IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155246

Countify: A Text-to-Image Generation Model

Prof. Dr.S.S.More[1], Athrav Raju Ugale[2], Vaidehi KeshavVaze[3], Vaishnav Bajrang Patil[4], Alisha Mubarak Attar[5], Sanika Appasaheb Patil[6]

Assistant Professor, Computer Science Engineering (Artificial Intelligence), DKTE Ichalkaranji[1]

Final Year B.Tech, Computer Science Engineering (Artificial Intelligence), DKTE Ichalkaranji[2-6]

Abstract : This paper presents **Countify**, a system that integrates text-to-image generation with object detection and an iterative feedback mechanism to ensure precise object counts in generated images. The system utilizes diffusion-based image generation via ClipDrop API and object detection using YOLOv8. A validation loop continuously refines outputs until the detected object count matches the requested count.

Experimental results demonstrate that Countify significantly improves numerical accuracy in generated images, making it suitable for applications requiring precision such as dataset generation, education, and industrial automation.

Keywords: Text-to-Image Generation, Object Counting, YOLOv8, Diffusion Models, Feedback Loop, Generative AI, Computer Vision

I. INTRODUCTION

1.1 Background

Text-to-image (T2I) generation has emerged as one of the most impactful applications of generative artificial intelligence, enabling systems to convert natural language descriptions into realistic images. Models such as diffusion-based generators and transformer-based architectures have demonstrated remarkable capabilities in generating visually coherent and semantically aligned outputs.

Despite these advancements, current T2I systems struggle with enforcing **numerical constraints**, particularly when the input prompt specifies an exact number of objects. For example, a prompt like "three dogs" may produce images containing two, four, or an inconsistent number of objects. This limitation arises because object count is a **global attribute**, requiring consistent coordination across multiple spatial regions in an image.

1.2 Problem Statement

Existing T2I systems exhibit the following limitations:

- Inability to generate exact object counts
- Lack of verification mechanisms
- No feedback loop to correct errors
- Poor reliability in precision-based applications

1.3 Research Objectives

- To generate images from textual prompts
- To ensure **exact object count accuracy**
- To automatically verify object count using AI
- To implement an **iterative retry mechanism** for correction
- To store and track results for analysis

1.4 Significance

This research enhances the reliability of generative AI systems by introducing **quantitative validation**, making them suitable for:

- Educational tools
- Synthetic dataset generation
- Industrial design automation
-

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155246

II. LITERATURE REVIEW

2.1 Text-to-Image Generation Models

Diffusion-based models such as Stable Diffusion generate images by gradually refining noise into structured visuals. While these models excel in realism, they lack mechanisms for enforcing numerical constraints.

2.2 Object Detection Techniques

Object detection models like YOLO (You Only Look Once) perform real-time detection using convolutional neural networks (CNNs). YOLOv8 provides efficient and accurate detection with bounding box predictions and class labels.

2.3 Controllable Text-to-Image Generation

Existing controllable T2I methods primarily focus on:

- Spatial layout control
- Style and attribute alignment

However, they lack mechanisms for enforcing **exact numerical constraints**, which limits their effectiveness in applications requiring precise object counts.

2.4 Research Gap

The key research gaps identified are:

- Absence of differentiable counting models for T2I integration
- Lack of explicit quantity control mechanisms
- Limited generalization across diverse object categories

III. METHODOLOGY

3.1 System Overview

The proposed system, **Countify**, is designed to ensure precise object count control in text-to-image generation by integrating three major components: **image generation**, **object detection**, and a **feedback-based validation loop**.

The first component focuses on generating images from textual prompts using a diffusion-based model accessed through the ClipDrop API. The second component utilizes a deep learning-based object detection model, specifically YOLOv8, to identify and count objects present in the generated image. The third component is an iterative feedback mechanism that validates whether the generated output satisfies the required object count and triggers regeneration if necessary.

By combining these components, the system creates a closed-loop pipeline that continuously refines outputs until the desired numerical accuracy is achieved. This architecture ensures both **visual realism and quantitative correctness**, which is a significant improvement over traditional generative models.

3.2 System Workflow

The workflow of the Countify system begins with the user providing a textual input prompt, such as “5 cats”. The system first processes this input using a prompt parsing module that extracts the required object count and object category.

Once the input is parsed, the system generates an image using a text-to-image generation API. The generated image is then passed to the YOLOv8 object detection model, which identifies all instances of the specified object and counts them.

The detected object count is compared with the requested count. If both values match, the image is accepted and returned to the user. Otherwise, the system activates a retry mechanism, regenerating the image and repeating the detection process. This loop continues for a maximum of five attempts or until a correct result is achieved.

This workflow ensures that the system does not rely on a single generation attempt but instead employs **iterative refinement**, significantly increasing the probability of achieving accurate results.

3.3 Algorithms Used

3.3.1. Diffusion Model (via ClipDrop API)

- Used for image generation
- Generates images from text prompts

3.3.2. YOLOv8 (You Only Use Inference)

- Object detection algorithm
- Uses bounding box detection
- Counts objects

3.3.3. Rule-based Parsing Algorithm

- Regex-based extraction of:

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155246

- number
- object name

3.3.4. Iterative Retry Algorithm (Core Contribution)

```
for attempt in range(1, MAX_ATTEMPTS):
   generate image
   count objects using YOLO
   if count == required:
      return success
return best result
```

3.4 Mathematical Representation

To formally define the validation process, let:

- Kreq represent the required object count specified in the input prompt
- Kdet represent the number of objects detected by the YOLO model

Validation Condition

The system accepts the generated image only if:

$$\text{Kdet} = \text{Kreq}$$

This condition ensures that the output satisfies the exact numerical requirement.

Loss Function

To quantify the error between the generated output and the desired result, a loss function is defined as:

$$\text{L} = |\text{Kdet} - \text{Kreq}|$$

This loss represents the absolute difference between detected and required object counts. The objective of the system is to minimize this value, ideally reaching zero, which indicates perfect numerical alignment.

IV. EXPERIMENTAL SETUP

4.1 Dataset Description

The proposed system does not involve training of models and therefore does not require custom datasets.

Instead, it utilizes pre-trained models:

• ClipDrop API: A diffusion-based text-to-image generation model trained on large-scale text-image datasets
• YOLOv8: A pre-trained object detection model trained on the COCO dataset (Common Objects in Context), which includes 80 object categories such as person, car, bottle, dog, etc.

These pre-trained models enable the system to perform image generation and object detection without additional training.

4.2 Evaluation Metrics

The performance of the system is evaluated using the following metrics:

• Counting Accuracy
Measures whether the detected object count matches the requested count.

• Absolute Count Difference (ACD)
ACD = |Detected Count − Requested Count|

• Success Rate
Percentage of prompts where the system achieves exact object count within the allowed attempts.

• Attempt Efficiency
Number of attempts required to achieve correct output.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155246

V. RESULTS AND DISCUSSION

5.1 Experimental Results

The proposed system demonstrates the following results:

• High accuracy for small object counts (e.g., 3–7 objects)
• Moderate accuracy for medium counts (e.g., 8–15 objects)
• Reduced accuracy for large counts (e.g., 20+ objects), due to limitations of generative models

The iterative retry mechanism significantly improves the probability of achieving the correct object count compared to single-pass generation.

5.2 Analysis

• The feedback loop approach improves numerical accuracy without modifying the generative model
• YOLOv8 effectively detects common objects but may struggle with complex scenes or non-standard objects
• Diffusion-based models do not inherently guarantee exact object count, requiring external validation
• The system performs best when objects are clearly separated and visually distinct

VI. PRACTICAL APPLICATIONS

The YOLO-Count framework has wide-ranging applications, including:
• Synthetic dataset generation with controlled object counts
• Educational tools for demonstrating AI concepts
• Automated testing of generative AI models
• Inventory visualization systems
• Design and simulation tools requiring precise object placement

VII. LIMITATIONS AND FUTURE WORK

7.1 Limitations

• The system depends on external APIs (ClipDrop), which may have usage limits
• YOLOv8 supports only predefined object classes (COCO dataset)
• Performance decreases for large object counts or complex prompts
• Generated images may not always follow spatial instructions (e.g., "in sky", "in row")
• Multiple attempts increase latency

7.2 Future Enhancements

• Integration with more advanced generative models (e.g., Stable Diffusion, DALL·E)
• Improving prompt engineering for better control over object placement
• Supporting multi-object counting (e.g., "5 cats and 3 dogs")
• Real-time optimization to reduce generation time
• Fine-tuning object detection models for domain-specific use cases

VIII. CONCLUSION

This paper presented Countify, a system designed to address the problem of numerical inconsistency in text-to-image generation. By combining diffusion-based image generation with YOLOv8-based object detection and an iterative feedback loop, the system ensures improved accuracy in object counting.Unlike traditional approaches, Countify does not modify the generative model but introduces a validation mechanism to verify and correct outputs. Experimental results show that the system significantly improves the reliability of generating images with exact object counts.This approach demonstrates a practical and scalable solution for enhancing controllability in generative AI systems.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155246

REFERENCES

[1] N. Amini-Naieni, K. Amini-Naieni, T. Han, and A. Zisserman, "Openworld text-specified object counting," in Proc. British Machine Vision Conf. (BMVC), 2023.

[2] N. Amini-Naieni, T. Han, and A. Zisserman, "CountGD: Multi-modal open-world counting," in Advances in Neural Information Processing Systems (NeurIPS), 2024.

[3] C. Arteta, V. Lempitsky, and A. Zisserman, "Counting in the wild," in Proc. European Conf. on Computer Vision (ECCV), 2016.

[4] A. Bansal, H.-M. Chu, A. Schwarzschild, S. Sengupta, M. Goldblum, J. Geiping, and T. Goldstein, "Universal guidance for diffusion models," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW), 2023.

[5] H. Chefer, Y. Alaluf, Y. Vinker, L. Wolf, and D. Cohen-Or, "Attend-and excite: Attention-based semantic guidance for text-to-image diffusion models," ACM Trans. Graph., vol. 42, no. 4, 2023.

[6] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "YOLO-World: Real-time open-vocabulary object detection," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2024.

[7] R. Gal, Y. Alaluf, Y. Atzmon, O. Patashnik, A. H. Bermano, G. Chechik, and D. Cohen-Or, "An image is worth one word: Personalizing text-to image generation using textual inversion," in Proc. Int. Conf. Learning Representations (ICLR), 2023.

[8] R. Jiang, L. Liu, and C. Chen, "CLIPCount: Towards text-guided zero shot object counting," in Proc. ACM Int. Conf. Multimedia (ACM MM), 2023.

[9] A. Gupta, P. Dollar, and R. Girshick, "LVIS: A dataset for large ´ vocabulary instance segmentation," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2019.

[10] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: Ultralytics documentation

