TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

A

PROJECT REPORT

ON

IMAGE CAPTIONING AND SEARCH

SUBMITTED BY:

NAYAN PANDEY (PUL077BCT049)

PRATIK SHRESTHA (PUL077BCT059)

SUJAN KAPALI (PUL077BCT086)

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS & COMPUTER ENGINEERING

November, 2024

Acknowledgments

First and foremost, we extend our deepest gratitude to Dr. Arun Kumar Timalsina for his exceptional guidance, profound expertise, and unwavering encouragement, which have been instrumental in shaping the trajectory of this project.

We also express our heartfelt appreciation to the Samsung Innovation Campus (SIC) team for their steadfast support, encouragement, and guidance throughout the course. Their dedication to fostering innovation and providing a platform for learning has been invaluable.

A special note of thanks to our supervisor, Mr. Daya Sagar Baral, and mentor, Dr. Suresh Pokharel, for their consistent support, insightful supervision, and expert direction. Their invaluable contributions have been integral to the successful initiation and progress of this project.

Lastly, we are sincerely grateful to our colleagues, friends, and peers for their enriching interactions, constructive feedback, and collaborative spirit. Their intellectually stimulating discussions and encouragement have greatly enhanced the overall quality of our work.

Abstract

The rapid growth of digital image collections on personal devices and cloud platforms has led to the need for advanced tools to efficiently organize and retrieve visual content. This report presents the development of a gallery app that leverages state-of-the-art image captioning and text-based search technologies to enhance user experience in managing and exploring photo collections. The project focuses on creating a parameter-efficient image captioning model that can generate accurate and contextually relevant descriptions, suitable for running natively on edge devices. Additionally, a text-based search engine is designed to allow users to perform natural language queries to find specific images within their collections. The app features an intuitive user interface ensuring broad compatibility and robust privacy measures.

Keywords: Image Captioning, Text-based Search, Deep Learning, Parameter Efficiency, Image Retrieval, Natural Language Processing, Edge Devices.

Contents

List of Figures

List of Abbreviations

GPU	Graphics Processing Unit
MLP	Multi Layer Perceptron
SOTA	State of the art
AI	Artificial Intelligence
LLMs	Large Language Models
CBIR	Content Based Image Retrieval
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network

1. Introduction

1.1 Background

The increasing volume of digital images stored on personal devices and cloud platforms has created a demand for efficient and intuitive ways to organize and retrieve visual content. As photo collections grow, traditional methods of searching and categorizing images become insufficient, leading to frustration and inefficiency. Recent advancements in computer vision and natural language processing have opened up new possibilities for enhancing how users interact with their digital galleries.

Image captioning and search technologies are now capable of automatically generating descriptive tags and allowing users to search for images using natural language queries. This shift from manual tagging to automated, intelligent systems is particularly valuable for users who want to quickly find specific images based on content, context, or even emotions conveyed in the photos. By leveraging these technologies, a gallery app can transform the way users manage and explore their photo collections, making it easier to find the exact image they are looking for.

1.2 Problem statements

Despite the advancements in image captioning and search technology, there are still significant challenges in developing a user-friendly gallery app that allows for accurate and contextually relevant text-based photo searches. Current image captioning algorithms may not fully capture the specific details, context, or emotional nuances of photos, leading to inaccurate or generic tags. This can make it difficult for users to retrieve the images they want, especially when dealing with complex or abstract visual content.

Additionally, search algorithms may struggle to understand and process natural language queries effectively, resulting in search results that do not align with user expectations. For instance, a user searching for a "sunset at the beach with friends" may receive results that are either too broad or miss the emotional context of the query. These limitations can undermine the usability and appeal of the gallery app, reducing user satisfaction.

This project aims to develop an advanced gallery app that leverages state-of-the-art image captioning and search technologies to provide users with a seamless and intuitive experience. By improving the accuracy of image descriptions and enhancing the contextual understanding of search queries, the app will enable users to efficiently find and organize

their photos, making their digital collections more accessible and enjoyable to explore.

1.3 Objectives

The objectives of the project are:

- Create an image captioning algorithm that accurately and contextually describes the content of each photo in the gallery
- Design and integrate a text-based search engine that allows users to search for photos using natural language queries.
- Develop a user-friendly interface that makes it easy for users to interact with the gallery app.

1.4 Scope

The scope of this project includes developing a gallery app with advanced image captioning and text-based search functionalities, enabling users to efficiently organize and retrieve photos. The app will feature an intuitive user interface optimized for real-time performance, supporting natural language queries in multiple languages, including context-sensitive searches. Integration with popular cloud storage services and local device storage will be a key focus, ensuring seamless access and organization of photo collections across platforms. Additionally, robust privacy and data security measures will be implemented to protect user data, while the app will be designed to run on major mobile operating systems, ensuring broad compatibility.

2. Literature Review

The rapid advancements in computer vision and natural language processing have significantly contributed to the development of image captioning and text-based image retrieval systems. These technologies have become integral to various applications, from social media to digital libraries, enhancing how users interact with visual content.

Image captioning involves generating descriptive text for images, a task that bridges the gap between visual data and natural language. Early approaches to image captioning, such as template-based methods, relied heavily on predefined structures and were limited in their ability to capture the complexity and diversity of visual scenes. More recent approaches have leveraged deep learning techniques, particularly convolutional neural networks (CNNs) combined with recurrent neural networks (RNNs), to generate more accurate and contextually relevant captions. A seminal work by Vinyals et al. (2015) [1] introduced the use of an encoder-decoder framework where CNNs encode the image and RNNs generate the captions, showing significant improvements in the quality of generated descriptions . Another important contribution was the introduction of attention mechanisms, which allow the model to focus on specific parts of an image when generating captions, as demonstrated by Xu et al. (2015)[2] .

The concept of image-based search has evolved from traditional text-based search methods to more sophisticated systems that allow users to query with images or text. Early image retrieval systems, known as content-based image retrieval (CBIR), focused on low-level visual features such as color, texture, and shape. However, these methods often struggled with semantic understanding, leading to irrelevant results. The integration of deep learning, particularly CNNs, has greatly enhanced the capability of image-based search systems by enabling the extraction of high-level semantic features from images. The work by Liu et al. (2016) [3] on deep learning for image retrieval highlights the significant improvements made in this field, particularly in handling large-scale image datasets . Additionally, the development of cross-modal retrieval systems, as discussed by Wang et al. (2017)[4], allows for more effective searches across different types of media, such as using text to search for images.

Quantization has emerged as a pivotal technique in optimizing deep learning models for resource-constrained environments, such as edge devices and mobile applications. It involves converting floating-point computations and parameters into lower precision formats, such as

8-bit integers, to reduce memory requirements and improve computational efficiency. Early works in quantization, such as that by Gong et al. (2014) [5], explored vector quantization methods to compress deep networks, demonstrating significant memory savings with minimal accuracy loss. Jacob et al. (2018) [6] formalized the practical implementation of quantization-aware training (QAT), enabling models to simulate low-precision arithmetic during training and achieve improved post-quantization accuracy. Another milestone in the field was the introduction of post-training quantization (PTQ), as detailed by Zhao et al. (2019) [7], which allowed for efficient quantization without requiring extensive retraining. Furthermore, advanced techniques such as mixed-precision quantization, where layers in a network use varying bit-widths to balance accuracy and efficiency, have been explored in works like that of Wang et al. (2019) [8]. These advancements underscore the growing importance of quantization as a key enabler for deploying deep learning models on hardware with limited computational resources.

2.1 Related work

Several products and applications have been developed that utilize image captioning and text-based image retrieval technologies, each offering unique features and capabilities that have influenced the development of gallery and search applications.

2.1.1 Samsung Gallery

Samsung's default photo management app, Samsung Gallery, is integrated into its Galaxy line of smartphones. The app offers robust photo organization and search capabilities powered by AI. Users can search for images using keywords or phrases, as the app automatically tags photos with relevant metadata based on the content of the images. Samsung Gallery also supports face recognition and scene categorization, making it easier for users to locate specific photos by searching for people, locations, or objects depicted in the images.

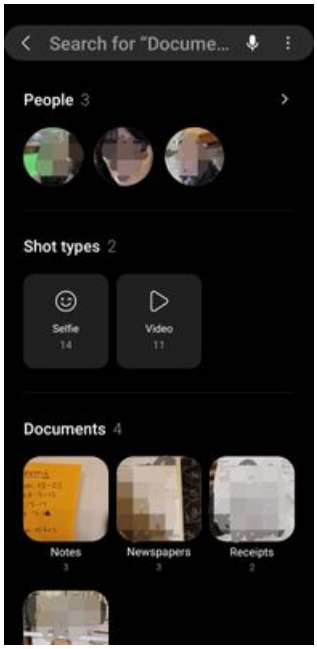

Figure 2.1: Samsung Gallery App

2.1.2 Google Photos

Google Photos is one of the most widely used photo management apps, incorporating advanced AI-driven features for image organization and search. The app uses machine learning algorithms to automatically tag photos with relevant keywords and offers powerful search capabilities that allow users to find images based on text descriptions, dates, locations, and even the objects or people in the photos. Google Photos also leverages facial recognition and scene detection to improve search accuracy, making it a go-to solution for users with large

photo collections.

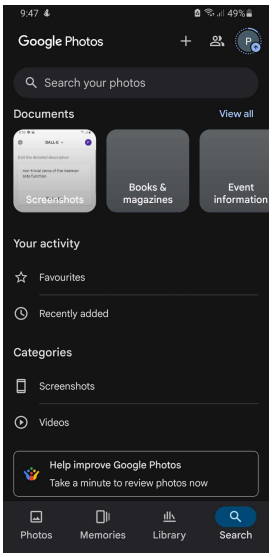

Figure 2.2: Google Gallery App

2.1.3 Pinterest

Pinterest has integrated visual search capabilities that allow users to find images and products based on visual inputs. The platform uses image recognition technology to analyze and categorize images, enabling users to search for similar content by selecting an object or area within an image. Pinterest's Lens feature further extends this capability, allowing users to take photos of real-world objects and search for related items on the platform.

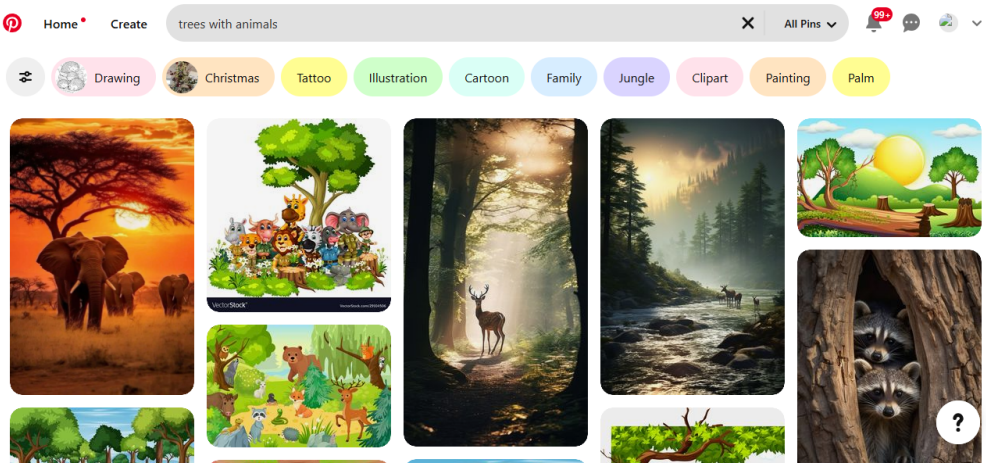

Figure 2.3: Pinterest

3. Methodology

Many research have already dealt with generating captions with high accuracy. But, they rely on LLMs and large image encoders which can be hard to train and cannot be run natively on edge devices like mobile phones. Our focus is on parameter-efficient models which can give good enough results. For this, we'll follow the following steps:

3.1 Dataset

For this project, the required dataset consists of a large number of images with paired with corresponding textual descriptions or annotations. The dataset of that kind is generally widely available and well documented. Some of the examples of the available datasets are as follows:

3.1.1 MS COCO

MS COCO (Microsoft Common Objects in Context) is a large-scale object detection, segmentation and captioning dataset. It was developed by Microsoft and is designed to provide a rich and diverse collection of images that reflect everyday scenes and objects in natural settings. The dataset contains over 330,000 images, with more than 200,000 of these annotated with detailed descriptions in the form of five unique captions per image.

3.1.2 Flickr8k

The Flickr8k dataset is a widely used resource in the field of computer vision, particularly for tasks related to image captioning. It consists of 8,000 images sourced from the Flickr photo-sharing platform, each paired with five different descriptive captions. These captions are written by humans, providing natural language descriptions that capture the key visual elements and contexts within each image.

The dataset is designed to facilitate research in connecting visual content with natural language, making it an excellent choice for training and evaluating models that generate textual descriptions of images or understand visual scenes. Despite its relatively small size compared to other datasets like MS COCO, Flickr8k is valued for its high-quality annotations and has been a foundational dataset in the development of image captioning techniques.

3.2 Image Captioning Model

Image captioning is a challenging task that involves two components: Image Encoder and Text Decoder.

3.2.1 Image Encoder: DeiT, VIT

DeiT (Data-efficient Image Transformers) is a family of transformer-based models introduced by researchers at Facebook, designed to achieve high performance on image classification tasks while minimizing the amount of labeled data required for training. Among these models, DeiT-Tiny is a lightweight variant that employs a patch-based approach, dividing input images into smaller patches and processing them using transformer architecture. Specifically, the DeiT-Tiny model uses a patch size of 16x16 and operates on images of size 224x224 pixels.

The key innovation behind DeiT is its data-efficient training strategy, which incorporates knowledge distillation from a pre-trained teacher model. This technique enables the DeiT models to leverage the rich representations learned by larger models, enhancing their performance without the need for extensive datasets. As a result, DeiT-Tiny is particularly suitable for resource-constrained environments where computational efficiency is paramount.

Due to its compact size and effective training methodology, DeiT-Tiny has gained popularity for various computer vision applications, including image classification and feature extraction, offering a powerful yet efficient alternative to traditional convolutional neural networks (CNNs) and larger transformer models.

3.2.2 Text Decoder: GPT 2, Distil GPT2, Tiny GPT2

DistilGPT-2 is a smaller, distilled version of the original GPT-2 (Generative Pre-trained Transformer 2) model developed by Hugging Face, aimed at providing a balance between performance and computational efficiency. As a transformer-based language model, DistilGPT-2 retains much of the capabilities of its larger predecessor while being significantly lighter and faster. This is achieved through a process known as knowledge distillation, where the distilled model learns to replicate the behavior of a larger teacher model, effectively transferring its knowledge to a more compact architecture.

Specifically, DistilGPT-2 is approximately 60 percent smaller than GPT-2, featuring fewer parameters while maintaining a similar level of performance on various language tasks such as text generation, completion, and conversational applications. The reduction in size not only allows for faster inference times but also makes the model more accessible for deployment in resource-constrained environments.

3.3 Grad-CAM for Activation Visualization

Grad-CAM (Gradient-weighted Class Activation Mapping) is an interpretability technique that provides insights into the regions of an image that influence the predictions of a deep learning model. In this project, Grad-CAM is used to visualize the attention of the image captioning model, highlighting the areas in the image that contribute to generating specific words in the caption. This aids in understanding how the model aligns visual features with textual outputs.

The Grad-CAM process begins by computing the gradients of the target textual token probability, denoted as y_c, with respect to the feature maps A_k from the image encoder. The importance weight for each feature map is calculated as:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y_c}{\partial A_k^{i,j}},$$

where α_k^c represents the weight for the k-th feature map, Z is the total number of pixels in the feature map, and $\frac{\partial y_c}{\partial A_k^{i,j}}$ is the gradient of y_c with respect to the (i, j)-th spatial location in A_k.

The class activation map is then computed as a weighted combination of the feature maps, followed by a ReLU activation to focus on positive contributions:

$$L_c = \text{ReLU}\left(\sum_k \alpha_k^c A_k\right).$$

This activation map L_c is resized to the input image dimensions and overlaid on the original image to generate a heatmap, highlighting regions that influenced the word prediction.

By applying Grad-CAM at each step of caption generation, it is possible to visualize the evolving attention of the model as it processes the image to generate a descriptive caption. This enhances the interpretability of the image captioning model and provides transparency in its decision-making process.

3.4 Search

Once captions are generated for each image, searching through them becomes a trivial task. We can generate the embedding for the query and vectorize the input text. Then a similarity metric like cosine similarity can be computed between the vectorized input text and each vectorized image caption. Then, the images can be retrieved based on the similarity score achieved.

3.5 Quantization

Quantization involves mapping the floating-point weights and activations of a neural network to lower bit representations, such as 8-bit or 4-bit integers. This process significantly reduces the model's memory footprint and computational complexity, enabling faster inference and lower power consumption.In post-training quantization (PTQ), the trained model's weights and activations, initially represented as 32-bit floating-point values (R), are converted to lower precision formats, such as 8-bit integers (Z_{256}), using a process that minimizes the quantization error. This involves defining a quantization scale (s) and zero-point (z) to map the range of floating-point values to the integer domain. The quantization of a floating-point value x is given by:

$$q = \text{round}\left(\frac{x}{s}\right) + z,$$

where

$$s = \frac{x_{\max} - x_{\min}}{2^b - 1}$$

is the scaling factor for b-bit quantization, and z ensures the integer representation aligns with zero in the floating-point domain. The dequantization process, which converts the quantized values back to the floating-point domain for inference, is expressed as:

$$x' = s \cdot (q - z).$$

By performing quantization after training, PTQ avoids the computational overhead of retraining but may introduce slight accuracy degradation due to the approximation errors in representing floating-point weights and activations with lower-precision integers.

3.6 Web interface

The end user will interact with the system through a web interface. The user can upload their photos to the system. The user can then provide the textual description of the image that they want to search. The model will perform the necessary computations and show only the relevant images to the user.

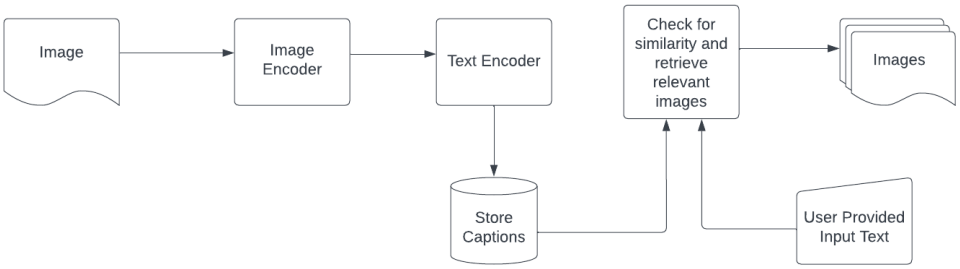

Figure 3.1: Block Diagram

4. Results and Discussion

4.1 Training Vision Encoder Decoder Model

We utilized Hugging Face's VisionEncoderDecoder API to train our models, which facilitates the integration of transformer-based vision and language models for sequence-to-sequence generation tasks. This API provides a straightforward framework for combining visual and textual data, enabling the development of models that can generate descriptive text based on image input. In our experiments, we tested various combinations of encoders and decoders to assess their performance in generating captions from images.

4.1.1 Training Larger Models for few epochs vs Training Smaller Models for higher epochs (under resource constraints)

We can construct a large variety of image captioning models because of a large number of available vision encoder and language decoders. But, we had limited computation resources and time. So, we made choices to construct the 2 largest possible model and one smallest possible model that can be trained in our experimental setup. The largest one has google/vit-base-patch16-224 as encoder and openai-community/gpt2 as decoder sizing to 3.8GB. The second largest one has google/vit-base-patch16-224 as encoder and distilbert/distilgpt2 as decoder sizing to 400MB. The smallest one has facebook/deit-tiny-patch16-224 as encoder and sshleifer/tiny-gpt2 as decoder. We trained all the models for around 6 hours.

The distilgpt model achieved superior performance than tinygpt despite being trained for just 3 epochs, and the gpt2 model almost matched the performance in just one epoch and got higher F1measure in 2 epochs. Hence, we found that larger model can get higher accuracy even when trained for fewer epochs.

Epoch	Training Loss	Validation Loss	Rouge2 Precision	Rouge2 Recall	Rouge2 Fmeasure
1	2.615400	2.395875	0.030200	0.272400	0.052600
2	2.135400	2.225292	0.035100	0.308900	0.061000

Figure 4.1: Largest Model (GPT2 decoder)

[3036/3036 5:29:25, Epoch 3/3]

Epoch	Training Loss	Validation Loss	Rouge2 Precision	Rouge2 Recall	Rouge2 Fmeasure
1	No log	2.681621	0.026200	0.222300	0.045700
2	3.382100	2.418616	0.030100	0.281100	0.053600
3	2.532800	2.346840	0.031800	0.297500	0.056800

Figure 4.2: Larger Model (Distilgpt2 decoder)

[9108/9108 5:17:34, Epoch 9/9]

Epoch	Training Loss	Validation Loss	Rouge2 Precision	Rouge2 Recall	Rouge2 Fmeasure
1	No log	10.720781	0.000100	0.000900	0.000100
2	10.785900	10.512541	0.002300	0.028800	0.004300
3	10.613000	10.305236	0.002700	0.028300	0.004800
4	10.399900	10.115464	0.003000	0.037000	0.005600
5	10.197700	9.952544	0.003100	0.036900	0.005700
6	10.019100	9.822090	0.003000	0.035900	0.005500
7	9.871900	9.727283	0.002700	0.027600	0.004800
8	9.759500	9.669962	0.003100	0.035700	0.005600
9	9.684900	9.650848	0.003100	0.035700	0.005600

Figure 4.3: Smaller Model (Tinygpt decoder)

4.2 Visualization using GradCam

Grad-CAM (Gradient-weighted Class Activation Mapping) was implemented to visualize the areas of an image that influence the captions generated by the VisionEncoderDecoder model. The model's activations and gradients were captured during the forward and backward passes, respectively, to identify which regions of the image contributed most significantly to the caption generation. Grad-CAM was applied by averaging the gradients across the spatial dimensions and combining them with the activation maps. This produced a heatmap that highlighted the important areas within the image. Finally, the original image was blended with the heatmap to create a visual representation of the model's focus, allowing for insights into how the model interprets visual information to generate descriptive text. The results were saved as PNG images, showcasing the correlation between the generated captions and the visual features identified by the model.

Figure 4.4: Image passed to model

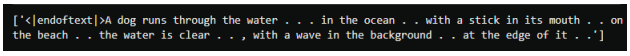

Figure 4.5: Caption Generated

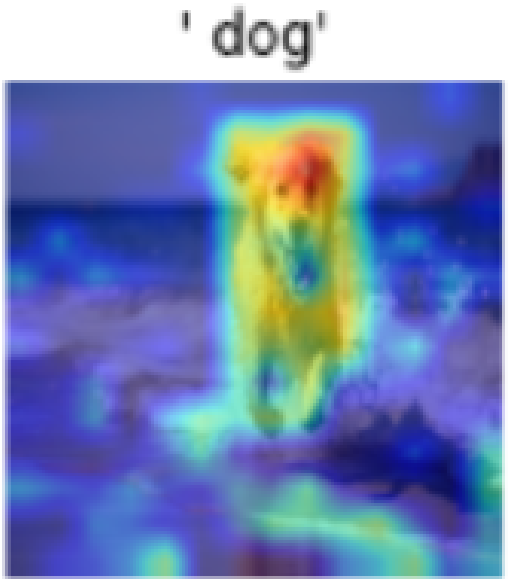

Figure 4.6: Activation map on the image when generating the word "dog"

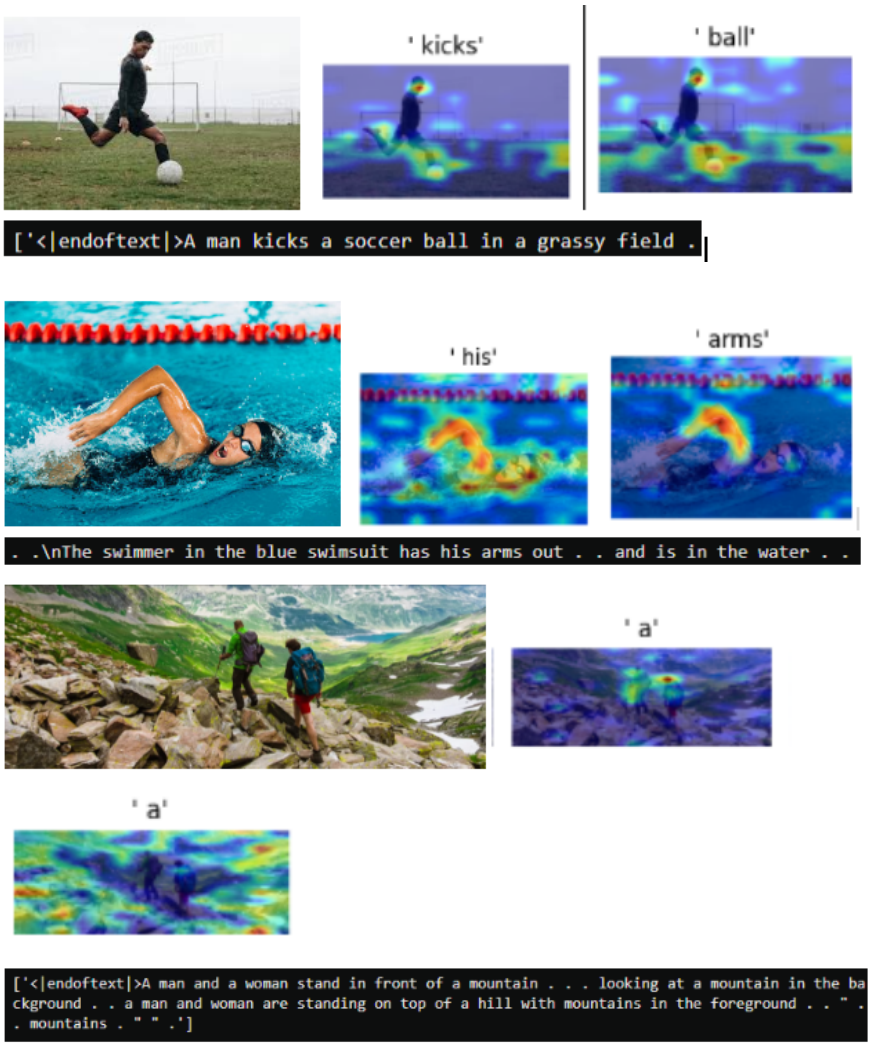

Figure 4.7: Image along with its caption generated by model and activation visualizations.

4.3 Image Search

Once the captions are generated, searching for images using textual descriptions becomes an NLP task. For searching, the embeddings of each of the captions are generated using the bert-base-uncased model. The sentence is first tokenized and the embedding for each token is generated using this pre-trained model. To obtain the sentence level embedding, the embedding of the CLS token is used since it captures the whole context of the sentence. For simplicity, we computed the cosine similarity between the embedding of the caption and the embedding of the textual description provided. If the similarity score crosses a threshold of 0.85, the image corresponding to the caption is displayed.

This approach performs quite well if the textual description provided is lengthy. But if the user provides the keywords associated with each image only, then the caption carries way more context than the provided description, so this approach fails. To overcome this issue, we also search for keywords provided by the user. If all the provided keywords are present in a caption, then the image corresponding to that caption is also shown.

4.4 Downsizing Models

The best model obtained by using vit as encoder and gpt2 as decoder is about 3.8GB which can be problematic to load in memory for edge devices. Furthermore, the weights are in float32 format which demands significant computational power. To decrease the memory and computational requirement, we utilized the BitsAndBytes API to quantize the weights of the model to 4-bit integers. This reduced the size of the model by 16x reaching final size of 250MB. Furthermore, integer arithmetic is more efficient and less demanding than floating point arithmetic. Hence, we obtain a smaller model with comparable accuracy via quantization.

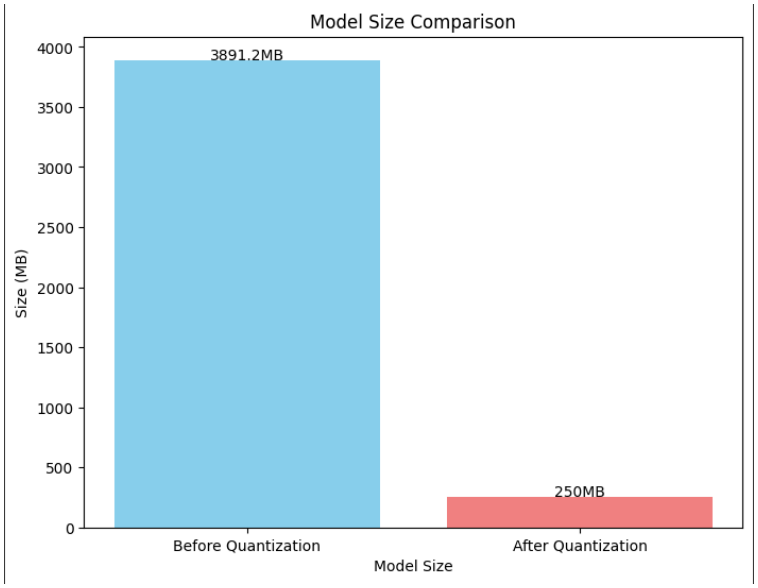

Figure 4.8: GPT2 + VIT Model size before and after quantization

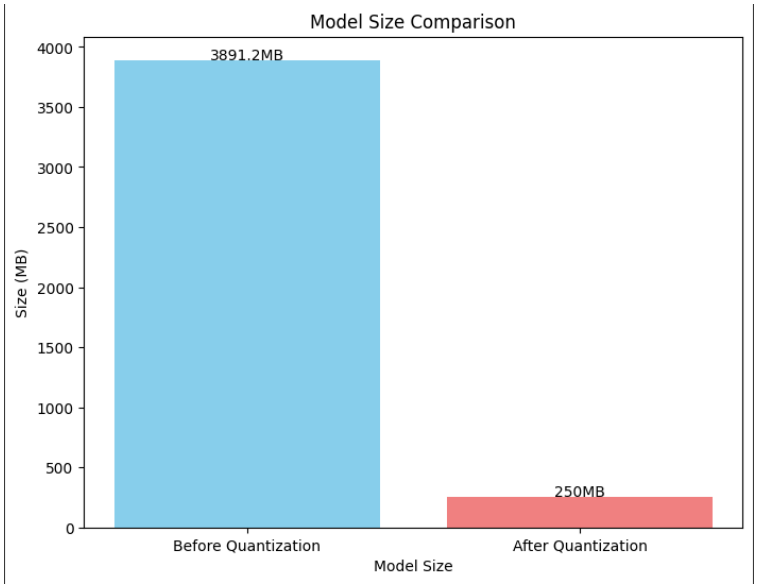

Figure 4.9: BERT Model size before and after quantization

4.4.1 Before and After Quantization Results

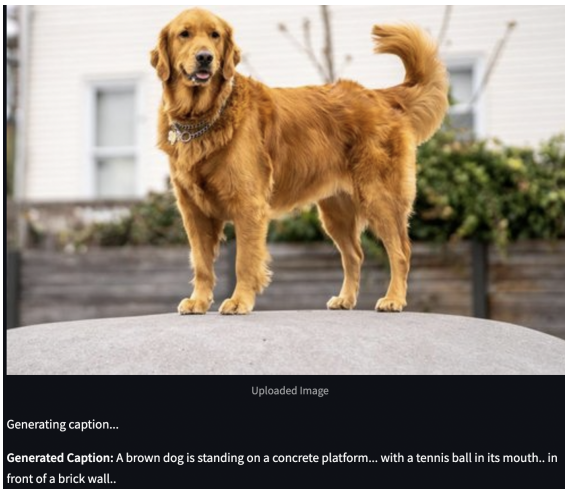

Figure 4.10: Result before Quantization

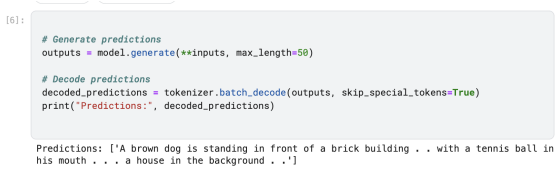

Figure 4.11: Result after Quantization

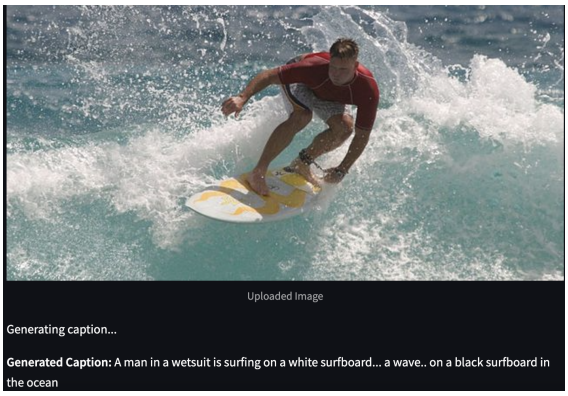

Figure 4.12: Result before Quantization

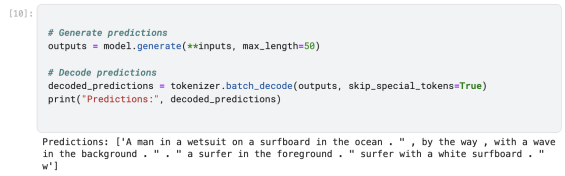

Figure 4.13: Result after Quantization

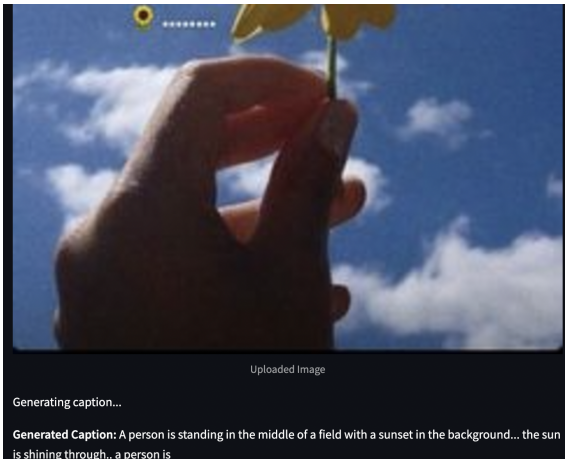

Figure 4.14: Result before Quantization

```
12]: # Generate predictions
     outputs = model.generate(**inputs, max_length=50)

     # Decode predictions
     decoded_predictions = tokenizer.batch_decode(outputs, skip_special_tokens=True)
     print("Predictions:", decoded_predictions)

Predictions: ['A person is standing in front of a beautiful sunset . . . the sun is shining through the air . . a person is sitting in the middle of it . . and there is a person in the foreground . . of the image . . with']
```

Figure 4.15: Result after Quantization

After quantization, the model occasionally hallucinates or generates incoherent sentences, though it still captures the overall semantics in the captions.

4.5 Frontend Interface

We used the streamlit library to implement a simple frontend interface from which we can access the captioning and search models.

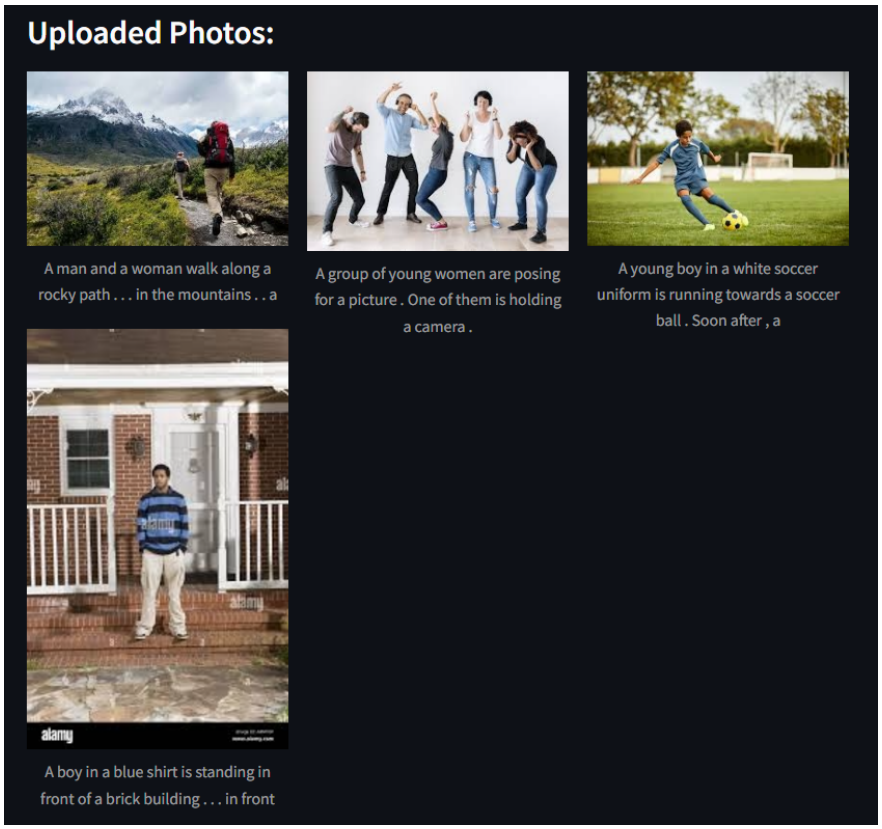

Figure 4.16: Uploaded Images along with their captions

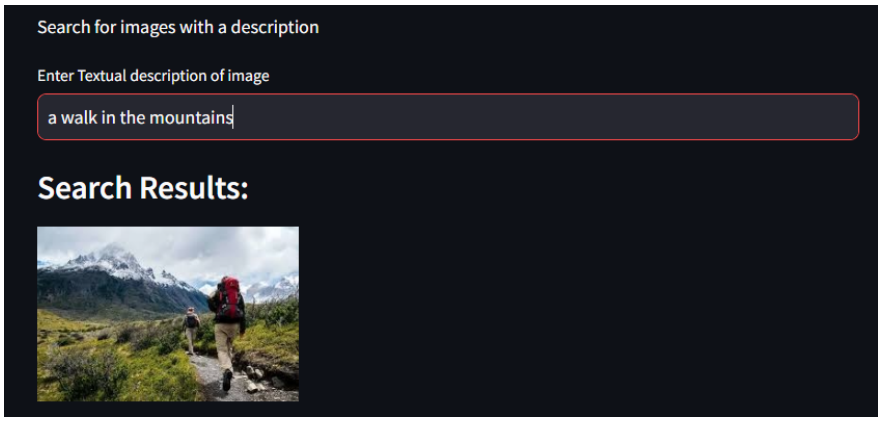

Figure 4.17: Search result

5. Conclusion

In conclusion, the system aims to bridge the gap between visual content and textual understanding. By utilizing state of the art image encoders and text encoders, our system aims to provide accurate and meaningful captions for images, enhancing both accessibility and searchability. Furthermore, models are quantized for efficiency in both storage and computation making them suitable for running in edge devices. The integration of advanced techniques in computer vision and natural language processing ensures that the system not only generates high-quality captions but also allows users to perform efficient and intuitive searches based on those captions. This capability is particularly valuable in a wide range of applications, from digital asset management and e-commerce to social media and content discovery.

References

[1] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator, 2015.

[2] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention, 2016.

[3] Wei Chen, Yu Liu, Weiping Wang, Erwin Bakker, Theodoros Georgiou, Paul Fieguth, Li Liu, and Michael S. Lew. Deep learning for instance retrieval: A survey, 2022.

[4] Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks, 2018.

[5] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir D. Bourdev. Compressing deep convolutional networks using vector quantization. *CoRR*, abs/1412.6115, 2014.

[6] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew G. Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. *CoRR*, abs/1712.05877, 2017.

[7] Ritchie Zhao, Yuwei Hu, Jordan Dotzel, Christopher De Sa, and Zhiru Zhang. Improving neural network quantization without retraining using outlier channel splitting. *CoRR*, abs/1901.09504, 2019.

[8] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. HAQ: hardware-aware automated quantization. *CoRR*, abs/1811.08886, 2018.