Image Captioning

Grigor Grigoryan, Vincent Heuer, Philip Zetterberg

ABSTRACT

This paper explores advancements in image captioning through modifications to encoder-decoder architectures, text representations, and training methodologies. We examine various encoder models (ResNet, VGG16, ViT) and decoders (GRU, LSTM, GPT-2) alongside character and word-level tokenization. The study also investigates optimization strategies, including adaptive optimizers, learning rate scheduling, and hyperparameter search with Optuna, to improve training efficiency and reduce overfitting. Results show that transformer-based architectures, particularly ViT-GPT2, outperform traditional CNN-RNN models, with fine-tuning playing a critical role in performance enhancement. Additionally, the integration of synthetic data, generated via diffusion models, demonstrates potential for augmenting model robustness, though excessive reliance on synthetic data reduces caption quality. The findings underscore the necessity of balanced architectural choices, training strategies, and dataset composition in advancing image captioning technologies.

I. Introduction

IMAGE Captioning is the automatic generation of captions for natural images using neural network based methods. It has numerous applications and benefits. For example, it improves content searchability. When images are tagged with accurate captions, it becomes easier to search for them using text-based queries. This enhances the efficiency of image databases and search engines. Furthermore, image captioning has the potential to increase accessibility for people with visual impairments. By generating descriptive captions, image captioning ensures that those who are blind or have low vision can understand the content through screen readers.

Overall, AI-driven image captioning helps bridge the gap between visual content and textual information. For this reason, we explore state-of-the-art (SOTA) image captioning methods. The first part of this report focuses on establishing a baseline. The goal is to train a basic model on a public dataset that contains images of food along with their respective titles. The model aims to predict the caption for each dish. We test common best practices, such as experimenting with different encoders, decoders, and text representations. The encoders and decoders are depreciated by most modern standards but this forms the foundation for the second part of the report. The aim here is to delve deeper into the mechanisms and intricacies of image captioning models. This will be achieved by using different SOTA techniques. The encoders and decoders are changed with vision transformers and LLMs. These architectures require different training techniques due to model size and functionality.

The first section highlights related work to establish a theoretical basis. Next, the methodology is outlined. The third section introduces the experimental design, followed by the presentation and in-depth analysis of the results. The report concludes with a final discussion and suggestions for future work.

II. Related Work

Several studies have explored image captioning using AI, addressing challenges such as feature extraction, sequence modeling, and evaluation metrics. In this section, we review three key papers that contribute to these areas.

Liming Xu et al. conducted an in-depth review of common practices in image captioning. [1] Their work provides valuable insights into various methodologies. They highlight the fundamental building blocks of image captioning models. The review covers traditional approaches such as template-based captioning, as well as modern deep learning methods, including encoder-decoder architectures. Additionally, they discuss different types of image feature extraction techniques, such as convolutional neural networks (CNNs), and their impact on caption generation. Their study emphasizes the importance of incorporating attention mechanisms to improve the contextual relevance of generated captions.

The survey by Lakshita Agarwal and Bindu Verma provides a comprehensive overview of the distribution of different model architectures used in image captioning. [2] They categorize models based on their structure, including CNN-RNN hybrids, Transformer-based architectures, and reinforcement learning-based models. Furthermore, the paper discusses common evaluation metrics such as BLEU, METEOR, ROUGE, and CIDEr, detailing their strengths and limitations. A key contribution of this survey is its analysis of how different models perform across various datasets, offering insights into the trade-offs between computational efficiency and caption quality.

Matteo Stefanini et al. focus on different methods for extracting and utilizing image features in captioning models. [3] They describe three primary techniques: using global CNN features, applying attention mechanisms over grid-based features, and implementing object-level attention. Their study highlights the effectiveness of object-based attention models, which leverage object detection frameworks like Faster R-CNN to capture fine-grained details. They also explore the challenges of balancing global context preservation with localized feature extraction, concluding that hybrid approaches combining both strategies yield superior results.

All three papers provide in-depth reviews or surveys on deep learning-based image captioning. A key takeaway is that while significant progress has been made, many challenges remain, such as improving caption diversity, handling rare objects, and reducing reliance on large-scale labeled datasets. Most modern methods rely on CNN-based encoding-decoding architectures that leverage attention mechanisms and global feature preservation. Furthermore, all studies recognize the importance of semantic understanding and systematic image representation in generating meaningful captions.

The insights gained from these papers inform the experimental setup and improvements explored in this project. By understanding the strengths and limitations of existing approaches, we aim to develop a more effective image captioning model that builds upon the latest advancements in the field.

III. Methodology

This methodology focuses on enhancing image captioning through encoder modifications (e.g., ResNet and VGG16), decoder changes (e.g., LSTM with varying layers), and different text representations (character to word-level). We test optimizers like SGD, Adam, and AdamW, along with training strategies such as learning rate adjustments and teacher forcing.

We explored various encoder and decoder architectures to understand how different design choices affect model performance. Below is a detailed explanation of the architectures we used for both the encoders and decoders.

Encoder Architectures

- **ViT (Vision Transformer)**: The ViT model is a transformer-based architecture specifically designed for image processing. It works by dividing the image into fixed-size patches, which are then flattened and passed through a series of transformer layers. Each patch is treated as a token, and the model captures the relationships between them using self-attention mechanisms, enabling it to model long-range dependencies across the image. This ability to capture global context makes ViT highly effective for tasks where understanding spatial relationships across the entire image is crucial.
  In our experiments, we observed that ViT's strength lies in its ability to model these global spatial relationships. This capability was particularly beneficial for sequence generation in downstream tasks, where understanding the full context of an image or sequence is essential. By leveraging the self-attention mechanism, ViT could capture subtle dependencies and interactions across the image, which contributed positively to task performance.

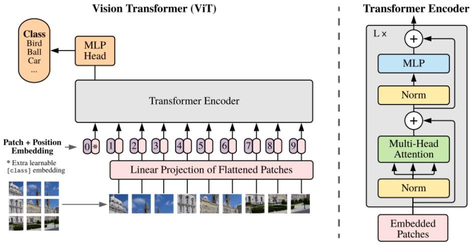

Fig. 1: ViT Model overview [4]

- **ResNet-18**: ResNet-18 is a convolutional neural network with 18 layers, using residual connections to help train deeper networks by addressing the vanishing gradient problem. The architecture consists of convolutional blocks followed by fully connected layers, with skip connections that allow the network to learn residual mappings. This design ensures better gradient flow, making it easier to train deeper models without performance degradation.
  ResNet-18's shallow depth allows it to capture local features efficiently, making it ideal for tasks that focus on low-level pattern recognition, such as edge detection and basic image classification. While computationally efficient, it benefits from the powerful residual learning mechanism.

- **VGG16**: VGG16 is a classical deep CNN known for its simple yet effective design, utilizing small 3x3 convolutional filters stacked in deep layers. With 16 layers in total—13 convolutional and 3 fully connected—VGG16 excels at feature extraction by consistently learning hierarchical representations. Its straightforward architecture allows it to extract local features effectively, which makes it well-suited for image classification tasks. Although not as advanced as some newer models, VGG16 remains strong in various computer vision applications due to its reliable and consistent performance in feature extraction. In our work, we used VGG16 to explore the impact of traditional convolutional approaches. Its deep architecture enables it to learn complex features through a series of convolutional layers, making it a valuable tool for analyzing feature extraction in computer vision tasks.

Decoder Architectures

- **LSTM (Long Short-Term Memory)**: LSTM networks are a specialized type of recurrent neural network (RNN) designed to model long-range dependencies in sequential data. The architecture of an LSTM cell consists of key components: the input gate, forget gate, output gate, and cell state. The input gate controls how much new information enters the memory, the forget gate decides which information to discard, and the output gate determines what part of the memory should be passed to the next time step. The cell state acts as the memory of the network, carrying relevant information across time steps and being updated by the gates. This unique structure allows LSTMs to preserve important information over long sequences, mitigating the vanishing gradient problem that often affects traditional RNNs.

The LSTM network is typically stacked in multiple layers, each layer consisting of multiple LSTM cells. The deeper the network, the more complex relationships it can learn, especially when capturing temporal dependencies at various levels of abstraction. Each additional layer allows the model to refine its understanding of the sequence, passing information from one time step to the next. The architecture can be designed with one or more LSTM layers depending on the complexity of the task at hand.

In our setup, we experimented with varying the number of LSTM layers and the size of the hidden state. Adding more layers allows the model to capture increasingly complex temporal relationships in the data, providing a deeper understanding of sequential dependencies. Increasing the hidden state size enables the model to store and process more information, which can improve performance on tasks that require understanding complex patterns. However, larger hidden states come with the trade-off of increased risk of overfitting, particularly when the dataset is small or lacks sufficient variety. To mitigate this risk, we applied dropout, which randomly deactivates a fraction of the network units during training, encouraging the model to learn more robust and generalizable features.

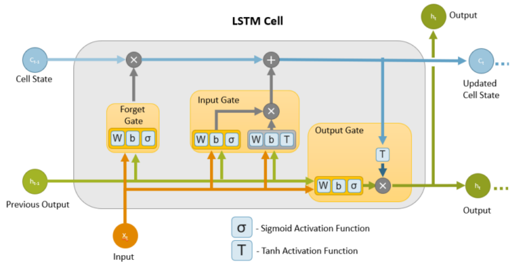

Fig. 2: The Architecture of a LSTM Cell [5]

- **GPT-2 (Generative Pre-trained Transformer 2)**: GPT-2 is a state-of-the-art, transformer-based language model that is highly effective in autoregressive sequence generation. It leverages a stack of transformer blocks, each equipped with self-attention mechanisms, which enable the model to capture long-range dependencies and intricate patterns within sequences. This architecture allows GPT-2 to process and understand the relationships between words or tokens over long stretches of text, making it especially powerful for tasks that require context and coherence over extended passages.

  In our study, we utilized GPT-2 as a decoder, examining its ability to generate coherent and contextually appropriate sequences based on features extracted by an encoder. This approach helped us assess the model's capacity to generate meaningful text from a set of encoded inputs. Due to its pre-training on vast amounts of diverse text data, GPT-2 has learned to generate high-quality sequences that are contextually relevant and semantically rich. The pre-training process exposes the model to a wide range of language structures, topics, and writing styles, which enables it to perform well across various natural language processing tasks, including but not limited to language modeling, text completion, and creative text generation. Its ability to generate fluent and contextually aware text makes GPT-2 an invaluable tool in a variety of applications in the field of artificial intelligence.

Our goal was to enhance the way textual information is represented within the model, ensuring that input and output sequences are encoded more effectively. In parallel, we aimed to refine the training process through optimization strategies that promote faster convergence, reduce overfitting, and improve the model's generalization ability across a variety of tasks.

- **Textual Representation Levels**: We tested two different levels of text representation: character-level and word-level. Character-level representations offer fine-grained control, allowing the model to handle out-of-vocabulary words and capture spelling variations. However, they result in longer sequences, which can increase computational complexity. On the other hand, word-level representations are more concise and computationally efficient, as they map entire words to tokens, but they may struggle with rare or unseen words that are not present in the vocabulary. These variations helped us assess the trade-offs between granularity, efficiency, and the model's ability to generalize to new, unseen inputs.

- **Training Strategies and Optimizers**: We explored different training strategies and optimization techniques to enhance model performance. Several optimizers were tested, including SGD (with momentum), Adam, and AdamW, each offering distinct advantages in terms of stability and adaptability. In addition, learning rate schedules, such as step decay and cosine annealing, were implemented to refine the training dynamics and ensure efficient convergence.

  To further improve the model's performance, we employed Optuna for hyperparameter optimization. This allowed us to systematically search for the optimal combination of hyperparameters, including learning rate, batch size, and regularization factors, to achieve the best results. Optuna's efficient search algorithm enabled us to explore a wide hyperparameter space while minimizing computational resources. Early stopping based on validation loss was also applied to prevent overfitting, and teacher forcing was used during early training to stabilize the learning process in the decoder.

After experimenting with different encoder and decoder architectures for image captioning, we expanded our study by generating synthetic data using diffusion models. Our objective was to evaluate whether incorporating high-quality, generated images could improve the overall performance and robustness of our captioning models. To achieve this, we used several versions of Stable Diffusion models, each offering different architectural improvements and performance characteristics.

The models we used were:

- **Stable-Diffusion-2.1**: Stable-Diffusion-2.1 is a latent diffusion model that generates images in a compressed latent space using a U-Net architecture. The model starts by encoding the image into a latent space using a Variational Autoencoder (VAE), making the process more efficient. The U-Net then iteratively denoises this latent representation to generate a clean image. Cross-attention layers condition the image generation on a text prompt, guiding the model to create relevant images. This approach reduces computational complexity and speeds up the image generation process while maintaining high-quality results.

- **Stable-Diffusion-2.1-turbo**: Stable-Diffusion-2.1-turbo is an optimized version of Stable-Diffusion-2.1 designed for faster image generation. It uses the same architecture but reduces the number of denoising steps, significantly speeding up inference time. While the VAE and U-Net remain intact, the model sacrifices some fine details and texture accuracy for quicker generation. The cross-attention layers still condition the image generation on the text prompt, but the reduced steps may slightly impact fidelity in complex scenes or intricate textures. This makes it suitable for real-time applications where speed is more critical than perfect image detail, such as in interactive design tools or rapid prototyping environments. As a result, users can expect a balance between speed and visual quality, with a focus on efficiency.

- **Stable-Diffusion-XL (SDXL)**: Stable-Diffusion-XL is a more advanced version with a two-stage pipeline. In the first stage, a U-Net generates a low-resolution image, and in the second stage, the model refines the image to a higher resolution, adding finer details. The model is larger and includes enhanced attention mechanisms, allowing it to generate photorealistic images that accurately reflect complex prompts. This two-stage refinement process ensures both structural coherence and high-quality details, making SDXL ideal for generating detailed and realistic images from text.

- **Stable-Diffusion-XL-turbo (SDXL-turbo)**: SDXL-turbo is a faster version of SDXL, designed to generate images more quickly without drastically compromising quality. It uses the same two-stage pipeline as SDXL but reduces the number of diffusion steps required for both stages, resulting in faster generation times. While this trade-off might slightly reduce image fidelity, SDXL-turbo retains much of the photorealistic quality of SDXL, particularly in terms of detail and color accuracy. The model is optimized for speed, making it suitable for real-time image generation applications, such as interactive tools or large-scale batch processing. This increased efficiency comes at the cost of slightly less intricate details, but it strikes a strong balance between performance and visual output.

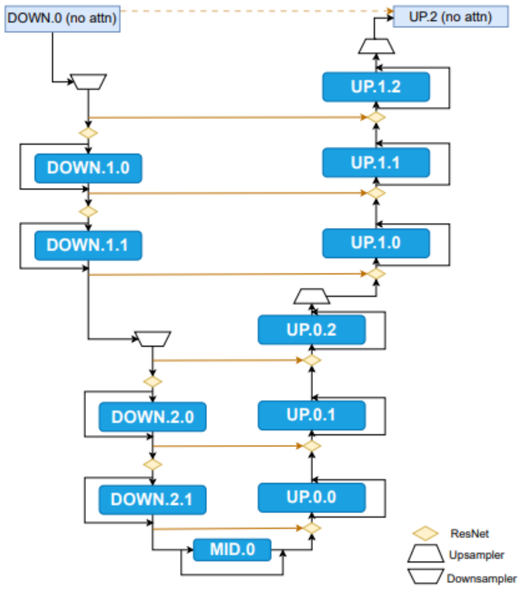

Fig. 3: Cross-attention transformer blocks in SDXL's U-net [6]

IV. Experimental Settings

A. Datasets

The project focuses on image captioning using the Food Ingredients and Recipes dataset. This dataset provides images of various food items, and the goal is to generate accurate captions that describe the contents and context of the food images.

Food Ingredients and Recipes

The Food Ingredients and Recipes Dataset includes 13,582 images of various dishes, with metadata such as dish titles, ingredients, instructions, and a cleaned list of ingredients. The dataset is split into 80% for training, 10% for validation, and 10% for testing. The images are linked to the metadata via an Image_Name column, which maps to files in a zipped folder containing the food images. The objective of this dataset is to predict the title of a dish based on its image, leveraging both visual features and ingredient-related text. [7]

B. Metrics

- **BLEU-1:** Measures the precision of individual words (1-grams) in the predicted text compared to the reference. It calculates the percentage of matching 1-grams in the predicted output and the reference text, penalizing for missing or incorrect words. However, it does not consider word order or context, making it a simple but effective measure for evaluating basic word-level accuracy. [8]
- **BLEU-2:** Extends BLEU-1 by evaluating 2-grams (pairs of adjacent words). It calculates the precision of 2-grams in the predicted text and compares it to the reference. This metric captures some syntactic structure and provides more context than BLEU-1, rewarding models for matching consecutive word pairs. [8]
- **METEOR:** Evaluates translation quality by considering exact word matches, synonyms, stemming (word root matching), and word order. METEOR calculates a score by aligning words in both the predicted and reference text, offering a higher score for more semantic alignment (e.g., matching synonyms) and penalizing longer mismatches. It provides a more nuanced evaluation, especially for natural language tasks like translation and summarization. [9]
- **ROUGE-L:** Focuses on the longest common subsequence (LCS) between the predicted and reference texts. The LCS is the longest sequence of words that appear in both texts in the same order. ROUGE-L evaluates both precision and recall based on the LCS, measuring how well the structure and flow of the text are preserved. It's particularly useful for evaluating tasks like summarization, where maintaining the sequence of ideas is important. [10]

V. Results

A. Quantitative Comparison Across Weeks

Model	Week	BLEU-1	BLEU-2	ROUGE-L	METEOR	Training Time (hr)	Device
Baseline (ResNet+GRU)	1	0.0012	0.0000	0.0024	0.0012	8	GTX 3090
VGG19 (Frozen)	1	0.0376	0.0000	0.0414	0.0182	1.5	GTX 3090
LSTM	1	0.0019	0.0003	0.0042	0.0023	8	GTX 1660
ViT-GPT2	2	0.1019	0.0362	0.1261	0.1079	1	GTX 3090
ViT	2	0.0654	0.0067	0.1055	0.0840	3.5	GTX 1660
GPT2	2	0.1009	0.0301	0.0986	0.0775	1	GTX 3090
Gemma 12B	2	0.0119	0.0030	0.0662	0.0381	11	GTX 3090
ViT + Llama 1B	2	0.0307	0.0057	0.0430	0.0414	3.5	GTX 3090
ViT + Llama 3B	2	0.0189	0.0049	0.0315	0.0324	7.5	GTX 3090
30% Synthetic Data	3	0.0392	0.0042	0.0697	0.0469	2	GTX 3090
Full Synthetic Data	3	0.0969	0.0159	0.1273	0.1159	1	GTX 3090

TABLE I: Performance metrics, training time and hardware used across experimental weeks.

*1) **Week 1**:* In the first phase, we implemented a baseline encoder-decoder model using a ResNet encoder and a GRU decoder. Despite 10 epochs of training, performance remained very low (BLEU-1: 0.0012, ROUGE-L: 0.0024, METEOR: 0.0012). The model showed strong underfitting, producing short, repetitive, and meaningless captions such as starting many outputs with ”Coa”.

Switching the encoder to VGG19, while initially leading to worse results when training all layers, showed clear improvements once early layers were frozen. Metrics improved significantly (BLEU-1: 0.0376, ROUGE-L: 0.0414, METEOR: 0.0182), although training curves revealed signs of overfitting.

Replacing the GRU decoder with an LSTM led to only minor improvements compared to the baseline, suggesting that decoder choice alone was not enough to overcome the overall model limitations.

*2) **Week 2**:* In the second phase, we transitioned to transformer architectures by using Huggingface's ViT (Vision Transformer) and GPT2 models. Direct evaluation without fine-tuning already showed better captions compared to earlier CNN-RNN models.

Fine-tuning both ViT and GPT2 together led to the best results achieved so far (BLEU-1: 0.1019, ROUGE-L: 0.1261, METEOR: 0.1079). Freezing only ViT or GPT2 individually resulted in weaker performance, emphasizing the importance of updating both vision and language components.

Attempts to use large pretrained multimodal models like Gemma 12B and ViT-Llama 1B/3B performed significantly worse than the fine-tuned ViT-GPT2, despite their size. Additionally, inference times were considerably longer for these larger models (e.g., over 11 hours for Gemma 12B), highlighting that fine-tuning on the task-specific dataset was more crucial than relying on model size alone.

*3) **Week 3**:* In the final phase, we generated synthetic food images using Stable Diffusion XL Turbo and ChatGPT-generated prompts, aiming to improve model generalization.

Training with 30–40% additional synthetic data slightly improved results compared to using only the original dataset, particularly on ROUGE-L and METEOR scores. However, the most surprising outcome came from training solely on synthetic data: the model achieved performance comparable to or better than when using the original real dataset (BLEU-1: 0.0969, ROUGE-L: 0.1273, METEOR: 0.1159).

This suggests that the original dataset's ground truth captions were too abstract and inconsistent for effective learning, while synthetic prompts provided clearer, more learnable supervision.

B. Qualitative Analysis

The following table compares caption generation quality across three model variants. We evaluate how different training approaches affect the models' ability to produce accurate and coherent food descriptions.

Ground Truth	VIT-GPT2 Pretrained	VIT-GPT2 Fine-Tuned	VIT Fine-Tuned
Matzo Ball Soup	a bowl of soup and a glass of wine	Bibimbap at Home (Baba Ghanoush) with Garlic-Lime Matzo Balls and Leeks (Baba Bar Bihonada)	a pot roast with garlic bread and herbs
Smoked Salmon with Black Pepper Potato Chips	two pieces of bread with cheese and tomatoes	Smoked Trout with Horseradish Cream and Citrus Caviar Filling (Figs au romaine) Grapefruit Fritters with Grapefruit Compote and Horseradish Cream Toasts Grapefruit Compote au rom	a pizza with basil and cheese on a plate
Bacon Swiss Burgers with Tomato and Avocado	a sandwich with lettuce, tomato, and cheese	”Mushroom Burgers with Bologna and Cheddar Sliders on Polenta ”””Tacos””” with Curry Barbecue Sauce and Green Onions ”””Onions ”””Onions ”””Onions ”””Onions ”””Onions ”””Onions	a hamburger with cheese and onions on a bun

TABLE II: Qualitative comparison of model predictions across different training approaches (Week 2)

Model comparisons show distinct generation styles. The pretrained VIT-GPT2 gives simple but accurate descriptions lacking detail (”a bowl of soup” for Matzo Ball Soup). Fine-tuning adds culinary terms but causes verbosity, errors (mixing ”Bibimbap” with ”Baba Ghanoush”), and repetition (”Onions” six times). The VIT-only model is consistent but overly generic, reducing complex dishes to basic types (”Smoked Salmon” becomes ”pizza”). Overall, fine-tuning boosts specificity at the cost of coherence, while the VIT-only model lacks enough nuance for meaningful food description.

The following table demonstrates how different levels of synthetic data affect caption generation quality.

Ground Truth	+30% Synthetic Data	100% Synthetic Data
Instant Pot Braised Lamb with White Beans and Spinach	- Meataf with Beans Goat and Beans Yog Sauce Yog Sauce Yog Sauce Yog Sauce Yog	ris fried with and sauce lemonah lemonah lemonah vinegar olive oil lemonah oil garlic herbs olive
Hibiscus Tea Sorbet	- Ice Ice with Ice,,,,,,,,,,,,,,,	and ice with and creamzzle ice cream strawberryousse ice creamzzle creamzzle creamzzle creamzzle cream

TABLE III: Qualitative comparison of synthetic data augmentation approaches (Week 3)

The synthetic data results show a clear pattern: while the 30% synthetic model produces repetitive but somewhat coherent outputs (e.g., ”Beans Yog Sauce” repeated), the 100% synthetic version generates nonsensical phrases (”creamzzle”, ”lemonah”) with complete loss of meaning. Both approaches fail to accurately capture dish details, suggesting synthetic food data requires better integration with real examples to maintain quality. The excessive repetition indicates models overfit to synthetic patterns without developing true understanding.

VI. Conclusion

This project explored multiple ways to improve image captioning, focusing on encoder-decoder architecture changes, different levels of text representation, and advanced optimization strategies. By modifying encoder types (like ResNet and VGG16), adjusting decoder complexity (such as LSTM variants), and switching from character-level to word-level tokenization, we achieved more efficient and expressive caption generation. Training improvements, including adaptive optimizers (Adam, AdamW), learning rate scheduling, teacher forcing, and automated hyperparameter tuning with Optuna, significantly stabilized training and reduced overfitting.

Transitioning to transformer-based architectures, especially combinations like ViT-GPT2, demonstrated a substantial leap in captioning quality over traditional CNN-RNN setups. Fine-tuning both vision and language components proved essential for achieving strong alignment between image features and generated text. However, experiments with larger multimodal models showed that simply increasing model size without targeted fine-tuning led to poor performance and excessive resource demands.

Synthetic data experiments revealed that moderate augmentation (e.g., 30–40%) improved model generalization slightly, especially in capturing new food concepts. However, full reliance on synthetic data degraded output quality, highlighting the ongoing importance of real-world examples for maintaining semantic accuracy and coherence.

Overall, the findings emphasize that successful image captioning depends on a careful balance between architecture choice, training stability, and data quality. Future work should focus on better integration of synthetic and real data, as well as exploring architectures that can adaptively balance generalization and specificity for diverse captioning tasks.

References

[1] L. Xu, Q. Tang, J. Lv, B. Zheng, X. Zeng, and W. Li, “Deep image captioning: A review of methods, trends and future challenges,” *Neurocomputing*, vol. 546, p. 126287, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231223004101

[2] L. Agarwal and B. Verma, “From methods to datasets: A survey on image-caption generators,” *Multimedia Tools and Applications*, vol. 83, no. 9, pp. 28 077–28 123, March 2024. [Online]. Available: https://doi.org/10.1007/s11042-023-16560-x

[3] M. Stefanini, M. Cornia, L. Baraldi, S. Cascianelli, G. Fiameni, and R. Cucchiara, “From show to tell: A survey on image captioning,” *CoRR*, vol. abs/2107.06912, 2021. [Online]. Available: https://arxiv.org/abs/2107.06912

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv:2010.11929v2 [cs.CV]*, 2021, google Research, Brain Team. [Online]. Available: https://arxiv.org/abs/2010.11929

[5] A. Kjærran, E. S. Bugge, and C. B. Vennerød, “Time series long short-term memory rnn,” arXiv:2105.06756v1 [cs.LG], 2021, tDT4173 - Assignment 1. [Online]. Available: https://arxiv.org/abs/2105.06756

[6] V. Surkov, C. Wendler, M. Terekhov, J. Deschenaux, R. West, and C. Gulcehre, “Unpacking sdxl turbo: Interpreting text-to-image models with sparse autoencoders,” *arXiv preprint arXiv:2410.22366v2*, 2024. [Online]. Available: https://arxiv.org/abs/2410.22366v2

[7] S. Goel, “Food ingredients and recipe dataset with images.” [Online]. Available: https://www.kaggle.com/datasets/pes12017000148/food-ingredients-and-recipe-dataset-with-images

[8] Hugging Face, “Evaluate metric: Bleu,” https://huggingface.co/spaces/evaluate-metric/bleu, 2023.

[9] Hugging face, “Evaluate metric: Meteor,” https://huggingface.co/spaces/evaluate-metric/meteor, 2023.

[10] Hugging Face, “Evaluate metric: Rouge,” https://huggingface.co/spaces/evaluate-metric/rouge, 2023.