Machine Learning Approaches for Image Caption Generation: A Comparative Study

Team 4

Bhavya Anjali Potturi (G01462716)

Nidhi Janivara Somashekar(G01445948)

Manisha Suguru(G01478492)

AIT 736 – DL1 (Spring 2024), Applied Machine Learning

George Mason University

Dr. Emanuela Marasco

April 11, 2025

CHAPTER 1
INTRODUCTION

1.1 Objective

Image captioning is the task of automatically generating textual descriptions for given images. It's a challenging problem that merges techniques from both Computer Vision, which extracts relevant features from the image, and Natural Language Processing (NLP), which generates a descriptive sentence. The primary objective of image captioning systems is to "understand" the visual content of an image and to produce a coherent and meaningful sentence that accurately portrays the objects, actions, and context within that image. This is typically achieved using an encoder-decoder architecture, where the encoder processes the image, and the decoder generates the caption.

1.2 Introduction

Image Captioning is a challenging interdisciplinary task at the intersection of computer vision and natural language processing. It involves automatically generating textual descriptions for a given image. This task has numerous applications, including aiding visually impaired users, improving image search through better indexing, and enhancing social media automatically by generating alt texts.

With the advancements in deep leaning, especially in Convolution Neural Networks (CNNs) and sequence models like LSTM and Transformers, it is now possible to achieve highly accurate and fluent captions. This Project explores various combinations of CNN encoders and language decoders to determine the most effective approach.

The Models are trained and tested on the Flickr8k dataset, which includes 8,000 images paired with five human-written captions each. The evaluation of these models is based on quantitative metrics like BLEU scores and qualitative analysis of the generated captions. Image captioning is the task of generating textual descriptions for images. This task requires a system to understand the visual content, including objects, actions, and context, and then produce grammatically correct and semantically relevant sentences.

1.3 Motivation

Image captioning is driven by its potential to significantly enhance image accessibility, particularly for visually impaired users who benefit from textual descriptions. The technology enables automatic content tagging in extensive image databases, streamlining image search and organization. Furthermore, it contributes to improved search engine optimization by providing descriptive text that boosts image visibility in search results. Social media automation is another key motivator, as image captioning can generate descriptions for images, saving time and improving content accessibility. Beyond these applications, image captioning plays a vital role in advancing AI-powered systems. For instance, it aids virtual assistants in understanding and describing images and enhances surveillance systems by automating scene descriptions. Additionally, it empowers robots with the ability to interpret visual information. Ultimately, the core motivation lies in bridging the gap between visual and linguistic understanding, making images more informative and usable across various contexts.

1.4 Scope for the Work

This project focuses on developing and implementing various image captioning models. The primary aim is to explore different architectural combinations, specifically pairing CNN encoders with text decoders. CNN encoders like DenseNet, ResNet, and ViT are used to extract visual features from images. These encoders are combined with text decoders such as LSTM, Transformer, and GPT-2 to generate descriptive captions. The project's scope includes evaluating and comparing the models based on caption quality, training efficiency, and language fluency. The findings from this work will provide insights into the strengths and weaknesses of different model architectures for the image captioning task.

CHAPTER 2

PROJECT OVERVIEW

2.1 Description of the project

This project delves into the development and evaluation of various image captioning models. The core approach involves combining Convolutional Neural Networks (CNNs) as encoders to extract visual features from images, followed by leveraging text decoders like Long Short-Term Memory (LSTM) networks, Transformers, or GPT-2 to generate descriptive captions.

The project investigates the performance of different CNN architectures, including DenseNet and ResNet, as encoders. The workflow encompasses several key stages:

2.1.1 Flow Diagram

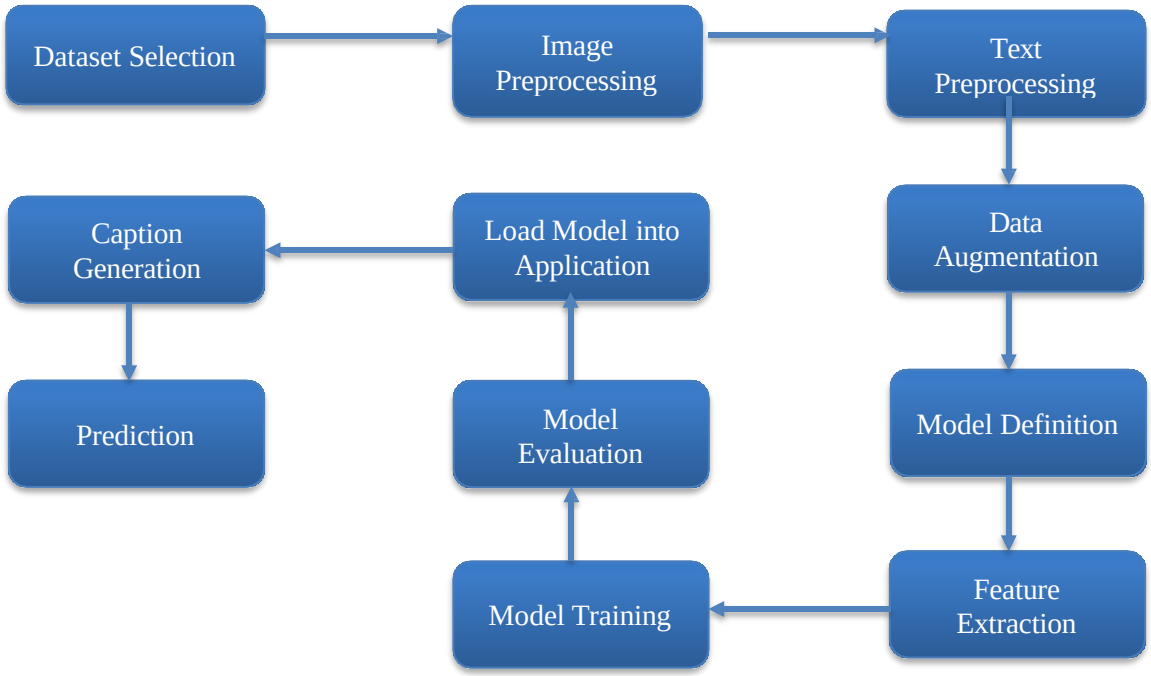


Fig: 2.1.1 Model Flowchart

The Process begins with ***Dataset Selection***, where a curated collection of images and their corresponding textual descriptions, such as the 8k Flickr dataset, is chosen. Following this, the data undergoes crucial preprocessing stages. ***Image Preprocessing*** involves preparing the visual input by resizing and normalizing the images. Simultaneously, ***Text Preprocessing*** is performed on the captions, including tokenization, lowercasing, handling punctuation, and adding special start and end tokens to define the caption boundaries. ***Data Augmentation*** techniques can be applied to both images and text to increase the training data's diversity and improve the model's robustness.

The next critical phase is ***Model Definition***, where the architecture of the image captioning model is established. This includes Feature Extraction using a Convolution Neural Network (CNN) encoder, such as DenseNet201, to extract meaningful visual features from the input images. These features are then fed into a decoder, like a Long Short-Term Memory (LSTM) network, which is defined to generate the caption. During ***Model Training***, the model learns the complex relationships between the visual features and the textual descriptions within the training

dataset. The model's performance is then assessed in the ***Model Evaluation*** stage using appropriate metrics like BLEU scores to quantify the quality and relevance of the generated captions. Finally, the trained model is prepared for deployment in the ***Load Model into Application phase,*** enabling it to perform ***Caption Generation (Prediction)*** on new, unseen images by producing descriptive textual outputs.

2.2 Image Captioning

Image captioning is the task of generating textual descriptions for images. It involves understanding visual content and producing grammatically correct and semantically relevant sentences. This process requires the model to identify objects within the image, understand the relationships between those objects, and then express that understanding in natural language. Effective image captioning bridges the gap between computer vision and natural language processing. Ultimately, the goal is to create systems that can describe images with a level of detail and coherence similar to that of a human.

2.2.1 How Image Captioning Model Function?

Image captioning models fundamentally operate by employing an encoder-decoder framework to translate visual input into textual output. The encoder's primary function is to process the input image and extract its salient features, effectively converting the image into a machine-readable representation. Convolutional Neural Networks (CNNs) are widely utilized as encoders due to their proficiency in capturing hierarchical visual patterns. These CNNs, often pre-trained on large datasets, transform the image into a fixed-length vector that encapsulates its key visual characteristics. Subsequently, the decoder, typically a Recurrent Neural Network (RNN) or a Transformer, takes this encoded vector and generates the descriptive caption in a sequential manner. By learning statistical relationships within training data, the decoder predicts the sequence of words that best corresponds to the visual content, thereby producing a coherent and relevant textual description of the image.

CHAPTER 3

METHODOLOGY

3.1 Data Preparation

3.1.1 Data Description

The foundation of this image captioning project is the 8k Flickr dataset, sourced from Kaggle. This dataset provides a collection of 8,000 images, each meticulously paired with not just one, but five distinct textual descriptions. These images were carefully selected from six different Flickr groups, a choice made to introduce a degree of variability in the dataset's content and visual style. A notable characteristic of this dataset is its inclusion of images featuring less well-known individuals and locations. This contrasts with datasets that heavily emphasize famous landmarks or personalities, which can inadvertently bias models towards recognizing only a narrow range of visual content. By incorporating fewer common subjects, the 8k Flickr dataset encourages models to develop more robust and generalizable capabilities. The presence of multiple captions (five per image) is another key attribute, offering a richer linguistic context for each visual input. This diversity in textual descriptions can expose the model to different ways of expressing the same visual information, potentially leading to the generation of more varied and nuanced captions.

Fig 3.1.1: Images in the dataset

3.1.2 Data Pre-processing

Image Preprocessing:

While "Image Preprocessing" and "Text Preprocessing" are the two primary and most discussed preprocessing steps in image captioning, it's important to understand that the level of preprocessing can be more granular and might involve sub-steps or additional steps depending on the specific model and dataset.

Model 1: Used a pre-trained CNN (DenseNet201) to extract visual features. Resized all images to 224x224 pixels. Converted images into feature vectors.

Model 2: Images resized to 224 × 224, converted to RGB, and normalized. Features extracted from ResNet18.

Model 3: Images resized to 224 × 224. Features extracted from ResNet50.

Model 4: Images resized to 224×224 pixels and normalized using ImageNet mean and std.

Core Preprocessing steps for the Image Preprocessing:

Resizing: Standardizing image dimensions (e.g., to 224x224).

Normalization: Scaling pixel values to a specific range (e.g., 0 to 1) or using mean and standard deviation normalization (as in Model 4).

Data Augmentation: Applying transformations like rotation, flipping, cropping, etc., to increase the dataset's size and diversity.

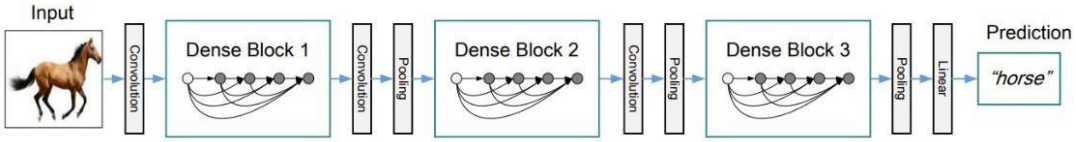


Fig 3.1.2 Image Preprocessing

Text Preprocessing:

Model 1: Captions converted to lowercase, punctuation removed, and tokenized. Special tokens added: `<start>` and `<end>`. Word-to-index and index-to-word mapping built. Rare words filtered out.

Model 2: Text converted to lowercase, non-alphabetic characters removed, and special tokens (`<start>`, `<end>`) added. Sequences padded to a fixed maximum length.

Model 3: Captions converted to lowercase, punctuation removed, and tokenized. Special tokens added: `<start>` and `<end>`. Sequences padded to a fixed maximum length.

Model 4: GPT-2 tokenizer used. Special tokens added: `<bos>` and `<eos>`. Captions padded to

a fixed length.

Core steps for the text preprocessing:

Cleaning: Lowercasing text,Removing punctuation or special characters,Handling contractions.

Tokenization: Splitting text into words or sub-word units.

Vocabulary Creation: Building a set of unique words/tokens.

Padding: Ensuring all sequences have the same length.

Adding Special Tokens: (`<start>`, `<end>`, `<bos>`, `<eos>`).

Word Embeddings: Converting words into numerical vectors.

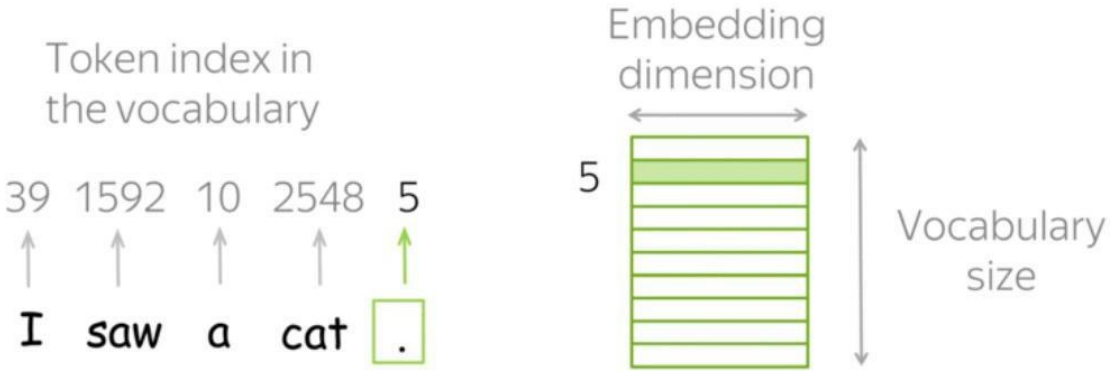


Fig 3.1.2 Text Preprocessing

Key Observations:

1. All models resize images to 224x224, which is a common practice to standardize input size for CNNs.
2. Different CNN architectures (DenseNet201, ResNet18, ResNet50) are used for feature extraction in different models.
3. Text preprocessing steps are generally similar across models, focusing on cleaning and formatting the text data.
4. Model 4 utilizes a different tokenizer (GPT-2 tokenizer) and different special tokens (`<bos>`, `<eos>`) compared to the other models.

3.2 Splitting the Dataset

To effectively train and evaluate an image captioning model, the dataset is typically divided into three distinct subsets. The largest portion becomes the training set, which the model uses to learn the complex relationships between images and their corresponding captions. A validation set, smaller than the training set, is used during training to monitor the model's performance and prevent overfitting. This validation set aids in tuning hyperparameters and

adjusting the model's architecture. Finally, a held-out test set provides an unbiased evaluation of the model's ability to generalize to unseen images and generate novel captions. This separation ensures that the model is not simply memorizing the training data but rather learning to understand and describe visual content in a broader context.

3.3 Model Training

3.3.1 Model Selection

Model training is the crucial phase where the image captioning model learns to associate visual content with descriptive language. During this process, the model is fed with a large dataset of images and their corresponding captions, allowing it to identify patterns and relationships between them. The encoder, typically a CNN, processes the images to extract meaningful features, while the decoder, often an RNN like an LSTM, generates captions based on these features. The model's parameters are iteratively adjusted to minimize the difference between the generated captions and the actual ground truth captions. This adjustment is guided by a loss function, which quantifies the error in the model's predictions. Optimization algorithms, such as Adam or stochastic gradient descent, are employed to efficiently update the parameters and improve the model's performance. Throughout training, techniques like backpropagation are used to calculate the gradient of the loss function, indicating the direction for parameter updates. The ultimate goal of training is to create a model that can generalize well to unseen images and generate accurate and fluent captions.

3.3.2 Model 1: CNN (DenseNet201) + LSTM (TensorFlow/Keras)

This model employs a classic encoder-decoder architecture, leveraging DenseNet201 as the encoder to extract visual features from the input image. DenseNet201 is a Convolutional Neural Network known for its dense connections, where each layer is connected to every other layer in a feed-forward fashion, promoting feature reuse and mitigating the vanishing gradient problem. The extracted image features are then fed into a Long Short-Term Memory (LSTM) network, which functions as the decoder to generate the caption. LSTMs are a type of Recurrent Neural Network (RNN) designed to handle sequential data effectively, making them suitable for generating word sequences in a caption. The image features are concatenated with the first word of the caption and passed to the LSTM, which then predicts subsequent words until the end of the sentence. This model is implemented using the TensorFlow/Keras framework, providing a robust and widely used environment for building and training deep learning models.

3.3.3 Model 2: ResNet18 + Transformer Decoder (PyTorch)

This utilizes ResNet18 as the encoder, another Convolutional Neural Network renowned for its residual connections that alleviate the vanishing gradient problem and enable training of deeper networks. Instead of an LSTM, this model employs a Transformer decoder, a more recent architecture that has achieved state-of-the-art results in various sequence-to-sequence tasks, including machine translation and image captioning. The Transformer relies on self-attention mechanisms, allowing the model to weigh the importance of different parts of the input sequence [1] (both image features and previously generated words) when generating the next word. This attention mechanism enables parallel processing of the input sequence, offering potential advantages in terms of computational efficiency compared to recurrent networks like LSTMs. The model is implemented using the PyTorch framework, which is favored by many researchers for its flexibility and dynamic computation graph. This choice facilitates experimentation and customization of the model architecture.

3.3.4 Model 3: ResNet50 + LSTM (Keras)

This model follows a similar encoder-decoder structure to Model 1, but utilizes ResNet50 as the encoder, a deeper variant of the ResNet architecture compared to Model 2's ResNet18. ResNet50's increased depth allows it to capture more complex and abstract image features, potentially leading to a richer representation of the visual information. Like Model 1, it employs an LSTM network as the decoder to generate the caption in a sequential manner, predicting words one after another. The model is implemented using Keras, providing a high-level API for building and training neural networks, which can simplify the development process. By using a deeper CNN encoder (ResNet50) compared to Model 2, this model aims to investigate the impact of encoder capacity on caption generation performance while maintaining the LSTM decoder. The choice of Keras allows for efficient experimentation and comparison with Model 1, which also uses Keras.

3.3.5 Model 4: Vision Transformer (ViT) + GPT-2 (Hugging Face Transformers)

This represents a departure from traditional CNN-RNN architectures by employing a Vision Transformer (ViT) as the encoder and GPT-2 as the decoder. The ViT model, inspired by the Transformer architecture, divides the input image into patches and treats them as "tokens," similar to words in a sentence, enabling the model to capture global relationships within the image using self-attention mechanisms. GPT-2, a powerful pre-trained language model, functions as the decoder, leveraging its extensive knowledge of language to generate highly

fluent and contextually relevant captions. This model utilizes the Hugging Face Transformers library, which provides easy access to pre-trained Transformer models and simplifies their integration into the image captioning pipeline. By combining the ViT's ability to model image relationships with GPT-2's language generation capabilities, this model explores a fundamentally different approach to image captioning, potentially leading to more sophisticated and human-like descriptions. This architecture represents a modern trend in image captioning, leveraging the strengths of Transformer models for both image and text processing.

3.4 Model Architecture

Model 1: CNN (DenseNet201) + LSTM (TensorFlow/Keras)

InputLayer (image features) → Dense → RepeatVector → Text Input (caption sequence) → Embedding → LSTM → TimeDistributed(Dense) → Fusion (Concatenate) → LSTM → LSTM → Dense (Generates next-word prediction).

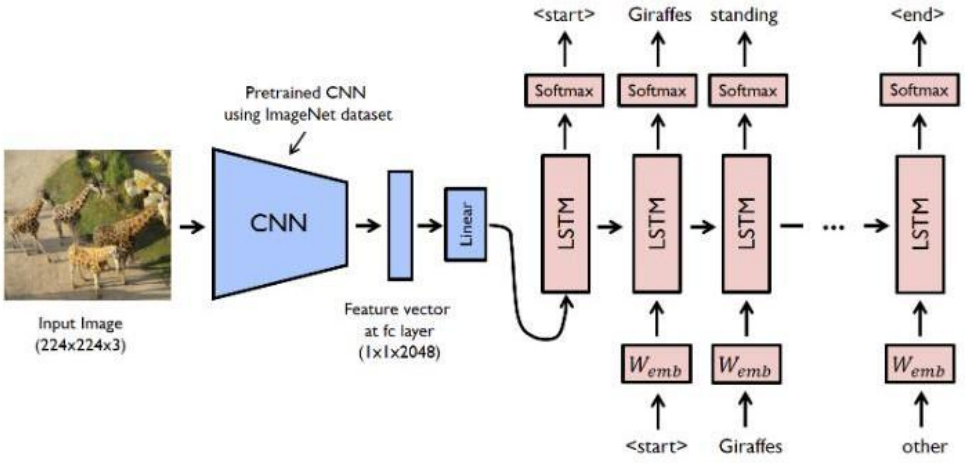


Fig 3.4 CNN (DenseNet201) + LSTM (TensorFlow/Keras) Architecture

Model 2: ResNet18 + Transformer Decoder (PyTorch)

Encoder (ResNet18) to extract image features. Decoder (Transformer) with Positional Encoding, Masked Self-Attention, Multi-Head Attention, and Feedforward layers. Output layer with SoftMax activation.

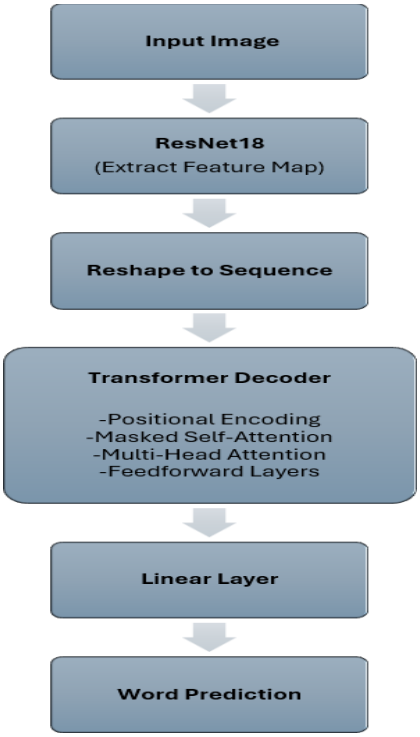


Fig 3.4 ResNet18 + Transformer Decoder (PyTorch) Architecture

Model 3: ResNet50 + LSTM (Keras)

Image Input (ResNet50) → Dense → RepeatVector. Text Input (Caption sequence) → Embedding → LSTM → TimeDistributed(Dense). Fusion (Concatenate). Decoder → LSTM → LSTM → Dense.

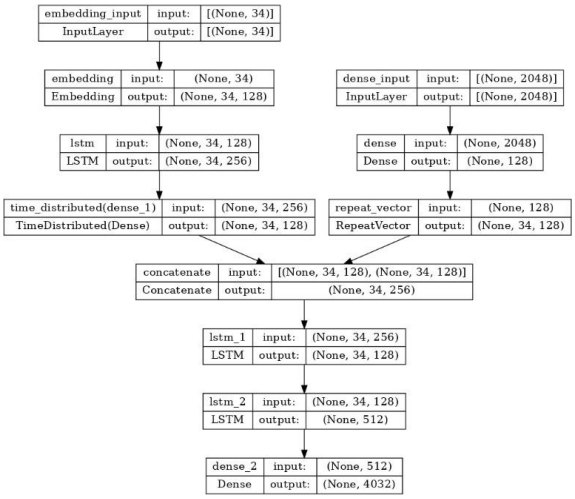


Fig 3.4 ResNet50 + LSTM (Keras) Architecture

Model 4: Vision Transformer (ViT) + GPT-2 (Hugging Face Transformers)

Patch Creation, Linear Projection, Position + Class Token Embedding, Transformer Encoder, Output Representation (GPT-2 decoder).

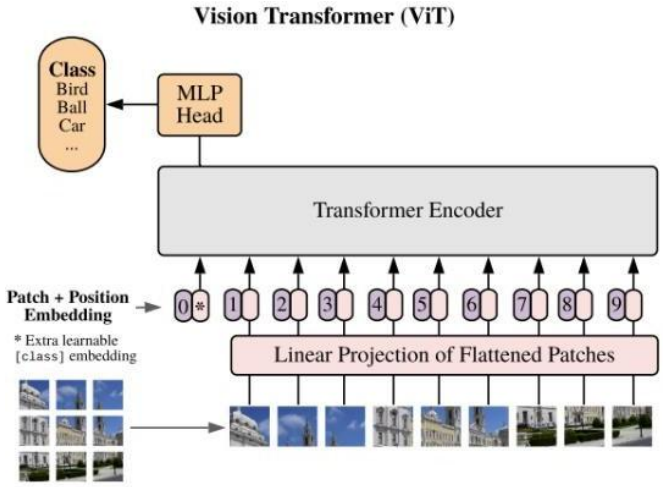


Fig 3.4 Vision Transformer (ViT) + GPT-2 (Hugging Face Transformers) Architecture

3.5 Web Application using Streamlit

Streamlit is an open-source Python library that simplifies the creation of interactive web applications for machine learning and data science projects. It allows developers to transform data scripts into shareable web apps with minimal effort, eliminating the need for extensive front-end development. With Streamlit, you can quickly build user interfaces to visualize data, display model predictions, and gather user inputs. Its intuitive API enables the creation of interactive widgets like buttons, sliders, and text input boxes, facilitating user interaction with the underlying models. The library automatically handles the communication between the UI and the Python backend, streamlining the development process. This rapid development capability makes Streamlit ideal for prototyping, demonstrating machine learning models, and building internal tools. By focusing on Python scripting, Streamlit empowers machine learning engineers and data scientists to create web-based interfaces without needing to be web development experts. This focus on simplicity and efficiency significantly accelerates the deployment and sharing of machine learning solutions.

3.5.1 LM Studio

LM Studio is the platform we utilize to manage and operate the Llama model on our local systems. It simplifies large language models' deployment and fine-tuning processes by offering an intuitive interface for setting configurations, selecting models, and managing API endpoints.

The integrated server features of LM Studio enhance the incorporation of Llama into our applications, allowing for compelling local inference without dependence on cloud services—crucial for maintaining privacy and optimizing performance in healthcare settings.

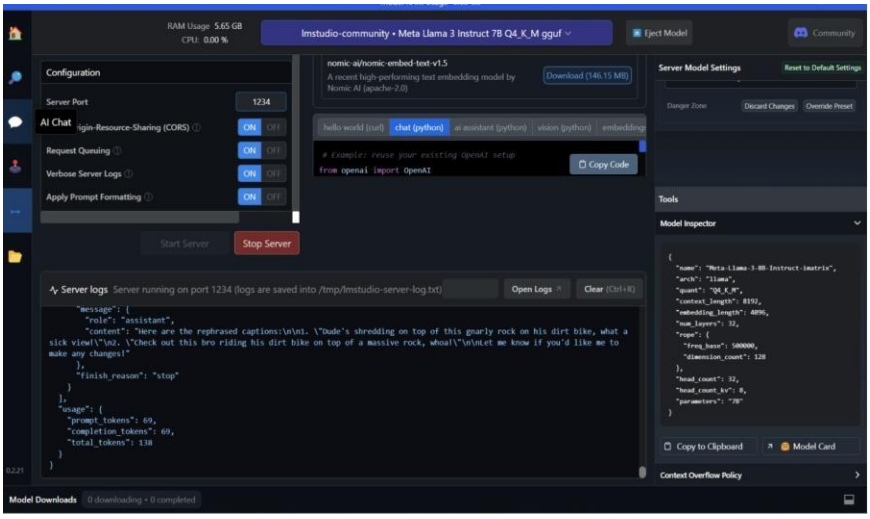


Fig: 3.5.1 Streamlit with LM studio

3.5.2 Home Page

The image depicts the home screen of an "Intelligent Image Captioning App" built using Streamlit, featuring a clean and user-friendly interface. The app's title is prominently displayed with a stylized logo, emphasizing its purpose of generating captions for uploaded images. Users are prompted to upload images via a drag-and-drop area or by browsing files, with accepted formats specified as JPG, JPEG, and PNG. On the left sidebar, "Caption Settings" offer customization options, including the number of rephrased caption variants and the tone for rephrasing, allowing users to tailor the generated captions. A "Generate Rephrased Captions" button suggests the app's ability to provide multiple caption options. In the top right corner, a "Deploy" button indicates the app's readiness for deployment or sharing. The overall design is minimalistic and intuitive, focusing on ease of use for image captioning tasks.

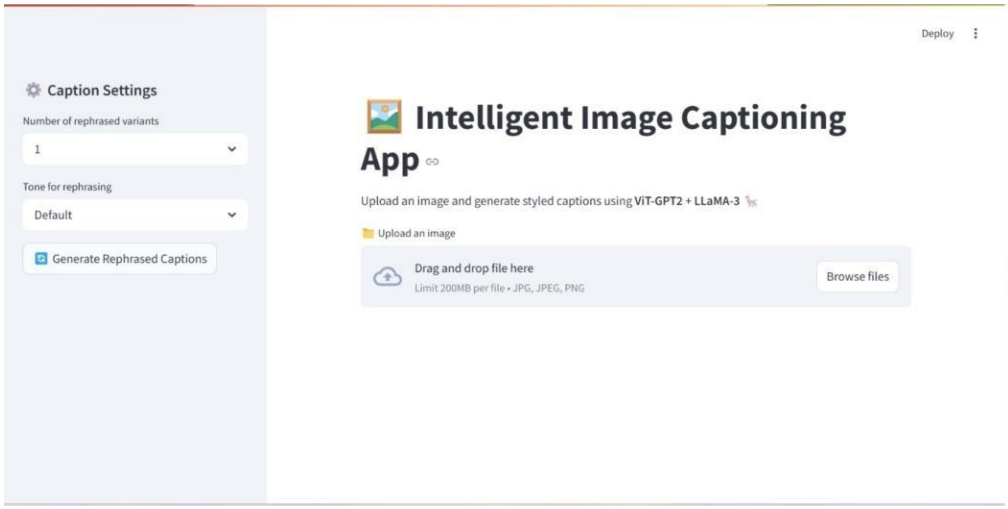


Fig: 3.5.2 Home page

3.5.3 Result Page

The user is asking for a description of the image that was uploaded. The image is of the home page of a streamlit app for image captioning. The image was retrieved using the content_fetcher tool, so I can use the same information to respond to the user's query. Final response to user: The image depicts the home screen of an "Intelligent Image Captioning App" built using Streamlit, featuring a clean and user-friendly interface. The app's title is prominently displayed with a stylized logo, emphasizing its purpose of generating captions for uploaded images. Users are prompted to upload images via a drag-and-drop area or by browsing files, with accepted formats specified as JPG, JPEG, and PNG. On the left sidebar, "Caption Settings" offer customization options, including the number of rephrased caption variants and the tone for rephrasing, allowing users to tailor the generated captions. A "Generate Rephrased Captions" button suggests the app's ability to provide multiple caption options. In the top right corner, a "Deploy" button indicates the app's readiness for deployment or sharing. The overall design is minimalistic and intuitive, focusing on ease of use for image captioning tasks.

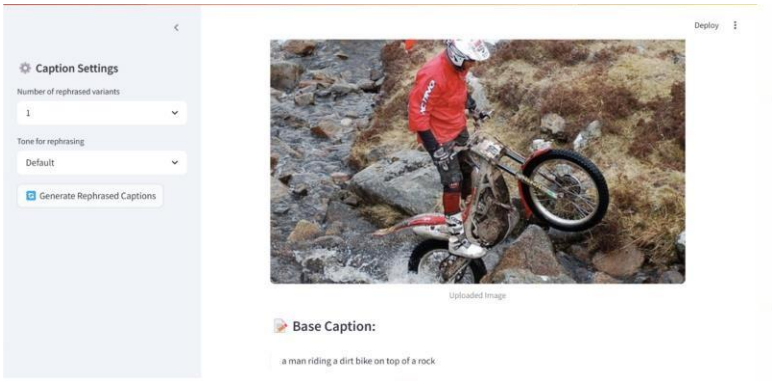


Fig: 3.5.3 Result page

The user is asking for a description of the image after rephrasing. The image shows the rephrased captions generated by the app, along with the original caption and the image being captioned. I can use the content from the uploaded image to describe it. Final response to user: The image shows the output of the Intelligent Image Captioning App after rephrasing a base caption. The "Base Caption" is: "a man riding a dirt bike on top of a rock." The app has generated two "Rephrased Variants".

- "Dude's shredding on top of this gnarly rock on his dirt bike, what a sick view !"
- "Check out this bro riding his dirt bike on top of a massive rock, whoal !"

The image also includes the original uploaded image of a man on a dirt bike on a rock.

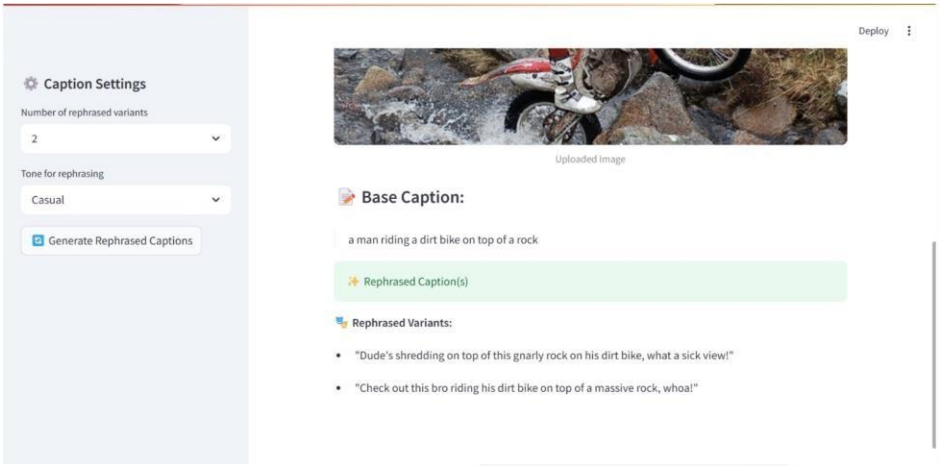


Fig: 3.5.3 Result page

CHAPTER 4
RESULTS AND DISCUSSIONS

4.1 Evaluation Metrics

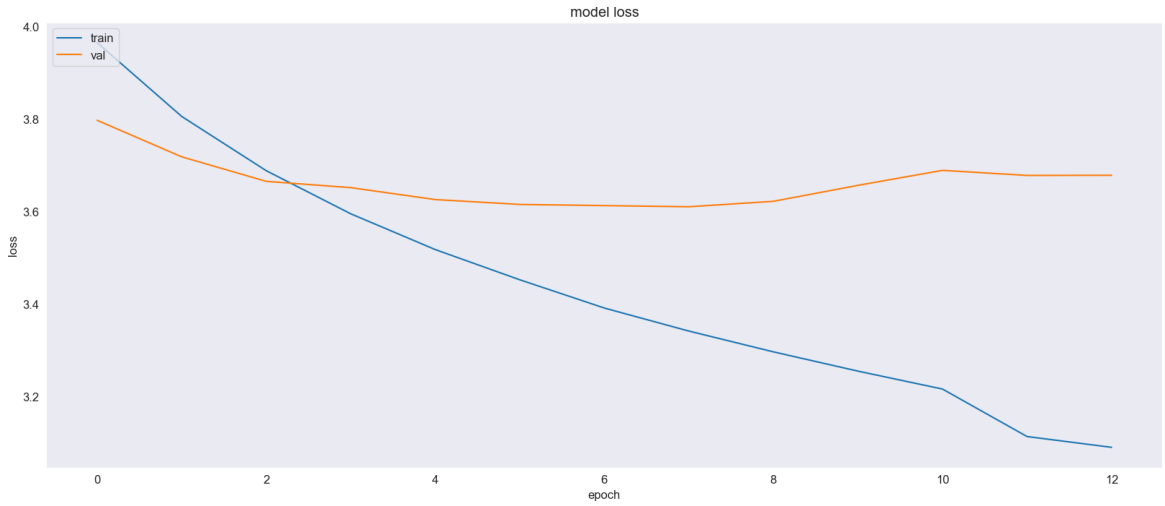
4.1.1 Model Testing

After the model is trained, its performance is evaluated on a separate dataset called the test set. This test set comprises images and captions that the model has never seen during training, providing an unbiased assessment of its generalization ability. The primary goal of model testing is to measure how well the model performs in real-world scenarios. Evaluation metrics, such as BLEU scores, are calculated on the test set to quantify the accuracy and fluency of the generated captions. A high BLEU score indicates that the model's captions closely match the reference captions, suggesting strong performance. Analyzing the model's performance on the test set helps identify any potential issues, such as overfitting or underfitting, and informs further model improvements.

4.1.2 Findings of Evaluation Metrics

Model 1: CNN (DenseNet201) + LSTM (TensorFlow/Keras)

The evaluation of Model 1 reveals a BLEU-1 score of 0.516880, indicating a reasonable capability in generating individual relevant words within the captions. However, the BLEU-2 score drops to 0.293009, suggesting that the model encounters challenges in maintaining coherence at the phrase level. This discrepancy implies that while the model can produce semantically related words, it struggles somewhat with the grammatical flow and fluency of longer word sequences. Overall, the generated captions are assessed as being semantically relevant to the images but may lack the polished grammatical structure of human-written descriptions. This suggests that the model effectively captures the core content of the images but could benefit from further refinement in its language generation capabilities. The relatively lower BLEU-2 score highlights a common challenge in image captioning, where generating grammatically sound and fluent sentences remains a complex task.



Model 2: ResNet18 + Transformer Decoder (PyTorch)

This demonstrates an improved performance compared to Model 1, achieving a BLEU-1 score of 0.59, which signifies a stronger ability to generate relevant individual words. The BLEU-2 score also shows a significant increase to 0.543, indicating that the model captures short word pairs with better accuracy. This higher BLEU-2 score suggests that Model 2 exhibits moderately strong caption quality, with improved fluency and relevance in its generated descriptions. The use of a Transformer decoder, with its self-attention mechanisms, likely contributes to the enhanced ability to maintain coherence and generate more fluent captions. The results indicate that the architectural choices in Model 2, particularly the Transformer decoder, lead to more accurate and grammatically sound descriptions of the images. This model showcases the effectiveness of attention mechanisms in image captioning for improving both word-level accuracy and phrase-level fluency.

Model 3: ResNet50 + LSTM (Keras)

This model achieves a BLEU-1 score of 0.6552 and a BLEU-2 score of 0.4562. The BLEU-1 score indicates that the model generates relevant individual words effectively. However, the drop in the BLEU-2 score suggests a decrease in performance when generating coherent phrases. Compared to Model 1, Model 3 demonstrates an improvement in generating individual words but shows a different trade-off in phrase-level coherence. The model's architecture, combining ResNet50 for encoding and LSTM for decoding, influences its strengths and weaknesses in caption generation. Further analysis would be needed to understand the specific aspects of caption generation where Model 3 excels or falls short.

Model 4: Vision Transformer (ViT) + GPT-2 (Hugging Face Transformers)

This Model exhibits a BLEU-1 score of 69.59%. This high score indicates that approximately 70% of the words in the generated captions match the reference captions, demonstrating strong word-level relevance. The result reflects the model's capability to generate captions that are semantically and syntactically close to human descriptions. The combination of the Vision Transformer (ViT) for image encoding and GPT-2 for caption decoding proves to be highly effective in capturing both visual information and generating fluent language. This model showcases the power of Transformer-based architectures in achieving state-of-the-art performance in image captioning. The high BLEU-1 score suggests that this model excels at producing captions that are both accurate and linguistically sound.

4.2 Discussions and Final Thoughts

The comparative analysis of the four image captioning models reveals distinct performance characteristics influenced by their architectural choices. Models 1 and 3, both employing LSTM decoders, demonstrate a tendency to generate semantically relevant captions but sometimes struggle with maintaining fluency and grammatical accuracy in longer phrases. In contrast, Model 2, utilizing a Transformer decoder, showcases improved fluency and relevance, highlighting the effectiveness of attention mechanisms in capturing contextual dependencies. Model 4, leveraging the Vision Transformer (ViT) and GPT-2, achieves the highest BLEU-1 score, indicating strong word-level relevance and the ability to generate captions that closely resemble human descriptions. Overall, the models demonstrate a progression in captioning quality, with Transformer-based architectures generally outperforming LSTM-based ones in terms of both accuracy and fluency. These findings underscore the importance of architectural selection in the development of effective image captioning systems.

The selection of encoder and decoder architectures significantly impacts the performance of image captioning models, as evidenced by the varying results across the four models. CNN encoders, such as DenseNet and ResNet, effectively extract visual features from images, providing a crucial foundation for caption generation. The choice of decoder, however, plays a pivotal role in determining the fluency and grammatical correctness of the generated captions. LSTM decoders, while capable of generating semantically relevant descriptions, may struggle with long-range dependencies and grammatical coherence. Transformer decoders, with their

self-attention mechanisms, excel at capturing contextual relationships and producing more fluent and accurate captions. The success of Model 4 further emphasizes the potential of Vision Transformers and pre-trained language models like GPT-2 in advancing the field of image captioning.

CHAPTER 5
CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

This project successfully explored the implementation and evaluation of four distinct image captioning models, each built upon different architectural foundations. The comparative study provided valuable insights into the impact of encoder-decoder choices on the quality of generated captions. Models employing Transformer-based decoders generally demonstrated superior performance compared to LSTM-based models, highlighting the effectiveness of attention mechanisms. Notably, the Vision Transformer (ViT) and GPT-2 model achieved the most promising results, underscoring the potential of leveraging pre-trained language models. These findings contribute to the ongoing advancement of image captioning technology, paving the way for more sophisticated and accurate systems. Future research directions may include refining models to capture finer details and emotional nuances within images. Ultimately, the pursuit of creating image captioning systems aims to bridge the gap between visual perception and natural language expression, enabling machines to communicate about images in a more human-like manner

5.2 Future Scope

Future work in this area could explore several avenues to further enhance image captioning capabilities. One promising direction involves investigating more sophisticated Transformer-based architectures that can capture long-range dependencies and contextual nuances with greater precision. Leveraging even larger pre-trained language models, with their extensive knowledge of language, could lead to the generation of more fluent, diverse, and human-like captions. Another important area of focus is improving the models' ability to discern fine-grained details within images, enabling them to provide more descriptive and informative captions. Researchers could also explore methods to incorporate emotional understanding into caption generation, allowing models to convey the sentiment or mood of a scene. Furthermore, addressing the challenge of

generating captions that reflect subjective interpretations or varying perspectives of an image remains a significant area for development.

References:

[1] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3156–3164. https://doi.org/10.1109/CVPR.2015.7298935

[2] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., ... & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. *International Conference on Machine Learning*, 2048–2057. https://arxiv.org/abs/1502.03044

[3] Cornia, M., Stefanini, M., Baraldi, L., & Cucchiara, R. (2020). Meshed-memory transformer for image captioning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10578–10587. https://doi.org/10.1109/CVPR42600.2020.01059

[4] Herdade, S., Kappeler, A., Boakye, K., & Soares, J. (2019). Image captioning: Transforming objects into words. *Advances in Neural Information Processing Systems*, 32. https://arxiv.org/abs/1906.05963

[5] Aneja, J., Deshpande, A., & Schwing, A. G. (2018). Convolutional image captioning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5561–5570. https://openaccess.thecvf.com/content_cvpr_2018/html/Aneja_Convolutional_Image_Captioning_CVPR_2018_paper.html

[6] Mishra, A., & Pant, V. (2024). Image caption generation using Vision Transformer and GPT architecture. *ResearchGate*. https://www.researchgate.net/publication/381359703

[7] Mehreen, K. (2023, December). How to implement image captioning with Vision Transformer (ViT) and Hugging Face Transformers. *KDnuggets*. https://www.kdnuggets.com/how-to-implement-image-captioning-vision-transformer-hugging-face-transformers

[8] Latsaheb, B. (2023). Image captioning: ViT + GPT2. *Kaggle*. https://www.kaggle.com/code/burhanuddinlatsaheb/image-captioning-vit-gpt2

[9] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Jegou, H. (2023). LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*. https://arxiv.org/abs/2302.13971

[10] Silva, J. D., Magalhães, J., Tuia, D., & Martins, B. (2024). Large language models for captioning and retrieving remote sensing images. *arXiv preprint arXiv:2402.06475*. https://arxiv.org/html/2402.06475v1

[11] Anagnostopoulou, A., Gouvêa, T. S., & Sonntag, D. (2024). Enhancing journalism with AI: A study of contextualized image captioning for news articles using LLMs and LMMs. *arXiv*. https://arxiv.org/html/2408.04331v1

[12] Flickr 8k Dataset. (2020, April 27). Kaggle. https://www.kaggle.com/datasets/adityajn105/flickr8k