placement within an image, which makes it extremely difficult for the Image Generation Models.
- There is limited training data for Text-in-Images. This limits the model's ability to learn the fine details of the texts within an image. This limitation becomes particularly evident when generating images based on specific topics, such as those related to complex subjects like nuclear power plants. (7)

This problem was also tested in real-time in different Generative AI Models using GANs or Diffusion Models as the core technology. The prompt *"Create an image of a highway board of the city of London."* was provided to all the models. The outputs received are shown in the following Figures 1.1, 1.2, and 1.3. In the generated image from Dall-E3 (By OpenAI) as shown in Figure 1.1, it's worth noting that apart from the given word "London", it produces gibberish and blurred text around the bottom. The Emu Model (By MetaAI) which uses the Llama-3 which is based on diffusion models, also produces nonsensical text with names of places which does not really exist as shown in Figure 1.2. Even the image generated from Copilot (By Microsoft) was mostly incorrect, as in Figure 1.3. Here the model tries hard to spell "Highway" correctly but it incorrectly spells it as "Highday".

There are certain softwares like Ideogram and Stockimg.ai, which do produce better results than Dall-E and other modern diffusion models, but still fails in complicated text scenarios.

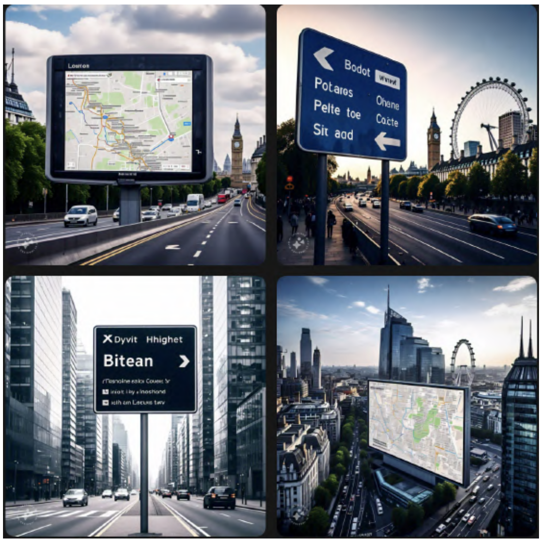

Figure 1.2: Image Generated from Emu (MetaAI) (4)

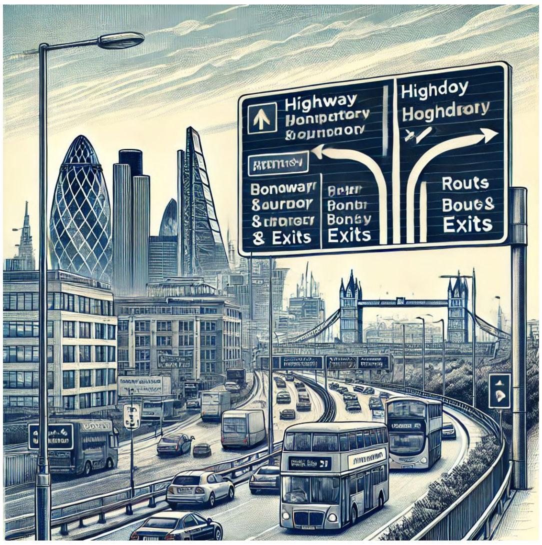

Figure 1.3: Image Generated from Copilot (Microsoft) (8)

From the perspective of the user point of view, the jumbled and incoherent text reduces the visual appeal, but the mismatch between different fonts and styles further complicates things, making the visual text unprofessional as well as distractive. What adds to this problem is the sheer amount of time it takes to correct these visuals manually. The ability of the generative models in producing accurate and coherent text within images is essential given their widespread use. So solving this challenge requires focusing on such intricacies like eliminating gibberish text, ensuring consistent fonts and styles, and automating corrections.

1.2 Research Approach and Outline

Our approach focuses on exploring how emerging models and techniques can be combined to tackle textual errors, semantic misalignment, style inconsistencies and other distortions (5). This paper seeks to advance the capabilities of automated image text correction by implementing a structured pipeline as well as exploring its different components.

To identify, correct, and integrate corrected text in GenAI-generated images, we propose a model named **GenFix**. This model employs a combination of OCR (TrOCR (9), EasyOCR (10)), vision-language models (BLIP) for context (11), diffusion-based inpainting (Stable Diffusion Inpainting)(12), and style preservation techniques. The model also utilizes a graph-based text alignment algorithm and energy optimization to enhance accuracy in text placement as discussed in Chapter 3. A dataset named **TextSynth-100** consisting of images with textual content generated by AI has also been proposed to evaluate text correction models on some fixed dataset benchmark. The source code and dataset are provided in https://github.com/Prajit-Sengupta/Correcting-Text-In-Images-Using-AI.

1.3 Image Generation Methods

Generating Images using AI involves various deep learning models and probabilistic algorithms. Currently, there are several models that are quite prevalent in the market, which creates realistic and novel images. However, these models face multiple challenges, as discussed in Chapter 2 (Section 2.1).

1. **Variational Autoencoders (VAEs)**

   VAEs are a type of deep learning model designed to generate new data that closely relates to a given dataset. These models add a taste of probabilistic twist to traditional autoencoders, making them more powerful for generating diverse and novel images. (13)

   The model has two core components as shown in Figure 1.4:

   - **Encoder:** It converts an image into a compact latent space representation (a compressed representation which captures the most important features) by mapping input data to a lower dimension.
   - **Decoder:** It takes that compressed information and reconstructs the original image or a brand new image, never-before-seen.

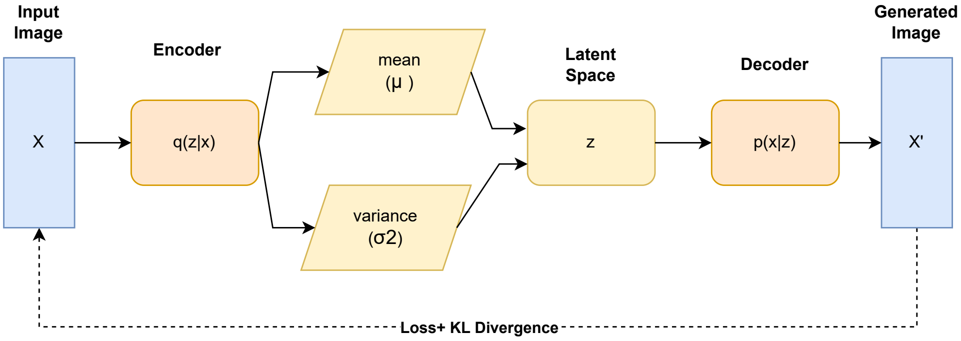

Figure 1.4: Variational Autoencoder Architecture

   At their core, VAEs rely on Bayesian probability and optimization techniques to learn efficient data representations.

$$\mathcal{L} = E_{q(z|x)} [\log p(x|z)] - D_{KL} (q(z|x) \parallel p(z)) \tag{1.1}$$

   where:

   - $E_{q(z|x)} [\log p(x|z)]$ ensures that the generated data resembles the real data.
   - $D_{KL} (q(z|x) \parallel p(z))$ is the Kullback-Leibler (KL) divergence.

2. **Autoregressive Models**

   Autoregressive (AR) models for image generation creates images **pixel-by-pixel** or **patch-by-patch**, by modeling the probability of each pixel (or patch) based on previously generated ones.

Autoregressive models treat an image as a sequence of pixels rather than generating the entire image at once. The probability of generating an image from the pixels is as follows:

$$P(X) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2) \dots P(x_N|x_1, x_2, ..., x_{N-1}) \tag{1.2}$$

- $X = (x_1, x_2, ..., x_N)$ represents the entire image.
- x_i is an individual pixel (or patch).

3. **Generative Adversarial Network (GANs)**

GANs are one of the most powerful and popular approaches introduced by Ian Goodfellow et al (14). It competes two AI models against each other, which gives the term ”Adversarial” to the name. As shown in Figure 1.5, it is an unsupervised learning method, consisting of two sub-models:

- **Generator:** Generates image samples (fake) similar to the training data.
- **Discriminator:** Classifies if the image is generated or a real sample from the domain.

The core idea is that the generator goes through multiple cycles of creating samples and improving itself until it can produce something so realistic that it fools the discriminator, ultimately minimizing the discriminator’s loss. Once training is complete, the discriminator is no longer needed during inference.

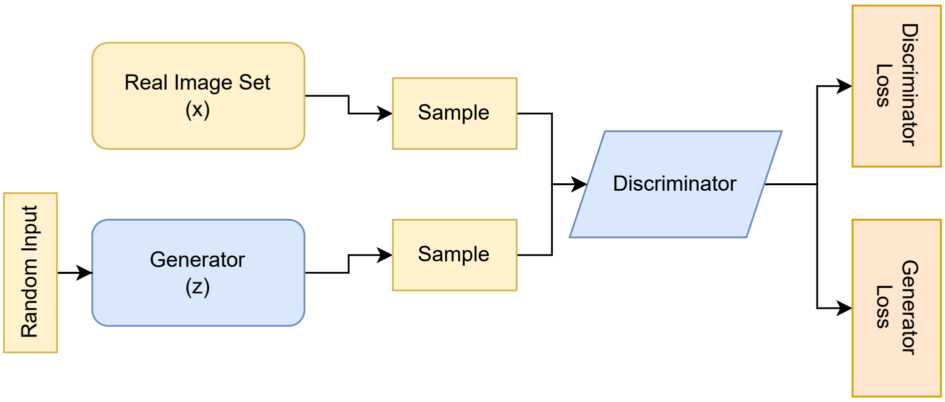

Figure 1.5: Generative Adversarial Network Architecture

The training objective of GANs is formulated as a **minimax game** between the generator G and the discriminator D:

$$\min_G \max_D V(D, G) = E_{x \sim p_{\text{data}}(x)}[\log D(x)] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{1.3}$$

- $p_z(z)$ is the prior distribution of latent variable z (e.g., Gaussian noise).
- $G(z)$ generates a fake sample from z.
- $D(x)$ is the probability that x is a real image.

4. **Diffusion Models**

   Diffusion Models have emerged as one of the most powerful techniques for image generation. Models like DALL-E 2 (15), Stable Diffusion (12) and Imagen (16) use Diffusion Models as their core technology. These models work by gradually adding noise to an image and then learning to reverse the process (denoising) as shown in Figure 1.6.

   - **Forward Diffusion (Adding Noise):** At each time step we add a little bit of noise to the image like a Markov Chain, until the image only consists of noise. (From x0 to x4)

   $$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \tag{1.4}$$

   - **Reverse Diffusion (Denoising):** The core idea of diffusion models is to train a neural network to undo this noise and recover the original image which is done through denoising

   $$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_\theta^2 I) \tag{1.5}$$

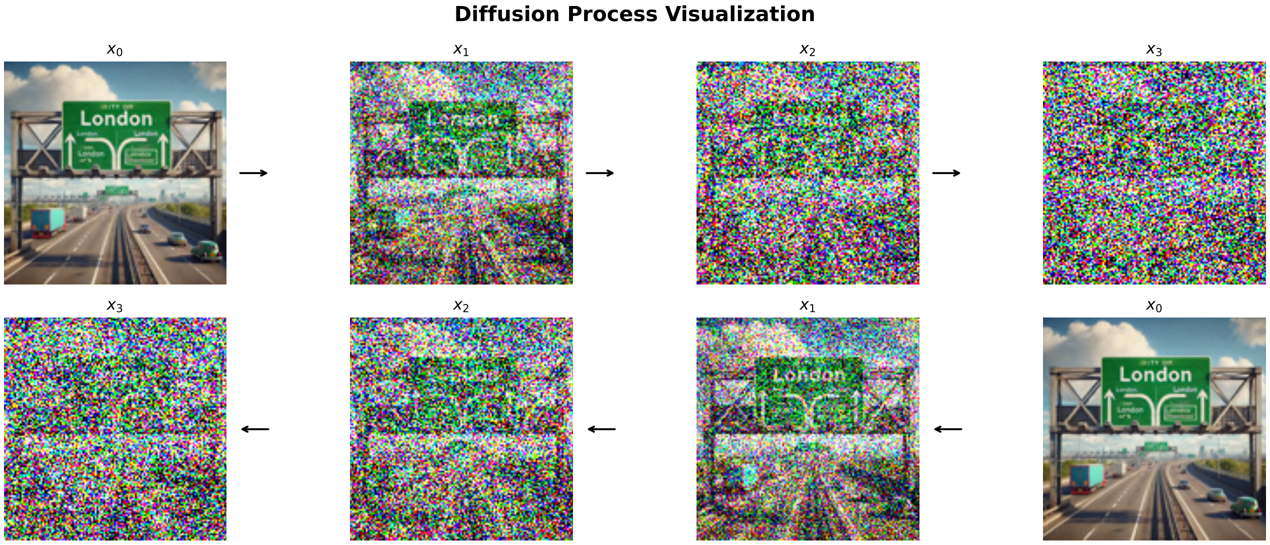

Figure 1.6: Diffusion Process (Forward Process + Reverse Process)

5. **Transformer-Based Models**

   Models like OpenAI's DALL·E 2 and Stability AI's Stable Diffusion use transformers and attention mechanisms to generate images from text descriptions.

   The transformer architecture enables the model to capture the relationships between words in the prompt, while the attention mechanism allows it to focus on different aspects of the text to create coherent and visually appealing images. Notably, DALL·E 2, uses a combination of a transformer and a diffusion model to improve image-text synthesis quality.

Chapter 2

Background and Related Work

Effective text rectification in images requires a multidisciplinary approach, combining text detection, correction, and smooth reintegration. To contextualize our work, we explore challenges in AI-generated text, (17; 18) review multidisciplinary techniques, and examine evaluation metrics for both textual accuracy and image quality.

2.1 Challenges in Image-Text Generation by AI-Models

Amara et al. (17) explores the challenges of making AI-generated text more explainable, identifying 17 key challenges across three areas: dataset creation, explanation design, and evaluation. One major issue is that AI models generate text using probabilistic methods like top-k sampling, meaning their word choices can be unpredictable and difficult to explain. Another major challenge with text-to-image synthesis as discussed by Chen (19) is semantic alignment, where generated images sometimes fail to accurately represent the given text prompt—an issue similar to how LLMs sometimes produce hallucinations or incoherent responses.

Building on these challenges, Zhu et al. (18) addresses the problem of restoring corrupted text images through their Global Structure-guided Diffusion Model (GSDM), which highlights another layer of complexity in AI-generated content. The authors introduce text image inpainting as a critical task, focusing on how environmental damage or human interference, like graffiti or incomplete signatures, can disrupt the readability and integrity of text. Expanding this discussion into scientific applications, Joynt et al.(20) conducted a comparative analysis of 20 text-to-image generative AI models, evaluating their ability to produce scientifically accurate images related to nuclear energy. While models like DALL-E, DreamStudio, and Craiyon performed reasonably well with general prompts, they struggled to depict technical details accurately, often defaulting to stereotypical imagery such as cooling towers.

Beyond these technical challenges, Ivezić and Bagić Babac (21) underscore the increasing importance of addressing the ethical, legal, and sustainability aspects of text-to-image generation. Although advancements in diffusion models and GANs have markedly enhanced image quality, persistent issues such as semantic misalignment, the reinforcement of harmful stereotypes, and the creation of misleading or copyrighted content continue to pose significant obstacles. Several of these key challenges in AI-generated text within images are summarized in Table 2.1, including issues like semantic alignment, style consistency, and structural integrity.

Category	**Description**	**Example**	**Possible Solutions**
Semantic Alignment	Generated text/images fail to match input prompts	Text prompt: *”Red apple”*, image shows an orange fruit	Improved text-image embedding, multimodal transformers (19; 22)
Style Consistency	Difficulty maintaining uniform style across generated content	Inconsistent font styles in restored handwritten documents	Global structure-guided models, style transfer techniques (18; 23)
Structural Integrity	Loss of shape or structure in text within images	Broken letter strokes in handwritten text restoration	Diffusion models with structural priors (24; 25)
Content Ambiguity	Ambiguous or unclear content due to incomplete data	Corroded text resembling different words (*“office”* vs. *“offico”*)	Incorporating context-aware correction models (26; 27)
Dataset Limitations	Lack of diverse and real-world datasets	Overfitting to synthetic data, failing in real-world text	Creation of comprehensive datasets (eg COCO Text Datset, DiffusionDB) (28; 29)

Table 2.1: Challenges in Text Generation in Images by AI Models

2.2 OCR and Image Text Extraction

OCR and Text extraction is an integral starting point of the proposed pipeline. Lot of research has been done in this field to identify text from different image scenarios.

The foundational research of OCR with AI began through the research work of LeCun et al (30) in 1998, which introduces CNNs (Convolutional Neural Network) for recognizing handwritten characters, paving the path for Neural Network based OCR architectures (CNN named LeNet-5 was used as the core technology in this paper). This paper also recognizes the advantage of training a recognizer to process entire words, instead of relying on pre-segmented individual characters using the concept of GTN (Graph Transformer Network). With this research in place, they had to face several issues, one of which is its reliance on predefined lexicons, limiting its ability to recognize out-of-vocabulary words.

In 2007 Smith (31) from Google came up with the Tesseract OCR Engine, which managed to overcome lot of the research gaps. The Tesseract could easily handle white-on-black text due to it's connected component analysis approach introduced in this paper. Also the adaptive classifier concept introduced here improves recognition accuracy greatly as it uses a different normalization technique compared to a static classifier by distinguishing upper and lower case characters as well as improves immunity to noise, retaining font differences. Such papers contributed to text detection and recognition research, as shown in Table 2.2, which compares performance.

Method	Dataset	Metric	Score	Remarks
Tesseract v4(31)	ICDAR 2015	CER	12.5	Struggles with noisy images
CRNN(32)	COCO-Text	CER	8.7	Handles curved text well
EAST(33)	ICDAR 2015	F1-Score	82.1%	Effective for scene text detection
TrOCR (9)	Synthetic + IAM	CER	4.22	Lightweight and Effective
PaddleOCR(34)	SynthText	CER	7.5	Lightweight and versatile
Google Vision OCR(35)	ICDAR 2015	WER	5.3	Best for multilingual scenarios

Table 2.2: Performance Comparison of OCR Methods on Benchmarks

With the hype of OCR technology, researchers also came up with the concept of Scene Text Recognition (STR) which detects and recognizes text from natural scenes, just like reading text with AI algorithms in real world scenario. Shi et al. (32) came up with a very interesting concept of sequence recognition in natural daily scenes. Unlike most objects in the real world, such as cars or buildings, which appear in a disordered manner, text in natural scenes (e.g, scene text) typically occurs in a sequential format. This makes it possible to predict a series of object labels as a sequence, enabling more effective recognition of scene text. To recognize these sequences they proposed a neural network named Convolutional Recurrent Neural Network (CRNN), which combines the concepts of RNN and DCNN.

Zhou et al (33) tried improving the scene text by developing EAST (An Efficient and Accurate Scene Text Detector) which directly predicts text lines of different orientations and shapes with a single neural network (FCN Model and NMS merging) unlike previous approaches. Due to it's simplification EAST was faster, but to achieve higher accuracy Baek et al. (36) came up with CRAFT (Character Region Awareness for Text Detection), which localizes individual characters and links to form text instances enabling better handling of different text shapes and sizes, unlike other methods (32; 33) which mainly train their deep-learning networks to localize word level bounding boxes over the image scene. This way it can handle way more challenging texts in real world scenarios.

2.3 Text Rendering and Correction

Visual Content like image-with-text generation have come a long way from GAN (Generative Adversarial Networks) based models to Diffusion Models, which is currently achieving state-of-the-art results compared to the previous one. Recent advancements in Diffusion Models have demonstrated models like DeepFloyd (37), Imagen (16), Swinv2-Imagen (38) or SeqDiffuSeq (39) which incorporates T5 series text encoders (40) significantly improves text generation capabilities compared to models utilizing the CLIP text encoder. While models like CLIP (Contrastive Language-Image Pretraining) (41) introduced by OpenAI excel at aligning text with visual content using contrastive learning, they still struggle to generate precise, coherent text within images, leading to errors in character rendering. To improve this, the T5 series text encoders, developed by Raffel et al. (40), which have been integrated into several diffusion models (37; 16; 38; 39) are used. These encoders provide a unified text-to-text framework. They convert all NLP tasks into a text generation problem, allowing models to generalize better across tasks. When integrated into diffusion models, T5 encoders enhance text generation fidelity, ensuring that the semantic and syntactic integrity of the text is maintained.

Despite the remarkable achievements, these methods still struggle to render accurate text. The text extracted from an image using OCR often might contain errors such as incorrect words, grammatical issues, or structural inconsistencies which might have been generated due to the limitations of Generative AI Models. Such textual content requires post-processing using text-correction techniques. Chen et al. (42) tried to address this by introducing TextDiffuser made by combining a Transformer and Diffusion Models, which mainly focuses on generating text coherent with the background. Some of the popular text-to-image models lack character-level input feature, which makes it more challenging to accurately predict the visual composition, as noted by Liu et al (43). They conducted series of experiments comparing character-blind and character-aware text encoders, proving how the latter outperforms their counterparts across various text rendering tasks.

To address this challenge, BLIP (Bootstrapping Language-Image Pretraining) (11) provides a context-aware text correction framework by jointly processing visual and textual information. It leverages a transformer-based architecture to generate captions that align with the visual context, providing semantic coherence between text and image. By utilizing noisy web data, BLIP generates high-quality captions and refines extracted text accordingly.

Advanced models like DiffUTE (44), Type-R (45) and GPT-k for Efficient Editing (46) enables accurate modifications to text within images while preserving the context. These methods particularly investigates the use of LLMs, which improves prompt editing and lowers typographic errors.

2.4 Image Inpainting and Style Preservation

Image Inpainting is a technique used to edit images by reconstructing missing or damaged parts of the image. Mainly it is an algorithm for editing textual information present in images in a convenient way similar to the conventional text editors. Earlier, researchers focused on using basic image properties like structure and texture to fill in missing areas. Some models were based on different geometrical features (47; 48; 49), which were not able to generalize well to the wide variety of textures and fonts. Unlike other methods Roy et al.(50), introduced STEFANN, which uses the FANnet (Font Adaptive Neural Network) in which a new character matches the original font style, even when only one character is available as a reference. Also, the Colornet network handles visual style preservation by transferring color attributes from the original character to the new one. In addition, the seam carving technique is used as part of the inpainting process to adjust the inter-character spacing, which ensures that the new text fits naturally within the scene

Recently, various deep learning architectures have been applied to image inpainting, including GANs (14), VAEs (13), and Transformers. The breakthrough in this field came as diffusion models emerged as a game changer. Our work focuses on text-conditional image inpainting using diffusion models, a specific area where the inpainting process is guided by text-based instructions. In particular, we employ *Stable Diffusion Inpainting*, which extends latent diffusion models (LDMs) (51).

2.5 Evaluation Metrics

Text-to-image generation models have seen a rapid growth last few years, which makes it increasingly important for the developers to develop robust evaluation methods to assess the visual as well as textual context within the image. There are several metrics to assess the quality of the generated images, such as CLIP (Contrastive Language–Image Pretraining, GIQA (Generative Image Quality Assessment), FID (Fréchet Inception Distance) and T-IQA (Text Image Quality Assessment).

- **CLIP (Contrastive Language–Image Pretraining)**: Measures the alignment between a generated image and a text prompt through contrastive learning. It employs a dual-encoder architecture (image and text encoders).
- **GIQA (Generative Image Quality Assessment)**: Assesses the perceptual quality of AI-generated images, focusing on sharpness, artifacts, and distortions.
- **FID (Fréchet Inception Distance)**: Measures the realism of generated images by comparing their statistical distribution to real images. It computes the Fréchet distance between feature distributions extracted from an Inception-v3 model.
- **T-IQA (Text Image Quality Assessment)**: Evaluates the quality of text in generated images, considering readability, clarity, and distortions. It employs deep learning models trained on labeled datasets to score text sharpness.

Wang et al (52) went to the extent of evaluating combinational creativity for generative images using coincident rate (CR) and average rank variation (ARV). Among all the evaluation metrics, GIQA performed the best to validate the images containing combinational creativity.

Chapter 3

Design and Implementation

Automated text correction in images, particularly those generated by generative AI models, poses significant challenges due to distortions, style variations, and misalignments. This paper presents a novel pipeline model named **GenFix** which integrates TrOCR (9) and EasyOCR (10) for OCR, BLIP-2 (11) for context-aware text correction, and Stable Diffusion (12) for style-preserving inpainting as represented in Figure 3.1. Additionally, we introduce a graph-based text alignment mechanism utilizing the Hungarian algorithm and an energy-based optimization function to improve text structure and readability.

3.1 Proposed Pipeline

The proposed method as detailed in Algorithm 1 performs text correction in AI-generated images by leveraging OCR, vision-language models, graph based alignment and diffusion-based inpainting.

3.1.1 OCR-Based Text Detection

Given an input image I, text regions $\mathcal{R}$ are detected using TrOCR (9) and EasyOCR (10), a Transformer-based OCR model. The detected text T is extracted from each region:

$$T_i = \text{OCR}(I, r_i), \quad r_i \in \mathcal{R}.$$

The extracted text is used for subsequent text correction and alignment.

3.1.2 Caption Generation for Contextual Understanding

To provide contextual information, a caption C is generated using BLIP-2, a vision-language model:

$$C = \text{BLIP-2}(I).$$

This caption aids in understanding the intended meaning of the extracted text within the image, helping the correction module make contextually appropriate modifications.

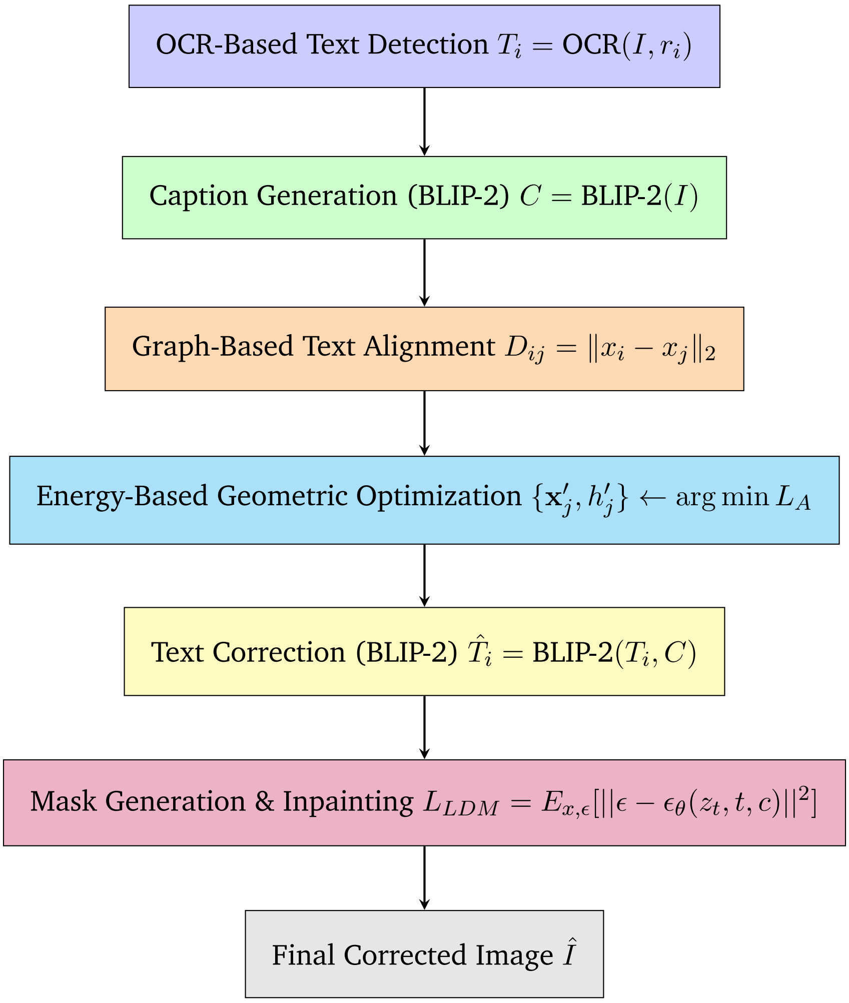
OCR-Based Text Detection $T_i = \text{OCR}(I, r_i)$
Caption Generation (BLIP-2) $C = \text{BLIP-2}(I)$
Graph-Based Text Alignment $D_{ij} = \|x_i - x_j\|_2$
Energy-Based Geometric Optimization $\{\mathbf{x}'_j, h'_j\} \leftarrow \arg\min L_A$
Text Correction (BLIP-2) $\hat{T}_i = \text{BLIP-2}(T_i, C)$
Mask Generation & Inpainting $L_{LDM} = E_{x,\epsilon}[||\epsilon - \epsilon_\theta(z_t, t, c)||^2]$
Final Corrected Image $\hat{I}$

Figure 3.1: Pipeline for Text Correction in AI-Generated Images

3.1.3 Graph-Based Text Alignment

This step resolves semantic misalignment between detected text regions and contextually corrected words. Generative images often place text in non-linear layouts (e.g., curved or scattered), causing OCR outputs to mismatch the corrected sequence from BLIP-2. We frame this as a bipartite graph matching problem:

- **Problem**: Detected text regions $\mathcal{T} = \{T_1, \ldots, T_N\}$ may not correspond to the corrected words $\mathcal{S} = \{s_1, \ldots, s_M\}$ in order.

- **Solution**: A cost matrix D is constructed where D_{ij} measures the edit distance between T_i and s_j. The Hungarian algorithm finds the optimal assignment σ^*, ensuring each T_i maps to the most semantically similar s_j. [Theorem 1] (See Appendix D)

This alignment preserves the intended reading order and contextual relevance of the corrected text.

Algorithm 1 Text Correction in AI-Generated Images with Graph-Based Alignment

```
Require: Input image $I$
Ensure: Corrected image $\hat{I}$ with refined text
Initialize: Load TrOCR for OCR, BLIP-2 for text correction, and Stable Diffusion for inpainting
Convert $I$ to RGB format
Extract text regions $\mathcal{R}$ using TrOCR
for each text region $r_i \in \mathcal{R}$ do
    Extract text $T_i \leftarrow OCR(I, r_i)$                ▷ Apply OCR on detected region
end for
Generate image caption $C \leftarrow BLIP(I)$                ▷ Contextual understanding
Graph-Based Text Alignment
Generate corrected candidates $\mathcal{S} \leftarrow \{BLIP(T_i, C)\ \forall T_i \in \mathcal{T}\}$
Construct cost matrix $D$ where $D_{ij} = \text{EditDistance}(T_i, s_j)$
Solve optimal assignment $\sigma^* \leftarrow \arg\min_\sigma \sum_{i=1}^{N} D_{i,\sigma(i)}$
Reorder regions $\mathcal{R}' \leftarrow \{r_{\sigma^{-1}(j)}\}_{j=1}^{N}$
Energy-Based Geometric Optimization
Initialize parameters $\lambda_1 = 0.5, \lambda_2 = 0.3, \mu = 0.1, \nu = 0.1$
Optimize layout: $\{\mathbf{x}'_j, h'_j\} \leftarrow \arg\min L_A$
for each aligned region $r'_j \in \mathcal{R}'$ do
    Original index: $i \leftarrow \sigma^{-1}(j)$
    if $\hat{T}_j = T_i$ then
        continue                                             ▷ Skip inpainting if no correction
    end if
    Create mask $M_j$ for optimized region $r'_j$            ▷ Use $\mathbf{x}'_j, h'_j$
    Resize $M_j$ to 512×512
    Generate $\hat{I}_j \leftarrow SD(I, M_j, \hat{T}_j)$    ▷ Style-preserving inpainting
    Resize $\hat{I}_j$ to original dimensions
    Blend $\hat{I}_j$ into $I$ using $M_j$
end for
Return corrected image $\hat{I}$
```

3.1.4 Energy-Based Geometric Optimization

Post-alignment, text regions may still be spatially inconsistent (e.g., uneven spacing or sizing). We formulate an energy function to refine geometry:

- **Problem**: Corrected text regions might overlap or disrupt the image's visual flow.

- **Solution**: Minimize an energy L_A combining:

$$L_A = \sum_{i=1}^{N} \left[\lambda_1 \|\mathbf{x}'_i - \mathbf{x}_i\|^2 + \lambda_2 (h'_i - h_i)^2\right] + \sum_{i=2}^{N} \left[\mu \|\mathbf{x}'_i - \mathbf{x}'_{i-1} - [d, 0]\|^2 + \nu (h'_i - h'_{i-1})^2\right]$$

  - *Fidelity*: Penalizes deviations from original positions (λ_1) and heights (λ_2).

- *Spacing*: Enforces horizontal alignment with gap d (μ).
- *Uniformity*: Ensures consistent text height (ν).

The convexity of L_A guarantees a unique solution, producing a better visual layout while respecting detection priors. [Theorem 2, Theorem 3] (See Appendix D)

3.1.5 Text Correction with Vision-Language Models

Each extracted text T_i is corrected using BLIP-2, conditioned on the caption C:

$$\hat{T}_i = \text{BLIP-2}(T_i, C).$$

The correction process optimizes the sequence-to-sequence objective:

$$P(\hat{T}|T, C) = \prod_{t=1}^{T} P(\hat{T}_t|\hat{T}_{<t}, T, C).$$

If $\hat{T}_i = T_i$, no correction is needed, and the original text remains unchanged.

3.1.6 Mask Generation for Inpainting

A binary mask M is created for regions requiring correction. The bounding box coordinates of the corrected text are used to define the inpainting area:

$$M_i = \text{Mask}(r_i).$$

The mask ensures that only the erroneous text regions are replaced, preserving the surrounding image context.

3.1.7 Text Inpainting Using Stable Diffusion

Stable Diffusion is used to replace erroneous text with corrected text in a visually consistent manner. The inpainting process follows a latent diffusion model objective:

$$L_{LDM} = E_{x,\epsilon\sim\mathcal{N}(0,1)}\left[||\epsilon - \epsilon_\theta(z_t, t, c)||^2\right],$$

where z_t is the latent representation, c is the conditioning input (including the mask and corrected text), and ϵ represents noise sampled from a normal distribution. The inpainting result $\hat{I}_i$ is blended into the original image to obtain the final corrected image $\hat{I}$.

3.1.8 Final Output

The final corrected image $\hat{I}$ retains the original structure while ensuring that text is accurately recognized, aligned, and visually similar. The integration of graph-based alignment and energy-based optimization improves text structure, enhancing both readability and aesthetics of the image-text pair.

Chapter 4

Experimental Results

4.1 Dataset

TextSynth-100 is a dataset specifically designed to evaluate the capability of generative AI models in creating images with embedded text as presented in Figure 6.1 (See Appendix B). Unlike traditional synthetic image datasets, which primarily focus on objects, landscapes, or artistic compositions, TextSynth-100 is uniquely centered around **textual content** in images. The dataset consists of 100 image-text pairs, equally distributed across four generative AI models: DALL·E (ChatGPT) (3), Copilot (8), Imagen (Gemini) (16), and Emu (Meta) (4). To ensure diversity, TextSynth-100 was created using varied text prompts, as illustrated in Figure 6.1 (See Appendix B). These include requests for street signs, movie screens, busy cityscapes with digital billboards, café menus, and laptop screens displaying slogans. The dataset is designed to test models in different real-world text-rendering scenarios just like the MARIO-10M (42) dataset. The TextSynth dataset is publicly available in the `TextSynth_dataset` zipped folder for further use.

TextSynth-100 serves as a benchmark for evaluating AI-generated text in images and can support further research in generative model improvement, OCR error correction, and multimodal AI training. To strengthen the generalizability of the GenFix model, the dataset was expanded by including real-world photographs sourced from the internet. This addition aimed to assess the model's adaptability in handling images containing incorrect text, extending beyond the synthetic examples in the original dataset. These real-world images came in various sizes and shapes, representing the wide range of inputs that GenFix is built to handle, highlighting its ability to work with images of different resolutions and aspect ratios.

4.2 Experimentation on different Datasets

The model was evaluated across various categories of errors in AI-generated image-text pairs. It demonstrated strong capabilities in correcting spelling mistakes, as illustrated in Figure 4.1. However, challenges remain, particularly in cases where the model hallucinates corrections, as seen in Figure 4.5.

4.2.1 Images with spelling errors

Spelling Correction: The original image Fig 4.1 had the misspelling ”SOTP” which was corrected to ”STOP” in the corrected image Fig 4.2. Additionally, the contrast of the text font was also improved due to the presence of *Stable Diffusion* component in the pipeline, specifically during the inpainting stage.

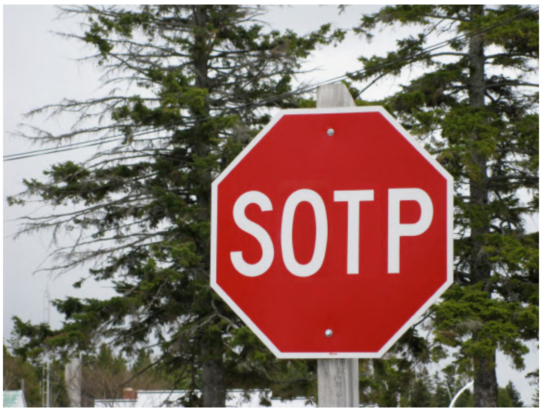

Figure 4.1: Image Generated from GenAI

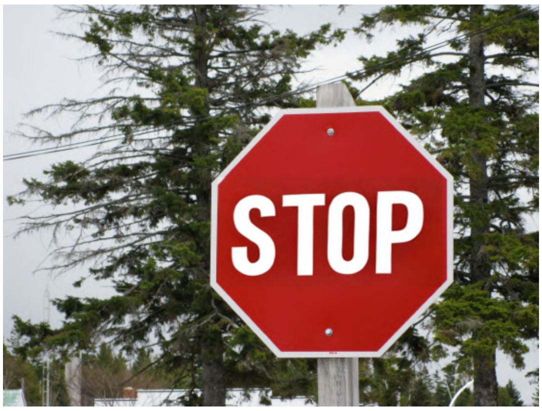

Figure 4.2: Image corrected through GenFix

4.2.2 Images with contextually incorrect text

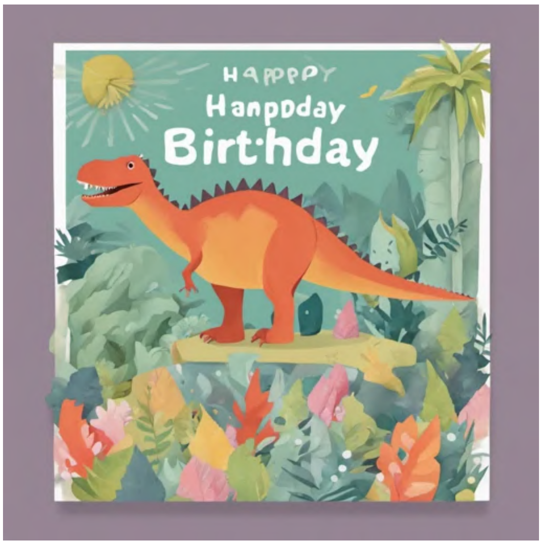

Figure 4.3: Image Generated from GenAI

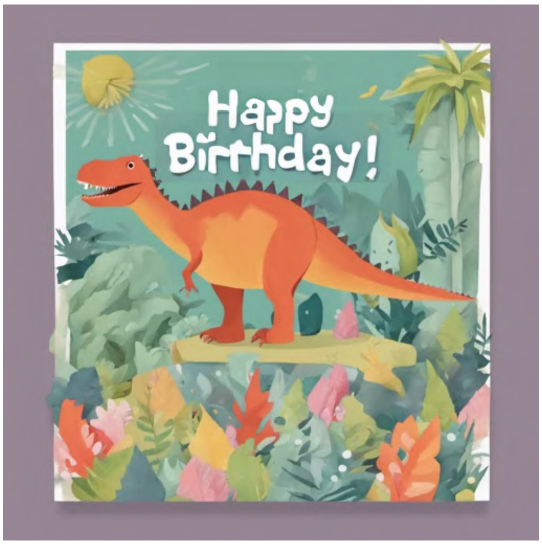

Figure 4.4: Image corrected through GenFix

Context Correction: In another instance, the model corrected the overall context of the text to better align with the visual content. As shown in Figure 4.4, it removed the redundant and misspelled occurrence of the word ”Happy”, which appeared twice and was contextually incorrect, to better fit the intended message.

4.2.3 Scene-Aware Text Generation

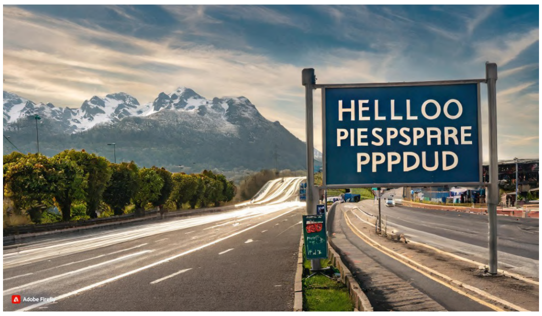

Figure 4.5: Image Generated from GenAI

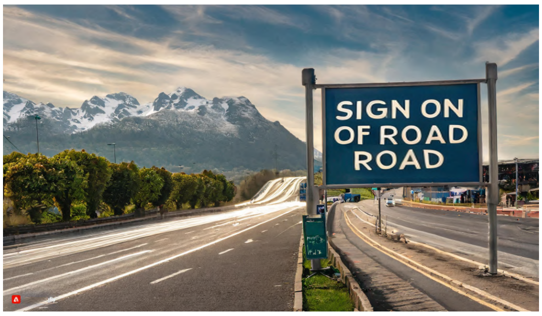

Figure 4.6: Image corrected through GenFix

Sentence Generation: GenFix can also generate new sentences in place of gibberish text depending on the scene. The models tries to find words related to the scene such as ”Road” or ”Signboard” as shown in Figure 4.6.

4.3 Model Evaluation

The text correction model, **GenFix** was evaluated using various metrics to assess both textual accuracy and image quality with the results detailed in Table 4.1.

Metric	Value	Interpretation
CER	‘0.0‘	Perfect character-level match between corrected and target texts.
WER	‘0.0‘	Perfect word-level match between corrected and target texts.
BLEU	‘0.1778‘	Misleading due to short texts; CER/WER are more reliable here (perfect).
SSIM	‘0.9555‘	Excellent structural similarity between original and corrected images.
PSNR	‘18.041 dB‘	Moderate pixel-level differences between original and corrected images.
FID	‘58.2983‘	Moderate feature-level differences; may indicate substantial corrections.

Table 4.1: Model Evaluation Metrics using the TextSynth Dataset

4.3.1 Text Accuracy Metrics

Character Error Rate (CER) and Word Error Rate (WER) both achieved perfect scores (0.0), indicating flawless text correction. The BLEU score (0.1778) is less informative due to the single-word nature of the test case used.

4.3.2 Image Quality Metrics

The Structural Similarity Index (SSIM) of 0.9555 suggests high structural preservation. The Peak Signal-to-Noise Ratio (PSNR) of 18.041 dB indicates moderate pixel-level modifications. The Fréchet Inception Distance (FID) of 58.2983 suggests moderate feature-level alterations, which is expected given the text correction task.

4.3.3 Discussion

GenFix demonstrates excellent performance in correcting textual content while maintaining high structural similarity. The moderate PSNR and high FID scores reflect the necessary changes made to correct the text from ”SOTP” to ”STOP” as in Figure 4.2. Future work could focus on minimizing pixel-level changes to improve PSNR.

Compared to existing methods as illustrated in Table 6.2 (See Appendix C), STEFANN (50) excels in maintaining style consistency and color fidelity, but is limited by its fixed-length word constraint, restricting its versatility. Its SSIM-equivalent (ASSIM) for FANnet is 0.7712, significantly lower than GenFix’s 0.9555, highlighting GenFix’s superior structural preservation. Additionally, GenFix achieves a significantly higher SSIM score (‘0.9555‘) compared to SRNet (0.79)(24) and Pix2Pix (0.63) (53), demonstrating superior image preservation. The PSNR of GenFix (18.041) is higher than that of Pix2Pix (16.54) and comparable to SRNet (21.12), demonstrating its ability to accurately correct text while having fewer distortions and noise.

The model was also evaluated for *word-level accuracy*, as shown in Table 4.2, and an *ablation study* was conducted, as illustrated in Table 6.1 (See Appendix A). The ablation study evaluated the individual impact of the model pipeline components by systematically removing key elements. (See Appendix A)

Performance Across Different Text Lengths

We evaluated the performance of our text correction pipeline on OCR-generated outputs across 5 images from TextSynth dataset, each containing text sequences of varying lengths, ranging from single-word instances to five-word phrases. The results, summarized in Table 4.2, show that shorter texts (1, 2, and 3 words) achieved near-perfect performance, with both F1 score and word accuracy at 1.0. However, as the text length increased, the model’s performance slightly declined, particularly in terms of the F1 score.

Text Length	F1 Score	Word Accuracy
1	1.0	1.0
2	1.0	1.0
3	1.0	1.0
4	0.8	0.8
5	0.6	0.8

Table 4.2: Performance of the text correction pipeline across different text lengths.

The table demonstrates that while word accuracy remained high for longer texts, the F1 score reflected some challenges in balancing precision and recall. This suggests that the model struggles more with correcting longer OCR outputs, which are typically more complex and prone to errors.

Chapter 5

Conclusion and Future Work

Generative AI models have shown impressive ability to create images with embedded text, yet ensuring the accuracy and coherence of this text remains a significant challenge. Currently, no open-source, fully automated solution exists for correcting text in AI-generated images. Instead, manual editing tools such as Photoshop are widely used, making the process labor-intensive and impractical for large-scale applications. Existing models like STEFANN (50), TextDiffuser (42), and AnyText (54) offer text-editing capabilities but require explicit prompts, lacking an end-to-end automated pipeline.

To address this gap, we introduced **GenFix**, an open-source automated system designed to detect, correct, and smoothly integrate text into input images. It surpasses existing approaches by combining OCR, caption generation (e.g., BLIP-2), graph-based alignment, energy-based geometric optimization and Stable Diffusion-based inpainting for precise word and sentence-level corrections. Unlike STEFANN's (50) character-level editing or SRNet's (24) reliance on synthetic data, GenFix offers better contextual understanding and adapts to complex datasets.

Future Work: Despite its effectiveness, there remain areas for further exploration. One key observation during the inpainting process was the variation in output quality across different hardware configurations. Notably, the outputs differed when executed on a CPU versus a GPU (Nvidia on Windows/Google Colab or MPS on Mac). The GPU-based implementations generally produced smoother inpainting results compared to those generated on a CPU. This might be due to Float-point precision where GPUs often use lower precision (e.g., FP16, TensorFloat-32) for performance optimization, whereas CPUs typically operate in FP32 or even FP64. This could be explored well in future research work.

In conclusion, **GenFix** provides an open-source solution for text correction in AI-generated images, by combining multiple methods along with mathematical optimization to ensure accuracy and style consistency. As the first fully automated model for this task, it fills a gap in the absence of similar software. The introduction of the **TextSynth-100** dataset further supports this effort by providing a benchmark dataset of image-text pairs for evaluating AI-generated text in images.

Bibliography

[1] OpenAI. ChatGPT: A Conversational AI Model; 2024. Available at https://openai.com/chatgpt. pages 1, 2

[2] OpenAI. Sora: AI-Powered Video Generation; 2024. Available at https://openai.com/sora. pages 1

[3] OpenAI. DALL·E: AI System for Image Generation; 2024. Available at https://openai.com/dall-e. pages 1, 16

[4] AI M. Emu: AI-Powered Image Generation; 2024. Available at https://ai.meta.com. pages 1, 3, 16

[5] Lakhanpal S, Chopra S, Jain V, Chadha A, Luo M. Refining Text-to-Image Generation: Towards Accurate Training-Free Glyph-Enhanced Image Generation; 2024. Available from: https://arxiv.org/abs/2403.16422. pages 1, 3

[6] Chen CT, Huang HH. Integrating LLM, VLM, and text-to-image models for enhanced information graphics: a methodology for accurate and visually engaging visualizations. In: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence. IJCAI '24; 2024. Available from: https://doi.org/10.24963/ijcai.2024/995. pages 1

[7] Joynt V, Cooper J, Bhargava N, et al. A comparative analysis of text-to-image generative AI models in scientific contexts: a case study on nuclear power. Scientific Reports. 2024;14:30377. Available from: https://doi.org/10.1038/s41598-024-79705-4. pages 2

[8] Microsoft, OpenAI. GitHub Copilot: AI-Powered Code Completion; 2024. Available at https://github.com/features/copilot. pages 3, 16

[9] Li M, Lv T, Cui L, Lu Y, Florêncio DAF, Zhang C, et al. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. In: AAAI Conference on Artificial Intelligence; 2021. Available from: https://api.semanticscholar.org/CorpusID:237581568. pages 3, 9, 12

[10] AI J. EasyOCR: Ready-to-use OCR with 80+ Supported Languages; 2020. Accessed: 2025-02-14. Available from: https://github.com/JaidedAI/EasyOCR. pages 3, 12

[11] Li J, Li D, Xiong C, Hoi SCH. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In: International Conference on Machine Learning; 2022. Available from: https://api.semanticscholar.org/CorpusID:246411402. pages 3, 10, 12

[12] Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B. High-Resolution Image Synthesis with Latent Diffusion Models . In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos, CA, USA: IEEE Computer Society; 2022. p. 10674-85. Available from: https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01042. pages 3, 6, 12

[13] Kingma DP, Welling M; 2019. Available from: https://ieeexplore.ieee.org/document/9051780. pages 4, 11

[14] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative Adversarial Nets. In: Ghahramani Z, Welling M, Cortes C, Lawrence N, Weinberger KQ, editors. Advances in Neural Information Processing Systems. vol. 27. Curran Associates, Inc.; 2014. Available from: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf. pages 5, 11

[15] Mishkin P, Ahmad L, Brundage M, Krueger G, Sastry G. DALL·E 2 Preview - Risks and Limitations; 2022. Available at https://github.com/openai/dalle-2-preview/blob/main/system-card.md. pages 6

[16] Saharia C, Chan W, Saxena S, Li L, Whang J, Denton EL, et al. Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. In: Koyejo S, Mohamed S, Agarwal A, Belgrave D, Cho K, Oh A, editors. Advances in Neural Information Processing Systems. vol. 35. Curran Associates, Inc.; 2022. p. 36479-94. Available from: https://proceedings.neurips.cc/paper_files/paper/2022/file/ec795aeadae0b7d230fa35cbaf04c041-Paper-Conference.pdf. pages 6, 10, 16

[17] Amara J, Lee S. Challenges of Making AI-Generated Text More Explainable. Journal of AI Research. 2023;45(3):210-25. pages 7

[18] Zhu S, Fang P, Zhu C, Zhao Z, Xu Q, Xue H. Text Image Inpainting via Global Structure-Guided Diffusion Models. Proceedings of the AAAI Conference on Artificial Intelligence. 2024 Mar;38(7):7775-83. Available from: https://ojs.aaai.org/index.php/AAAI/article/view/28612. pages 7, 8

[19] Chen G. Advancements and Challenges in Text-to-Image Synthesis: Exploring Deep Learning Techniques. Transactions on Computer Science and Intelligent Systems Research. 2024 Aug;5:515–521. Available from: https://wepub.org/index.php/TCSISR/article/view/2451. pages 7, 8

[20] Joynt V, Cooper J, Bhargava N, et al. A comparative analysis of text-to-image generative AI models in scientific contexts: a case study on nuclear power. Scientific Reports. 2024;14:30377. Available from: https://doi.org/10.1038/s41598-024-79705-4. pages 7

[21] Ivezić D, Babac MB. Trends and Challenges of Text-to-Image Generation: Sustainability Perspective. Croatian Regional Development Journal. 2023;4(1):56-77. Available from: https://doi.org/10.2478/crdj-2023-0004. pages 7

[22] Ramesh A, Dhariwal P, Nichol A, Chu C, Chen M. Hierarchical Text-Conditional Image Generation with CLIP Latents; 2022. Available from: https://arxiv.org/abs/2204.06125. pages 8

[23] Ma J, Zhao M, Chen C, Wang R, Niu D, Lu H, et al.. GlyphDraw: Seamlessly Rendering Text with Intricate Spatial Structures in Text-to-Image Generation; 2023. Available from: https://arxiv.org/abs/2303.17870. pages 8

[24] Wu L, Zhang C, Liu J, Han J, Liu J, Ding E, et al. Editing Text in the Wild. In: Proceedings of the 27th ACM International Conference on Multimedia. MM '19. New York, NY, USA: Association for Computing Machinery; 2019. p. 1500–1508. Available from: https://doi.org/10.1145/3343031.3350929. pages 8, 19, 20, 30

[25] Yang Q, Huang J, Lin W. SwapText: Image Based Texts Transfer in Scenes. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020. . pages 8

[26] Jayanthi SM, Pruthi D, Neubig G. NeuSpell: A Neural Spelling Correction Toolkit. In: Liu Q, Schlangen D, editors. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Online: Association for Computational Linguistics; 2020. p. 158-64. Available from: https://aclanthology.org/2020.emnlp-demos.21/. pages 8

[27] Li X, Liu H, Huang L. Context-aware Stand-alone Neural Spelling Correction. In: Cohn T, He Y, Liu Y, editors. Findings of the Association for Computational Linguistics: EMNLP 2020. Online: Association for Computational Linguistics; 2020. p. 407-14. Available from: https://aclanthology.org/2020.findings-emnlp.37/. pages 8

[28] Veit A, Matera T, Neumann L, Matas J, Belongie SJ. COCO-Text: Dataset and Benchmark for Text Detection and Recognition in Natural Images. CoRR. 2016;abs/1601.07140. Available from: http://arxiv.org/abs/1601.07140. pages 8

[29] Wang ZJ, Montoya E, Munechika D, Yang H, Hoover B, Chau DH. DiffusionDB: A Large-Scale Prompt Gallery Dataset for Text-to-Image Generative Models. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2023. Available from: https://aclanthology.org/2023.acl-long.51. pages 8

[30] Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998;86(11):2278-324. pages 8

[31] Smith R. An Overview of the Tesseract OCR Engine. In: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007). vol. 2; 2007. p. 629-33. pages 9

[32] Shi B, Bai X, Yao C. An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017;39(11):2298-304. pages 9

[33] Zhou X, Yao C, Wen H, Wang Y, Zhou S, He W, et al. EAST: An Efficient and Accurate Scene Text Detector. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 2642-51. pages 9

[34] Du Y, Li C, Guo R, Yin X, Liu W, Zhou J, et al. PP-OCR: A Practical Ultra Lightweight OCR System. CoRR. 2020;abs/2009.09941. Available from: https://arxiv.org/abs/2009.09941. pages 9

[35] Hosseini H, Xiao B, Poovendran R. Google's Cloud Vision API Is Not Robust To Noise. CoRR. 2017;abs/1704.05051. Available from: http://arxiv.org/abs/1704.05051. pages 9

[36] Baek Y, Lee B, Han D, Yun S, Lee H. Character Region Awareness for Text Detection . In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Los Alamitos, CA, USA: IEEE Computer Society; 2019. p. 9357-66. Available from: https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.00959. pages 9

[37] Lab SAMAR. DeepFloyd IF: A High-Resolution Text-to-Image Model with Pixel Diffusion Architecture; 2024. Accessed: 2025-02-13. Available from: https://github.com/deep-floyd/IF. pages 10

[38] Li R, Li W, Yang Y, Wei H, Jiang J, Bai Q. Swinv2-Imagen: hierarchical vision transformer diffusion models for text-to-image generation. Neural Computing and Applications. 2024;36:17245-60. Available from: https://doi.org/10.1007/s00521-023-09021-x. pages 10

[39] Yuan H, Yuan Z, Tan C, Huang F, Huang S. Text Diffusion Model with Encoder-Decoder Transformers for Sequence-to-Sequence Generation. In: Duh K, Gomez H, Bethard S, editors. Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). Mexico City, Mexico: Association for Computational Linguistics; 2024. p. 22-39. Available from: https://aclanthology.org/2024.naacl-long.2/. pages 10

[40] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. CoRR. 2019;abs/1910.10683. Available from: http://arxiv.org/abs/1910.10683. pages 10

[41] Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, et al. Learning Transferable Visual Models From Natural Language Supervision. In: Meila M, Zhang T, editors. Proceedings of the 38th International Conference on Machine Learning. vol. 139 of Proceedings of Machine Learning Research. PMLR; 2021. p. 8748-63. Available from: https://proceedings.mlr.press/v139/radford21a.html. pages 10

[42] Chen J, Huang Y, Lv T, Cui L, Chen Q, Wei F. TextDiffuser: Diffusion Models as Text Painters. In: Oh A, Naumann T, Globerson A, Saenko K, Hardt M, Levine S, editors. Advances in Neural Information Processing Systems. vol. 36. Curran Associates, Inc.; 2023. p. 9353-87. Available from: https://proceedings.neurips.cc/paper_files/paper/2023/file/1df4afb0b4ebf492a41218ce16b6d8df-Paper-Conference.pdf. pages 10, 16, 20

[43] Liu R, Garrette D, Saharia C, Chan W, Roberts A, Narang S, et al. Character-Aware Models Improve Visual Text Rendering. In: Rogers A, Boyd-Graber J, Okazaki N, editors. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Toronto, Canada: Association for Computational Linguistics; 2023. p. 16270-97. Available from: https://aclanthology.org/2023.acl-long.900/. pages 10

[44] Chen H, Xu Z, Gu Z, lan j, i Y, et al. DiffUTE: Universal Text Editing Diffusion Model. In: Oh A, Naumann T, Globerson A, Saenko K, Hardt M, Levine S, editors. Advances in Neural Information Processing Systems. vol. 36. Curran Associates, Inc.; 2023. p. 63062-74. Available from: https://proceedings.neurips.cc/paper_files/paper/2023/file/c7138635035501eb71b0adf6ddc319d6-Paper-Conference.pdf. pages 10

[45] Shimoda W, Inoue N, Haraguchi D, Mitani H, Uchida S, Yamaguchi K. Type-R: Automatically Retouching Typos for Text-to-Image Generation; 2024. Available from: https://arxiv.org/abs/2411.18159. pages 10

[46] Zhu W, Wang X, Lu Y, Fu TJ, Wang X, Eckstein M, et al. Collaborative Generative AI: Integrating GPT-k for Efficient Editing in Text-to-Image Generation. In: Bouamor H, Pino J, Bali K, editors. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Singapore: Association for Computational Linguistics; 2023. p. 11113-22. Available from: https://aclanthology.org/2023.emnlp-main.685/. pages 10

[47] Suveeranont R, Igarashi T. Example-Based Automatic Font Generation. In: Taylor R, Boulanger P, Krüger A, Olivier P, editors. Smart Graphics. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010. p. 127-38. pages 11

[48] Campbell NDF, Kautz J. Learning a manifold of fonts. ACM Trans Graph. 2014 Jul;33(4). Available from: https://doi.org/10.1145/2601097.2601212. pages 11

[49] Phan HQ, Fu H, Chan AB. FlexyFont: Learning Transferring Rules for Flexible Typeface Synthesis. Comput Graph Forum. 2015 Oct;34(7):245–256. Available from: https://doi.org/10.1111/cgf.12763. pages 11

[50] Roy P, Bhattacharya S, Ghosh S, Pal U. STEFANN: Scene Text Editor Using Font Adaptive Neural Network. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019:13225-34. Available from: https://api.semanticscholar.org/CorpusID:67856203. pages 11, 19, 20, 30

[51] Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B. High-Resolution Image Synthesis With Latent Diffusion Models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022. p. 10684-95. pages 11

[52] Wang B, Zhu Y, Chen L, Liu J, Sun L, Childs P. A study of the evaluation metrics for generative images containing combinational creativity. Artificial Intelligence for Engineering Design, Analysis and Manufacturing. 2023;37:e11. pages 11

[53] Isola P, Zhu JY, Zhou T, Efros AA. Image-to-Image Translation with Conditional Adversarial Networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017. p. 5967-76. pages 19, 30

[54] Tuo Y, Xiang W, He JY, Geng Y, Xie X. AnyText: Multilingual Visual Text Generation and Editing. In: The Twelfth International Conference on Learning Representations; 2024. Available from: https://openreview.net/forum?id=ezBH9WE9s2. pages 20

Chapter 6

Declaration

I confirm that this project report titled *Correcting Text in Images Generated from Generative AI Models* is my own work and has been carried out in partial fulfillment of the requirements for the MSc Advanced Computing degree at Imperial College London. Any use of external sources, including research papers, datasets, and software libraries, has been properly cited and acknowledged.

Use of Generative AI

This research leverages Generative AI models for generating AI-based image-text pairs, using pre-trained Vision-Language Models (BLIP-2) and Stable Diffusion-based inpainting techniques. I acknowledge the use of DALL·E 3 (OpenAI), Imagen (Gemini), Copilot, Emu (Meta AI), and Hugging Face only for dataset generation. Grammarly was used for grammar refinement. No AI-generated content was used; I confirm all findings and figures are based on rigorous experimentation and analysis.

Ethical Considerations

This research does not involve human participants. The datasets used consist of publicly available AI-generated images, ensuring compliance with ethical guidelines. No copyright-protected or confidential material has been included in this study.

Sustainability

The computational experiments were optimized to reduce energy consumption. Pre-trained models and efficient GPU utilization strategies were employed to minimize carbon footprint. Cloud-based resources (Google Colab) were used responsibly, and unnecessary re-training of models was avoided to improve sustainability.

Availability of Data and Materials

The source code and datasets used in this research are available at GitHub Repository. All supplementary materials, including experimental results and model evaluations, have been included in the report.

Appendix

A. Ablation Study on Text Correction Pipeline

In this study, we evaluate the effectiveness of different components of our text correction pipeline for images. The pipeline was tested under four distinct configurations to understand the impact of each component on the text correction performance. The conditions include: (1) **Baseline (No Correction)**: the system without any correction, (2) **With Simulated Annealing**: where simulated annealing is used for alignment, (3) **With OCR In-painting**: where OCR-based inpainting is applied to the detected text regions, and (4) **Full Model**: where both simulated annealing and OCR inpainting are utilized together.

Method	**Avg F1 Score**
Baseline (No Correction)	0.750
With Simulated Annealing	0.800
With OCR In-painting	0.850
Full Model	0.950

Table 6.1: Average F1 scores for each method evaluated on the dataset

The evaluation was carried out on our TextSynth-100 dataset, with performance measured using the word-level F1 score. As shown in Table 6.1, the baseline method, which applies no correction, achieved an F1 score of 0.750. Adding simulated annealing provided a moderate improvement, increasing the score to 0.800. The OCR in-painting approach further boosted performance to 0.850. However, the full model significantly outperformed all other methods, reaching an impressive F1 score of 0.950. Interestingly, while simulated annealing and OCR in-painting contributed to overall accuracy, their impact was less pronounced than expected.

B. TextSynth-100 Dataset

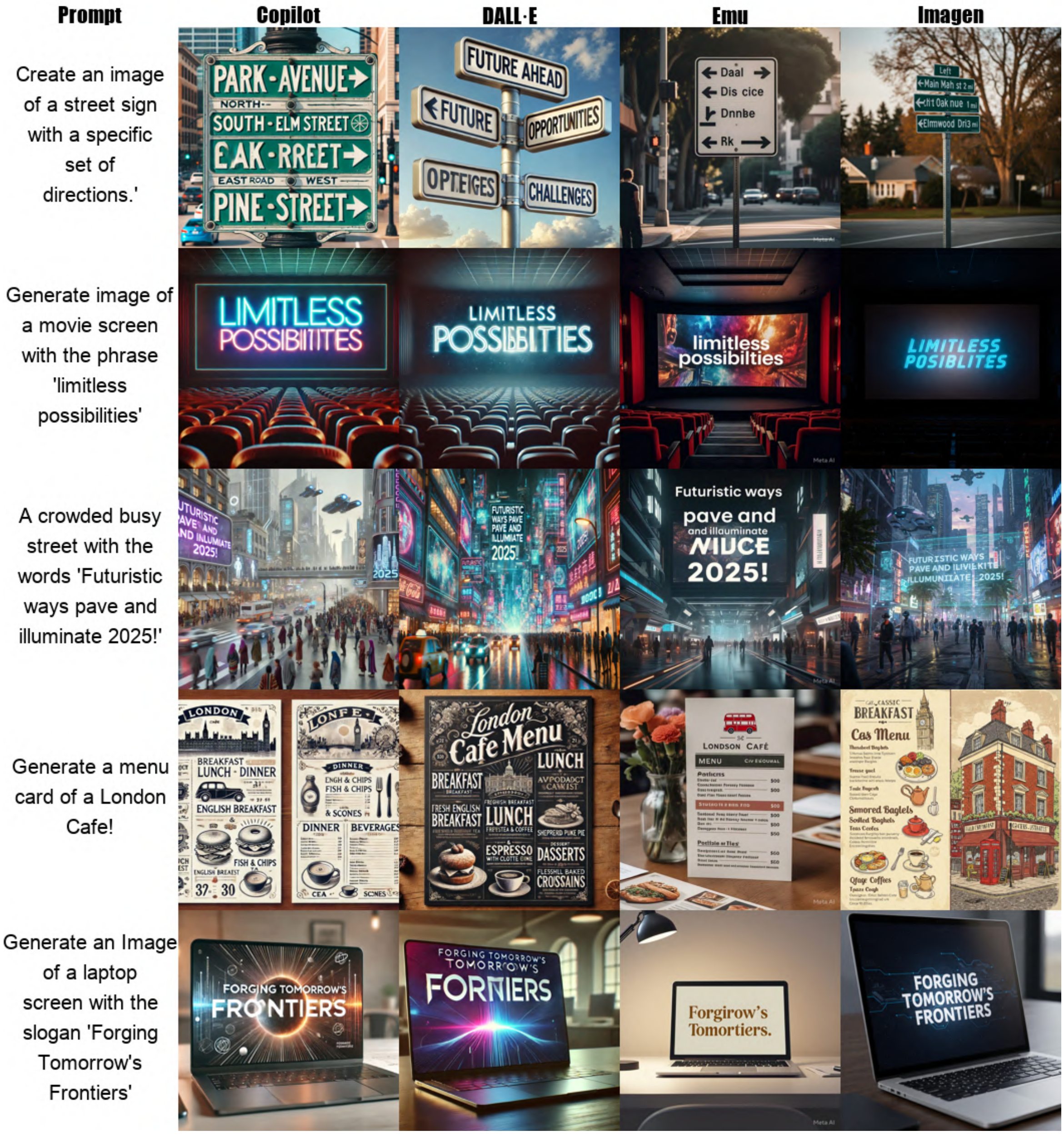
Prompt
Copilot
DALL·E
Emu
Imagen
Create an image of a street sign with a specific set of directions.'
Generate image of a movie screen with the phrase 'limitless possibilities'
A crowded busy street with the words 'Futuristic ways pave and illuminate 2025!'
Generate a menu card of a London Cafe!
Generate an Image of a laptop screen with the slogan 'Forging Tomorrow's Frontiers'

Figure 6.1: TextSynth-100 Dataset

The **TextSynth-100** dataset is available in the `TextSynth_dataset` folder on GitHub as a zipped file for easy access. Each image file is named after the prompt used to generate it, making it easier to track and analyze text correction performance. The dataset will be further expanded in size in future work to help improve the generalizability of the image-text correction models.

C. Comparative Analysis

Method	SSIM / ASSIM	PSNR	FID
GenFix	**0.9555**	18.041	High
STEFANN (FANnet) (50)	0.7712	-	High
SRNet (24)	0.79	21.12	High
Pix2Pix (53)	0.63	16.54	Moderate

Table 6.2: Performance comparison of GenFix with existing methods

For FID, GenFix was tested on the TextSynth-100 dataset, while other methods were evaluated on datasets like COCO-Text and ICDAR, which may differ in text complexity and distribution.

D. Proofs

Theorem 1: Optimality of the Hungarian Algorithm

Statement: The Hungarian algorithm computes the permutation σ^* that minimizes the total assignment cost $\sum D_{i,\sigma(i)}$ for a square cost matrix D.

Proof: The Hungarian algorithm constructs a primal-dual feasible solution, satisfying the complementary slackness conditions in linear programming. It is guaranteed to converge to the minimal cost in $O(N^3)$ time (Kuhn, 1955).

Theorem 2: Convexity of the Energy Function

Statement: The energy function L_A is strictly convex, ensuring a unique global minimum.

Proof: Rewriting L_A as $\mathbf{v}^T\mathbf{Q}\mathbf{v} + \mathbf{b}^T\mathbf{v} + c$, the Hessian $\mathbf{Q}$ is positive definite due to squared terms. Thus, L_A has no saddle points or local minima.

Theorem 3: Error Bound for Geometric Adjustment

Statement: The positional error post-optimization satisfies $\|\mathbf{x}'_i - \mathbf{x}_i\| \leq \epsilon_{\text{orig}} + \frac{\mu d}{\lambda_1}$.

Proof Sketch: From the KKT conditions, the adjustment $\Delta\mathbf{x}_i$ balances fidelity and alignment. Solving $\nabla L_A = 0$ bounds $\Delta\mathbf{x}_i$ by the trade-off parameter μ/λ_1.