EfficientNet and Transformer-Based Image Captioning with LLM Refinement

Deep Learning - COMP6826001

2702236925 Grace Birgitta Hadhinata

2702328831 Mirekel Tjoa

2702221841 Calista Paramitha Chandra

Computer Science Study Program
School of Computer Science
Universitas Bina Nusantara
5th semester
2025

EfficientNet and Transformer-Based Image Captioning with LLM Refinement

Image captioning is an important research area in both Computer Vision and Natural Language Processing. Recent Vision Language Models (VLMs) have achieved strong performance on standard benchmarks. However, they often produce generic captions and remain susceptible to visual hallucinations, especially in dynamic social media environments, where contextual relevance is critical. To address these limitations, this work proposed an integrated Object Detection (OD) - Large Language Model (LLM) pipeline for social media caption generation. The core captioning model uses a frozen EfficientNet-B0 Encoder for visual feature extraction and Transformer based model for autoregressive text generation. Due to computational restraints, the model was trained on a limited dataset from the COCO2017 dataset (approx. 10.102 Images). The architecture demonstrated a performance in this low-resource setting achieving a CIDER score of 0.490 and a ROUGE-L score of 0.394, indicating the model of competence in generating relevant semantic captions for relevant objects. The LLM transforms the core model's output into captions that are contextually relevant and linguistically expressive, specifically formatted with relevant hashtags for social media engagement, without impacting the core model's training and evaluation.

Keywords—Image Captioning, EfficientNet, Transformer, Deep Learning, Vision Language Models, Large Language Models, Object Detection, Social Media Caption

I. INTRODUCTION

Image captioning has emerged as an important research area in both Computer Vision and Natural Language Processing with the goal of combining visual understanding and natural language processing. [1] Image captioning requires the comprehension of visual information at a semantic level and the understanding to translate the information into a coherent description. [2] captioning has evolved extensively progressing from simple encoder-decoder architecture to advanced models like Vision Language Models (VLM) that leverages CNNs (EfficientNet) for feature extraction and transformer-based models to generate more contextually rich descriptions. These breakthroughs have enabled the formulation of more descriptive and detailed captions. However, the nature of the digital environment particularly social media introduces additional requirements that extend beyond accurate descriptions.

The nature of the digital environment, especially social media, fundamentally changes the requirements in the process of creating a description or a caption of an image. Posts from Instagram and X (formerly Twitter) rely heavily on captions to convey the post context, sentiment, and appeal to maximize user interaction.[3] Traditional VLM approach becomes less consistent since its approach is summarizing the whole visual input image, generating general description but missing important context, and susceptible to visual hallucinations, producing texts that doesn't reference the image.

To address this issue, we proposed an integrated Object Detection (OD) - Large Language Model (LLM) pipeline. For the OD architecture, we will be applying EfficientNet to effectively detect relevant objects in dynamic social media images. Then information will be extracted from these objects and used to guide caption refinement through an LLM. [4]LLM transform object-centered prompts into captions that are factually grounded and linguistically rich, leveraging its contextual ability to produce more natural text.

The main contribution from this research lies in the development of an object-aware captioning image pipeline that integrates LLMs. By explicitly grounding caption generation on detected objects, the proposed approach addresses key limitations of conventional Vision–Language Models in social media contexts, including generic descriptions and visual hallucinations. This integration enables the generation of captions that are not only factually grounded but also contextually relevant and linguistically expressive, making them better suited for further contextual and linguistic refinement by the Large Language Model (LLM).

II. RELATED WORK

A. EfficientNet

CNNs have achieved remarkable success in visual recognition tasks, primarily through increased model capacity via network depth, width, or input resolution. Early architecture such as VCG and ResNet demonstrated that deeper networks generally lead to improved accuracy, while later works explored wider network and higher resolution inputs to further enhance representational power. However, these strategies came along with a demerits of diminishing accuracy gains relative to computational cost as they are typically applied along a single dimension and relied heavily on empirical tuning.

[5] To address these limitations, Tan and Le proposed EfficientNet, which derived from a principled compound scaling strategy that uniformly scales network depth, width, and resolution in a coordinated manner. This introduces a single compound coefficient that governs how additional computational resources are allocated across all three dimensions. By explicitly accounting for the interdependence among these dimensions, it achieves substantially improved accuracy-efficiency trade-offs compared to conventional single-dimension scaling approaches. Building on this scaling principle, Tan & Le employed neural architecture search (NAS) to derive this baseline model, EfficientNetB0, jointly optimized for

classification accuracy and computational efficiency. Right now, it serves as the foundation for the entire EfficientNet family (B1 – B7). Despite its compact design of approximately 5.3 million parameters and 0.39 billion FLOPs, it achieves strong ImageNet performance, demonstrating high accuracy relative to its computational cost. Owing to its balanced architecture, EfficientNet-B0 can be systematically scaled using the compound scaling strategy to generate higher-capacity variants without architectural redesign.This property makes them highly versatile.

It's commonly employed both as a lightweight standalone model for resource-constrained settings and as a scalable backbone for transfer learning across diverse computer vision tasks, such as, [6] image classification in waste-sorting task, [7] computer-assisted diagnosis of breast cancer in medical world, to video deepfake detection, combined with vision transformers, to [8] video deepfake detection, combined with vision transformers.

B. *Decoder Transformer*

Early image captioning approaches primarily combined CNN-based visual encoders with recurrent decoders such as LSTMs. Although effective, these models were limited by sequential processing and difficulty in modeling long-range dependencies. Attention mechanisms improved visual grounding but didn't fully address these limitations inherent to recurrent architecture.

Transformer decoders overcome these issues by replacing recurrence with masked self-attention, enabling parallel training and more effective modeling of global linguistic context. In vision-language tasks, Transformer decoders are commonly conditioned on visual features via cross-attention, allowing generated tokens to attend explicitly to image representations from the encoder. This mechanism facilitates fine-grained alignment between visual regions and linguistic elements, leading to more accurate and descriptive captions.

Recent studies have increasingly adopted decoder-centric Transformer architectures paired with pretrained or fixed lightweight feature extractors. This modular design reduces computational complexity and improves generation, particularly when pretrained visual backbones are employed. Furthermore, it offers architectural flexibility that supports extensions such as multi-head cross-attention, deeper decoding stacks, and integration with external linguistic priors. [9].

C. *Mistral*

[10] Mistral refers to a family of Large Language Models (LLMs) developed by Mistral AI, designed to achieve high performance with significantly improved efficiency compared to conventional dense transformer models of similar capability. The most notable early model is Mistral-7B, which demonstrated that strong reasoning and generation performance can be achieved with fewer parameters, optimized attention mechanisms, and architectural refinements rather than sheer scale.

Its presence aims to answer the question; how far we can push transformer-based language models through architectural efficiency rather than parameter count. By its code architecture, mistral models follow a decoder-only transformer architecture, similar in principle to GPT-style models. It does its job by incorporating sliding window attention and grouped-query attention to reduce computational and memory costs while maintaining competitive performance.

D. *Metrics*

1) BLEU (Bilingual Evaluation Understudy)

[11] BLEU evaluates image captions by measuring n-gram precision, the proportion of N-grams in a generated caption that also appear in one or more reference captions, to be precise. To prevent overly short captions from achieving artificially high precision, BLEU applies a brevity penalty (BP). The final score is computed as the geometric mean of modified N-gram precisions up to order N, scaled by the brevity penalty, as shown in Eq. 1 below.

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right),$$

$$BP = \begin{cases} 1, & c \geq r \\ exp\left(1 - \frac{r}{c}\right), & c \leq r \end{cases}$$

Equation 1. BLEU Score Formula

2) METEOR (Metric for Evaluation of Translation with Explicit Ordering)

[12] METEOR evaluates captions using unigram alignment between candidates and reference sentences while accounting for unigram precision (P), unigram recall (R), stemming, and antonym matching. A penalty term Pen is applied in the metric to discourage fragmented alignments, as shown in Eq. 2 below.

$$F_{mean} = \frac{10PR}{R+9P},$$

$$\text{METEOR} = F_{mean} \cdot (1 - P_{en})$$

Equation 2. METEOR Score Formula

3) ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation)

[13] ROUGE-L measures caption quality based on the Longest Common Subsequence (LCS) between the generated caption and a reference caption. The LCS captures sentence-level similariy while preserving word order and allowing non-consecurive matches. That precision and recall score are combined into an F-measure, as shown in Eq. 3 below.

$$P_{LCS} = \frac{LCS(X,Y)}{|X|},\ R_{LCS} = \frac{LCS(X,Y)}{|Y|}$$

$$\text{ROUGE-L} = \frac{(1+\beta^2)P_{LCS}R_{LCS}}{R_{LCS}+\beta^2 P_{LCS}}$$

Equation 3. ROUGE-L Score Formula

4) CIDEr (Consensus-based Image Description Evaluation)

[14] CIDEr is specifically designed for image captioning and evaluates how well a generated caption matches the consensus of multiple human references. It represents each caption using TF-IDF weighted n-gram vectors and compute cosine similarity between the candidate and reference caption. The final score averages over multiple N-gram order, as shown in Eq. 4 below.

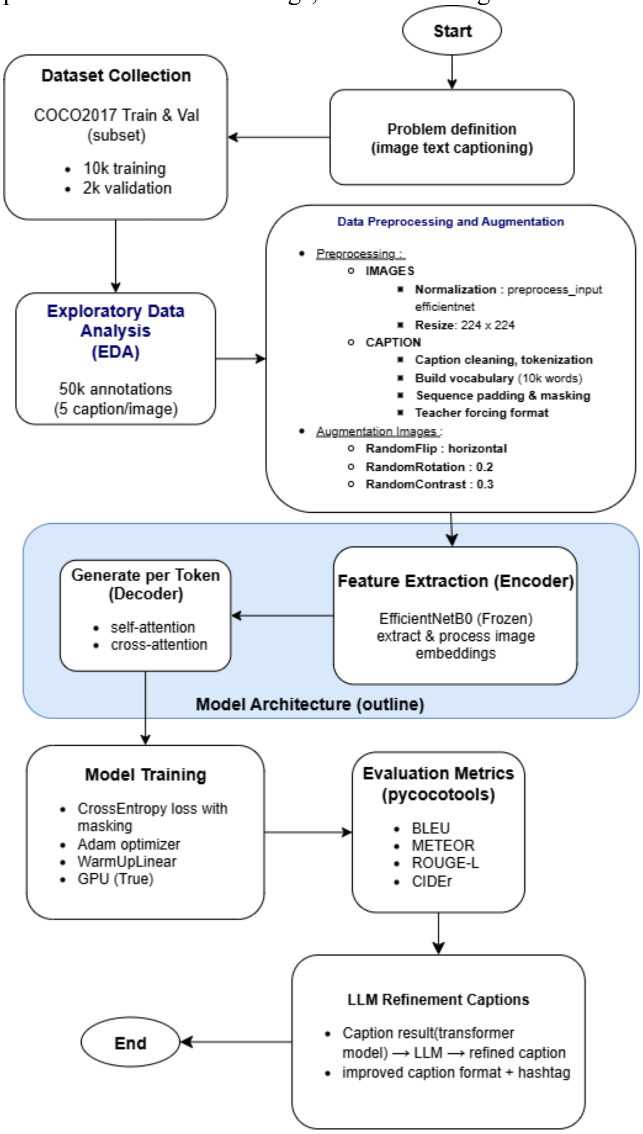
$$CIDEr_n(c,S) = \frac{1}{|S|}\sum_{s\epsilon S}\frac{g_n(c)\cdot g_n(s)}{||g_n(c)||\ ||g_n(s)||},$$

Equation 4. CIDEr Score Formula

III. Methodology

A. *Thinking Framework*

This experiment consists of seven main stages, following the problem definition, ranging from general Explanatory Data Analysis (EDA) to producing a caption model. The Large Language Model (LLM) is used for post-generation and does not affect the training and evaluation of the caption model results. The refined captions from the Large Language Model (LLM) are used to enhance the caption results, specifically to create engaging social media post captions with relevant hashtags, as shown in Fig. 1 below.



Start
Problem definition
(image text captioning)
Dataset Collection
COCO2017 Train & Val (subset)
10k training
2k validation
Exploratory Data Analysis (EDA)
50k annotations
(5 caption/image)
Data Preprocessing and Augmentation
Preprocessing :
IMAGES
Normalization : preprocess_input efficientnet
Resize: 224 x 224
CAPTION
Caption cleaning, tokenization
Build vocabulary (10k words)
Sequence padding & masking
Teacher forcing format
Augmentation Images :
RandomFlip : horizontal
RandomRotation : 0.2
RandomContrast : 0.3
Feature Extraction (Encoder)
EfficientNetB0 (Frozen)
extract & process image embeddings
Generate per Token (Decoder)
self-attention
cross-attention
Model Architecture (outline)
Model Training
CrossEntropy loss with masking
Adam optimizer
WarmUpLinear
GPU (True)
Evaluation Metrics (pycocotools)
BLEU
METEOR
ROUGE-L
CIDEr
LLM Refinement Captions
Caption result(transformer model) ⟶ LLM ⟶ refined caption
improved caption format + hashtag
End

Fig. 1 Thinking Framework

The stages start from the data collection stage from the very large, complete COCO2017 dataset, followed by image preprocessing and text captions from the dataset, then visual feature extraction using EfficientNetB0. The results of this feature extraction will be input for the Transformer encoder-decoder architecture to generate candidate tokens for composing captions from images. The output of the model is a short caption that describes the contents of the image. Then Caption is refined by LLM to improve the wording for social media, along with hashtags. Evaluation is carried out using standard metrics BLEU, METEOR, ROUGE-L, and CIDEr from pycocotools.

B. *Dataset*

- **Dataset Source :** COCO2017 is a public dataset with train2017 and val2017 formats accessible via JSON for captioning and object detection. This project uses a subset of the dataset due to device limitations: 10.102 train2017 images for training and 2.000 val2017 images for validation (this setting follows a low-resource training scenario). Each image has approximately five annotated captions describing the objects and scenes in the image. However, for training, the format is adjusted to have one caption per image, allowing the same image to appear five times in pairs with different captions.

- **Dataset characteristics:** COCO2017 contains numerous indoor and outdoor objects and scenes, as well as a variety of image angles, suitable for training image captioning models. The average caption length in the training dataset is between 6 and 20 words, as shown in Fig. 2.

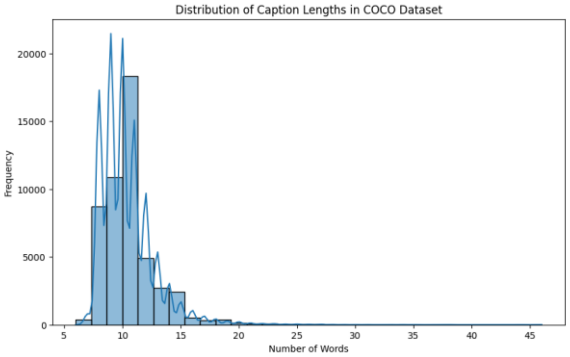


Fig 2 Distribution caption length in training data

C. *Data preprocessing and Augmentation*

1) *Image Preprocessing*

The image was resized to 224 x 224 pixels, consistent with the standard input of the EfficientNetB0 model in RGB format (channels=3). Normalization was performed using EfficientNet's built-in preprocessing (tf.keras.applications.efficientnet.preprocess_input).

2) *Caption Preprocessing*

Caption text preprocessed with custom standardization, including :

- Lowercasing to reduce word variation due to capitalization, thus reducing vocabulary size.
- Removing irrelevant symbols using regular expressions other than those used in special tokens (e.g., "<", ">").
- Text normalization by removing irrelevant characters in captions so the model only learns clean and consistent words.

3) *Vocabulary Construction*

Vocabulary creation is limited to 10.000 words after caption standardization to build a list of all unique words (total words in this experiment: 9.521 words). The vocabulary maps words to numeric indices so it can be processed by the transformer model. TextVectorization is used as a tokenizer and vocabulary builder with the following vocabulary-building steps:

1. Adding special SOS and EOS tokens, <start> to mark the start of the caption and <end> to mark the end of the caption, ensures structure consistency and allows the decoder to know when to start and stop generating tokens for captions.
2. Tokenization and Sequence Padding in the TextVectorization layer maps words to integers, normalizes the sequence length to a maximum length, and automatically applies padding if the output sequence is less than the maximum length.
3. Get Vocabulary and Index Lookup mapping

   The vocabulary is obtained from the get vocabulary vocab process, and the first four indexes, respectively, contain the padding token (""), the unknown word token ([unk]), <start>, and <end>. Then, index lookup is used to retrieve words from the model's output (the index sequence is translated into a sentence).

4) *Data Augmentation*

Augmentation to the training data to improve the reliability of the model:

- Random horizontal flip,
- Random rotation 0.2 (rad)
- Random contrast 0.3 (30%)

D. Model Architecture

1) Encoder

Extracting and representing information from the input (image) into a numerical representation (embedding).

a. **Visual Feature Extraction (EfficientNetB0)**

The feature extractor uses EfficientNetB0 with pretrained ImageNet model weights. All EfficientNet layers are made non-trainable (frozen) to save computation and prevent performance degradation due to the relatively small training data for training the pretrained model. The final classification layer (fully connected head) is omitted; all weights are frozen during training. The output, an image feature map, is then projected into a sequence of visual embeddings as a global representation.

b. **Transformer Encoder Block**

Processes visual embeddings to learn global spatial correlations between image parts or processes projections of visual feature sequences. It consists of:

- Multi-head self-attention: Models for global dependency relationships between image features
- Feed forward network (FFN): Nonlinear transformation of each token
- Layer normalization dan dropout: Maintain training stability and prevent overfitting

2) Decoder

Produces autoregressive captions based on the encoder's visual representation and the previously generated token sequence.

a. **Positional Embedding**

The caption dataset is processed as a sequence of discrete tokens. Each vector is mapped to a model dimension (D_MODEL) via the token embedding to preserve word order information.

b. **Transformer Decoder Block**

Accepts two inputs: the previous caption token embedding and the output image representation from the visual encoder. The decoder consists of:

- Masked multi-head self-attention : processes masked self-attention (look-ahead mask) to prevent future words (avoiding data leakage) and to ignore pad tokens during training.
- Cross attention (encoder-decoder) : encoder-decoder attention connects the caption (text) representation with the visual features of the image.
- Feed forward network (FFN) : generates token representations before predicting the next word.

E. Training Setup

- **Hyperparameters**

Optimizer	Adam
Model Dimension	512
Feed Forward Dim	2048
Num Layers	2
Max Caption Length	20
Vocab Size	10.000
Batch Size	64
Dropout Rate	0.3
Learning Rate (LR)	(initial) 1e-4

- **Training Configuration and Warm Up**

Training was conducted using an NVIDIA GeForce RTX 4060 Laptop GPU with 8GB of VRAM, enabling dynamic memory allocation to prevent out-of-memory (OOM) events during the training process.

WarmUpLinear was used to stabilize the initial training process with a combination of warm-up and linear decay, preventing unstable parameter updates while the model weights were still randomly initialized. For 1,000 training steps, the learning rate increased linearly to optimize stability. Afterward, the learning rate gradually decreased to (1x10-5). Scheduling useful for balancing training stability and convergence due to the limited training dataset. [15]

- **Checkpoint and Early Stopping**

  Early stopping was used to prevent overfitting with a patience of 10 epochs, saving the ".keras" model with the best accuracy in the final epoch.

F. *Evaluation Metrics*

The following metrics calculate the overlap of n-grams or word sequence structures between generated captions and ground truth captions (from datasets) with different focuses.
(pycocotools)

TABLE I. EVALUATION METRICS

Metrics	Explanation
BLEU	(Bilingual Evaluation Understudy) Measures the precision of n-grams between the generated captions (candidate captions) and the human reference captions. Evaluate accuracy by considering n-grams (1 to 4) that are sensitive to word similarity.
METEOR	(Metric for Evaluation of Translation with Explicit Ordering) Combines precision and recall between predicted captions and reference captions, taking into account the similarity of the generated words' meanings (sensitive to word semantics).
ROUGE-L	(Recall-Oriented Understudy for Gisting Evaluation) Calculates recall and precision of LCS (Longest Common Subsequences) by considering the similarity of sentence structure between predicted and reference captions.
CIDEr	(Consensus-based Image Description Evaluation) Measuring the similarity between generated captions and reference captions using TF-IDF on the n-grams of the entire dataset (giving higher weight to informative n-grams and lower weight to frequently occurring words).

IV. IMPLEMENTATION & RESULT

The COCO2017 dataset is very large, so there's a Google Colab. ipynb file to retrieve the dataset via the Python JSON library to download the entire zipped dataset. The train dataset only uses a subset of 50,000 annotations from the total 591,753 annotations in train2017, while val2017 only uses 2,000 images for evaluation.

The implementation of this project began with preparing dependencies for Tensorflow version 2.17.0, which is compatible with cuDNN 12.3.52, allowing for the use of a GPU environment for model training on a local device. Then, a compatible conda environment was created that could detect GPUs on Ubuntu 24.04.03 LTS, as Windows is not compatible with the integration of cuDNN and Tensorflow versions.

The Transformer encoder-decoder model is wrapped in the CaptionTrainer class to customize loss and accuracy calculations. The model was trained with a defined 30 epochs (early stopping at epoch 21) with a total time of approximately 3 minutes per epoch and a total training time of approximately 1 hour and 17 minutes. Model performance was evaluated using BLEU-1 to BLEU-4, METEOR, ROUGE-L, CIDEr, which shows the model's performance from the quality of its captions. The metric selection is based on the COCO Caption Evaluation Protocol (pycocotools) to measure the quality of the image captioning model. The quantitative evaluation results can be seen in the following evaluation

TABLE II. EXPERIMENT RESULTS

Metric	Description	Score
BLEU-1	Unigram precision	0,562
BLEU-2	Bigram precision	0,382
BLEU-3	Trigram precision	0.245
BLEU-4	4-gram precision	0.158
METEOR	Synonym matching, stemming, recall	0.189
ROUGE-L	Longest Common Subsequence	0.394
CIDEr	Consensus TF-IDF similarity	0.490

The metrics results, which provide numerical results for the model's performance, are shown in Table II. The model performed quite well across all COCO metrics. The CIDEr score (0.49) indicates a fairly good match between the generated caption and the reference caption (the dataset reference) based on TF-IDF similarity. The decreasing BLEU score pattern from BLEU-1 to BLEU-4 indicates the model consistently captures short tokens (unigrams) than long n-gram structures. Meanwhile, the METEOR score (0.189) indicates a fairly good match (synonymous equivalence) despite differences in phrase variation between the predicted and actual (reference) captions. ROUGE-L (0.394) indicates that the model can capture some word sequence structures, although not optimally.

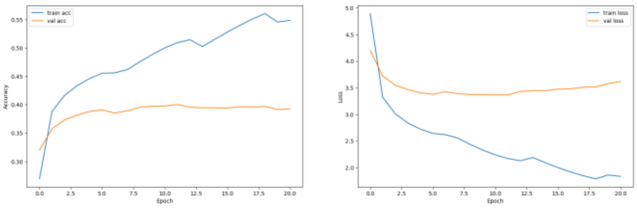
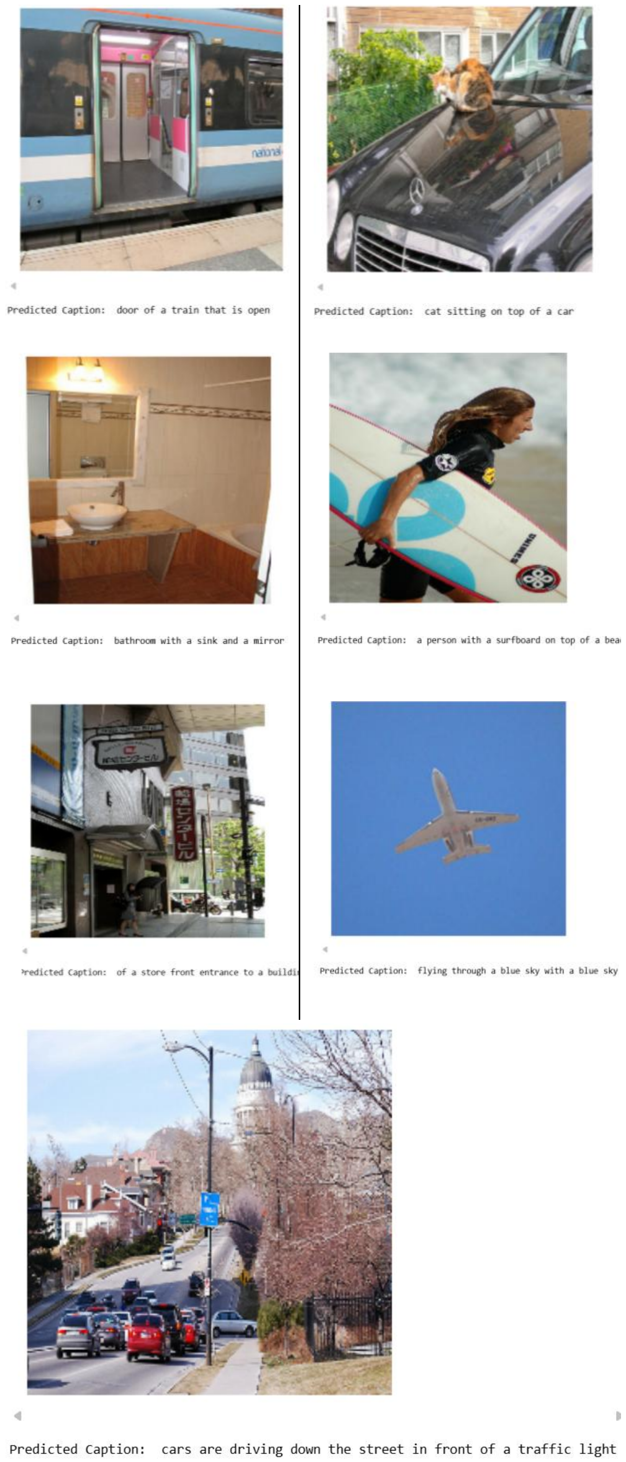


Fig 3 Plot Accuracy and Loss

The plot in Fig 3 shows an increase in training accuracy and a decrease in loss, but relatively stagnant accuracy and validation loss, especially above epoch 5. This pattern indicates a tendency toward overfitting related to the limited training dataset compared to the complexity of the Transformer model, which requires a large amount of training data.



```
Predicted Caption:  door of a train that is open
Predicted Caption:  cat sitting on top of a car
Predicted Caption:  bathroom with a sink and a mirror
Predicted Caption:  a person with a surfboard on top of a bea
redicted Caption:  of a store front entrance to a buildi
Predicted Caption:  flying through a blue sky with a blue sky
Predicted Caption:  cars are driving down the street in front of a traffic light
```

Fig 4 Sample Caption Prediction

Fig. 4 shows seven caption samples showing the predicted caption results from the validation set. Based on direct observation, the model can generate semantic captions for clearly visible objects (their shapes are unambiguously similar to other objects). Of these samples, five samples, representing the majority of images, generated captions that were contextually relevant, indicating the model can identify clearly visible objects and the spatial relationships between them. The remaining samples represent prediction results indicating the decoder failed to determine the initial token, resulting in "of" at the beginning of the sentence. The repeated words "..blue sky with a blue sky" in other images indicate semantic repetition due to the inability to fully capture visual variations.

As a final implementation, the image captioning model was deployed using Streamlit in a web format to load latest epoch model (epoch21.keras) from HuggingFace and integrate it with tokens for LLM API access(*mistral-small-3.1-24b-instruct:free* via OpenRouter). The resulting caption generation from the model will be shown and used as a context for refining the writing style for social media. Therefore, the LLM integration does not impact model training and evaluation.

V. Discussion & Limitations

A. Quantitative Metrics

Based on the accuracy and loss plots, the neural network exhibits an overfitting pattern, where training accuracy improves while validation performance stagnates early. This aligns with the theory that high-capacity neural networks easily learn specific patterns. This overfitting may occur because the dataset size (subset) of this experiment is limited for training purposes, compared to other studies that use hundreds of thousands of images.[16]

The results of evaluation metrics using BLEU, METEOR, ROUGE-L, and CIDEr show performance consistent with the characteristics of Transformer-based captioning models on a limited subset of datasets, as the performance of captioning models generally improves with an increase in model size as well as dataset size. More specifically, they show that larger models benefit more when the dataset size is above a certain threshold; with sufficient data, the performance continues to improve as the model size increases.

- **BLEU** : the BLEU-1 to BLEU-4 scores decrease sequentially, indicating the model is more accurate in predicting short token sequences (unigrams) because it can achieve a score >0.50, which is considered a fluent candidate sentence compared to long-range dependencies. This is because the higher the n-gram, the more accurate the model should be in predicting longer word sequences consistently, in accordance with theory. [17]
- **METEOR** : a relatively low score (0.189) indicates the model's limitations in capturing semantic matching (stemming, synonyms), which mean the model's limitations in understanding the semantic similarity between the generated caption and the reference caption.
- **ROUGE-L** : a score of 0.394 indicates that the generated sentence structure can construct LCS (Longest Common Subsequence) word sequences with the reference caption but has not yet successfully captured complex word sequences.

- **CIDEr** : Compared with previous research using the full dataset from COCO Train (approximately 113 thousand images and 591 thousand captions), which is usually able to achieve 0.9-1.2. This is because it only uses a subset of the train dataset, so the score of 0.490 shows a fairly good match between the predicted caption and the reference caption (ground truth) based on the TF-IDF weight, because the difference in the scores depends on the subset size.

B. Qualitative Observations

Based on direct analysis of the model's caption prediction results, the generated captions are quite relevant to the validation image provided for objects that are clearly visible, unambiguously resemble other objects, and do not overlap with other objects. However, the model is limited in predicting captions with opposite objects and often only generates generic captions if the object is not present in the training data.

Furthermore, the model also produces inaccurate captions or predicts phrases or words for objects that are not actually present in the input image (hallucinations), indicating visual language bias in the captioning model. Typically, the predicted words are those that appear most frequently and are learned in the training dataset, because standard approaches to image captioning are known to hallucinate objects that co-occur frequently. Various failure cases and shortcuts exploited by deep models, especially in vision and language tasks such as image captioning and visual question answering (VQA), in the form of object hallucinations, force the model to rely on language priors, focusing on the background, action bias, gender bias, etc. [18]

C. Limitations

This experiment encountered several limitations related to computational constraints during training. The training process limitation, namely GPU memory capacity, necessitated training using a subset of the COCO2017 dataset. From an architectural perspective, it relied solely on global visual features from the frozen EfficientNetB0 visual feature extractor. This reduced computational load and the risk of overfitting due to the small training dataset, but its ability to capture object details was limited. This limited dataset size limited the model's generalization ability, and the unfreezing of EfficientNetB0, which could be used to make the visual backbone effective when trained on full-scale datasets.

VI. Conclusion & Future Work

A. Conclusion

This experiment successfully implemented an image captioning model utilizing a frozen EfficientNetB0 encoder for visual feature extraction and a transformer encoder for autoregressive text generation. The architecture was tested on a subset of the COCO 2017 dataset with approximately 10.102 training images due to computational constraints.

The model itself exhibited a respectable performance for a low-resource scenario, achieving a CIDEr score of 0.490 and a ROUGE-L score of 0.394. This demonstrates the model's competence in generating semantically relevant captions for clearly visible objects and their spatial relationships, even though not optimal.

The primary limitations observed were a tendency toward overfitting due to the high-capacity transformer model being trained on a limited subset of the dataset. Qualitatively, the model occasionally exhibited visual hallucinations and semantic repetition. A key aspect of this work was the successful integration of LLM for post-generation refinement of captions into engaging, social media-ready text with hashtags, without impacting the core model's training.

B. Future Work

Future research/experiment should focus on these main areas:

- Scaling Data and Architecture, which includes training the model using the full dataset to improve generalization and consider unfreezing the EfficientNetB0 encoder to capture finer visual details.
- Mitigating Hallucinations, which includes integrating an explicit object detection layer to ground the model's predictions more accurately, which is expected to significantly reduce object hallucinations, a common problem in end-to-end Vision-Language Models (VLMs).
- Refining Linguistic Output, which includes experimenting with advanced decoding strategies to eliminate repetition and further refine the LLM integration for more nuanced stylistic control over the final social media captions.

References

[1] A. sharma, H. Singh, and M. Pant, "Pixels to Prose: Understanding the Art of Image Captioning," 2025. doi: 10.2139/ssrn.5351410.

[2] A. A. E. Osman, M. A. W. Shalaby, M. M. Soliman, and K. M. Elsayed, "A Survey on Attention-Based Models for Image Captioning," 2023. [Online]. Available: www.ijacsa.thesai.org

[3] X. Zhang *et al.*, "SoMeLVLM: A Large Vision Language Model for Social Media Processing," in *Findings of the Association for Computational Linguistics ACL 2024*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2024, pp. 2366–2389. doi: 10.18653/v1/2024.findings-acl.140.

[4] L. Celona, S. Bianco, M. Donzella, and P. Napoletano, "Improving image captioning descriptiveness by ranking and LLM-based fusion," *Neural Comput Appl*, vol. 37, no. 32, pp. 27279–27299, Nov. 2025, doi: 10.1007/s00521-025-11672-x.

[5] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," Sep. 2020, [Online]. Available: http://arxiv.org/abs/1905.11946

[6] Z. Feng, J. Yang, L. Chen, Z. Chen, and L. Li, "An Intelligent Waste-Sorting and Recycling Device Based on Improved EfficientNet," *Int J Environ Res Public Health*, vol. 19, no. 23, p. 15987, Nov. 2022, doi: 10.3390/ijerph192315987.

[7] P. R. Oza, P. Sharma, and S. Patel, "A Transfer Representation Learning Approach for Breast Cancer

Diagnosis from Mammograms using EfficientNet Models," *Scalable Computing: Practice and Experience*, vol. 23, no. 2, pp. 51–58, Aug. 2022, doi: 10.12694/scpe.v23i2.1975.

[8] D. A. Coccomini, N. Messina, C. Gennaro, and F. Falchi, "Combining EfficientNet and Vision Transformers for Video Deepfake Detection," 2022, pp. 219–229. doi: 10.1007/978-3-031-06433-3_19.

[9] M. Cornia, M. Stefanini, L. Baraldi, and R. Cucchiara, "Meshed-Memory Transformer for Image Captioning." [Online]. Available: https://github.com/aimagelab/

[10] A. Q. Jiang *et al.*, "Mistral 7B," Oct. 2023, [Online]. Available: http://arxiv.org/abs/2310.06825

[11] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation."

[12] S. Banerjee and A. Lavie, "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments."

[13] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries."

[14] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based Image Description Evaluation," Jun. 2015, [Online]. Available: http://arxiv.org/abs/1411.5726

[15] A. Defazio, A. Cutkosky, H. Mehta, and K. Mishchenko, "Optimal Linear Decay Learning Rate Schedules and Further Refinements," 2024. [Online]. Available: httpsarxiv.orgabs2310.07831

[16] C. Rohlfs, "Generalization in neural networks: A broad survey," *Neurocomputing*, vol. 611, p. 128701, Jan. 2025, doi: 10.1016/j.neucom.2024.128701.

[17] A. de S. Inácio and H. S. Lopes, "Evaluation metrics for video captioning: A survey," *Machine Learning with Applications*, vol. 13, p. 100488, Sep. 2023, doi: 10.1016/j.mlwa.2023.100488.

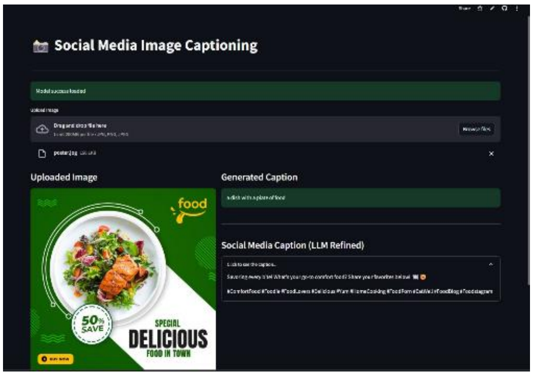
[18] A. F. Biten, L. Gomez, and D. Karatzas, "Let there be a clock on the beach: Reducing Object Hallucination in Image Captioning," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, IEEE, Jan. 2022, pp. 2473–2482. doi: 10.1109/WACV51458.2022.00253.

APPENDIX

[1] Team Contribution Statement

Name	Contribution
Grace Birgitta H.	Data Collection, Model Development, Deployment, Demo Apps, Report (Chapter III, IV, V)
Mirekel Tjoa	Report(Chapter I, Abstract, Demo Apps)
Calista Paramitha Chandra	Report(Chapter II, Chapter VI)

[2] Screenshots Apps



Social Media Image Captioning

Model success loaded

Upload image

Drag and drop file here

poster.jpg

Uploaded Image

Generated Caption

a dish with a plate of food

Social Media Caption (LLM Refined)

Click to see the caption...

Savoring every bite! What's your go-to comfort food? Share your favorites below! 🍽️ 😋

#ComfortFood #Foodie #FoodLovers #Delicious #Yum #HomeCooking #FoodForm #EatWell #FoodBlog #Foodstagram

[3] Code Snippets and links

- Github repository : https://github.com/gracebirgita/image-captioning-/tree/main
- Demo Video : https://youtu.be/WTf8PXiUMZA?si=qf4eXciwhSufqJ8X
- Apps Link : https://social-media-captioning.streamlit.app/