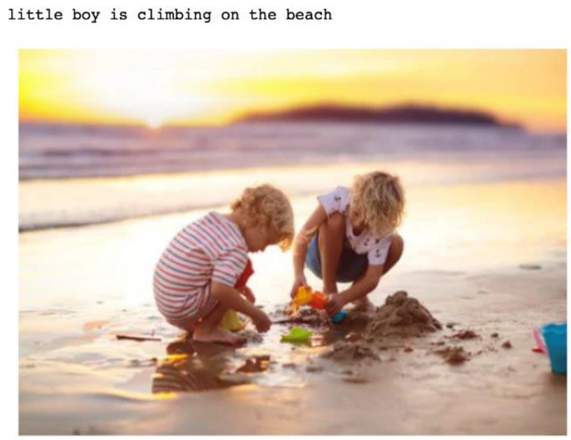
From Pixels to Words

Rishabh Agrawal(ra26) **Chiranjeevi Konduru (konduru4)** **Vansh Porwal (vporwal3)**

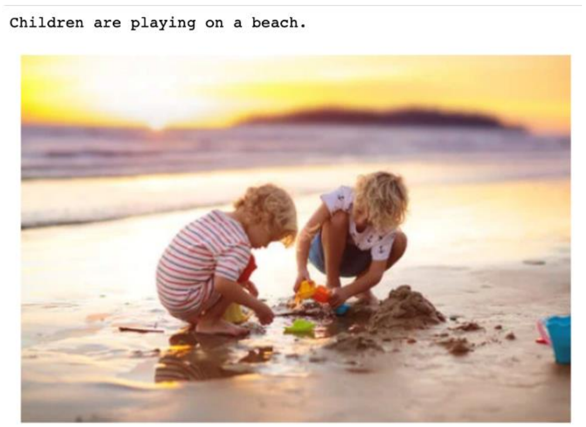
The objective of this project is to develop and compare a Deep learning model that gives the reasonable text descriptions for the provided image. Conventional architectures and CLIP with GPT2 integration is the main architecture that is used to solve this task and present the performance analysis of pre-trained and fine-tuned models.

Conventional vision-language models (VLMs) are trained on a dataset of text and images and can perform tasks such as image captioning, visual question answering, and visual dialogue. Our baseline model combines the VGG16 convolutional neural network (CNN) with Long Short-Term Memory (LSTM) networks. The VGG16 model extracts features from images, while the LSTM processes these features and generates a sequence of words to describe the image. Our main model, CLIPCap, uses CLIP (Contrastive Language-Image Pretraining) model and GPT-2. The CLIP model learns to associate images and text during pretraining, making it a strong candidate for image captioning tasks. CLIPCap is trained on a dataset of text and images, as well as a dataset of human-generated captions for images. This allows CLIPCap to learn the relationships between text and images, as well as the way that humans describe images. We integrate the GPT-2 model to generate coherent and contextually relevant captions based on the image features extracted by CLIP.

We train our baseline VGG16+LSTMs model on the Flickr8k dataset to learn the relationships between image content and natural language descriptions. For the main CLIPCap model, we fine-tune the pre-trained CLIP model to align with our same image captioning dataset.



```
little boy is climbing on the beach
```



```
Children are playing on a beach.
```

Our main model, CLIPCap with GPT-2 integration, demonstrates superior performance in the image captioning task (see 2nd image above) compared to the baseline VGG16+LSTMs model when evaluated on the Flickr8k dataset. The combination of CLIP's image-text understanding and GPT-2's natural language generation capabilities allows for more accurate and contextually rich captions.

Introduction

Image captioning is an interdisciplinary task that combines computer vision and natural language processing, aiming to generate a natural language description for a given image. The goal is to create models that can effectively analyze visual content and produce linguistically coherent sentences that capture the essence of the image, including objects, actions, and their relationships. This task has numerous practical applications, such as enhancing accessibility for visually impaired individuals, improving content discoverability on social media, and assisting in image indexing for search engines.

Over the years, various approaches to image captioning have been explored, from rule-based systems and statistical methods to deep learning techniques. The introduction of deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), has significantly improved the performance of image captioning systems. A widely adopted approach is to combine a CNN, such as VGG16 or Inception, with an RNN, like LSTMs or GRUs, to create end-to-end trainable image captioning models. More recently, attention mechanisms and large-scale pretrained models have been employed to further enhance the performance of these systems.

In this project, we will focus on two different approaches to image captioning. First, we will implement a VGG16+LSTM model, which combines the VGG16 CNN for feature extraction and an LSTM network for caption generation. This model serves as a baseline, representing a fundamental milestone in the development of image captioning techniques. Second, we will explore a more advanced approach by developing a CLIPCap model, which leverages the capabilities of OpenAI's CLIP (Contrastive Language-Image Pretraining) model, pretrained to associate images and text, and integrates it with a state-of-the-art language model, such as GPT-2.

Cross-entropy loss is a common loss function used in machine learning models, especially for classification tasks. In the context of models like CLIP, it's used to measure the dissimilarity between the predicted and actual values. CLIP employs a contrastive loss function, a variant of cross-entropy loss, to fine-tune alignment between image and text representations in a shared space. It is trained to enhance similarity between related image-text pairs and increase distance between unrelated ones. Assessing the performance of a trained model in image captioning can be quite challenging. To address this issue, several evaluation metrics have been developed. Common evaluation metrics found in the literature include BLEU, ROUGE-L, CIDEr, METEOR, and SPICE. BLEU, which stands for Bilingual Evaluation Understudy, is a widely used metric for evaluating text generation tasks.

The BLEU metric compares machine-generated text with one or more human-written reference texts, essentially measuring how closely the generated text aligns with the expected output. Although BLEU is primarily used in automated machine translation, it can also be applied to other tasks such as image captioning, text summarization, and speech recognition.

By comparing the performance of these two approaches based on the BLEU scores, we aim to gain insights into the advantages and limitations of each method and identify possible directions for future improvements in image captioning systems.

Overview of Approach

Baseline: VGG16+LSTMs

The VGG16+LSTM model merges the strengths of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to address image captioning challenges.

Image Model: The VGG16, a prominent CNN architecture, is employed to extract significant features from the input images. It contains 16 layers, including 13 convolutional layers and 3 fully connected layers. VGG16 is known for its excellent performance in image classification and recognition tasks, where it learns to identify and extract important features from images. This model pretrained on ImageNet to extract features from the input images. This dataset contains images from 1,000 different classes, making the pretrained VGG16 model capable of extracting robust and high-level features from a wide range of images. We removed the final classification layer (SoftMax) to obtain a feature vector for each image, which is then passed to the LSTM network for caption generation.

Language Model: We preprocessed the captions from the Flickr8k dataset by tokenizing the text, converting words to integer indices, and padding the sequences to a fixed length. We also create a mapping of integer indices to words for later decoding. Subsequently, these features are fed into an LSTM (Long Short-Term Memory) network. LSTMs are a type of RNN specifically designed to address the vanishing gradient problem, which occurs when training traditional RNNs. They are capable of learning and remembering long-range dependencies in sequences of data, making them suitable for tasks such as natural language processing. By processing the image features, the LSTM generates a series of words, ultimately forming a coherent caption that describes the image. The architecture is shown below fig 1.

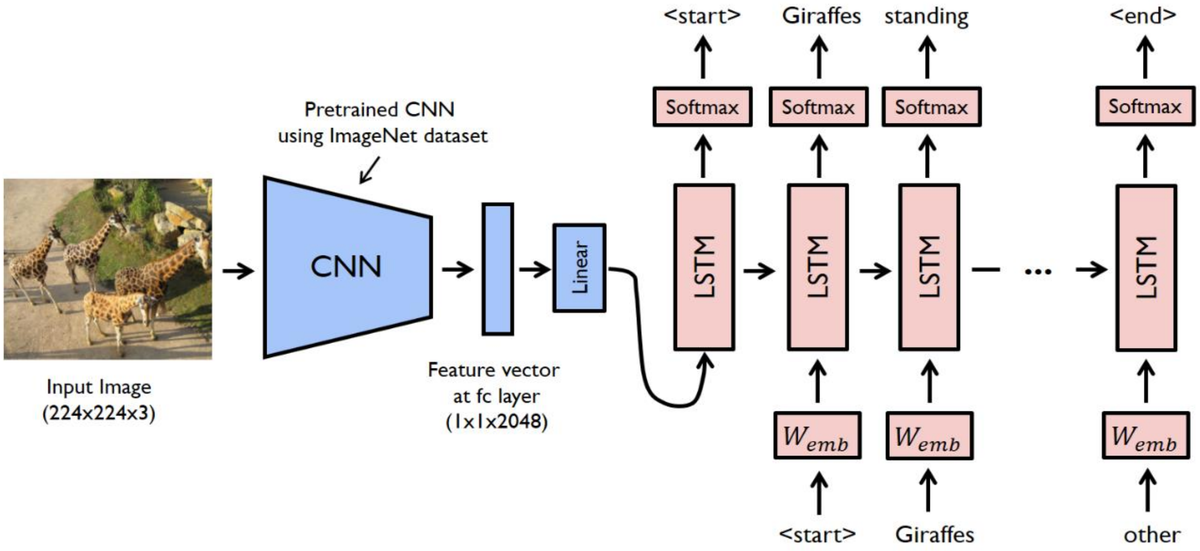
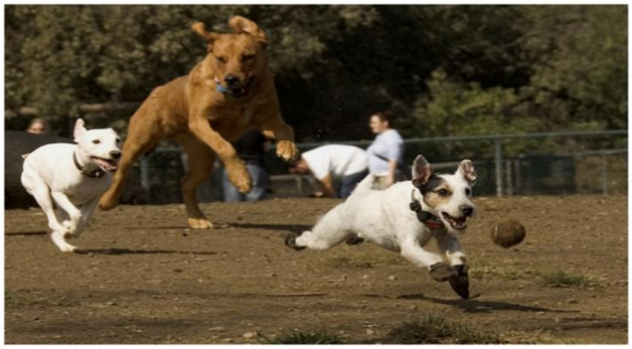


Fig 1: (VGG16 + LSTM)

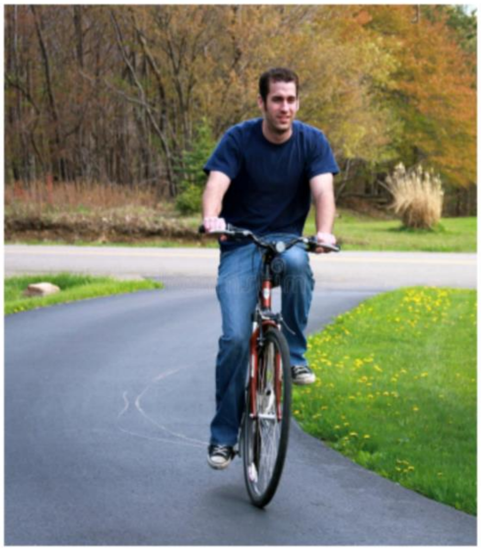
Training Details:

Learning_rate	$1e^{-3}$
optimizer	Adam
criterion	Cross Entropy
Epochs	20
dropout	0.5

white dog is running through the grass



man in red shirt is riding bike in the street



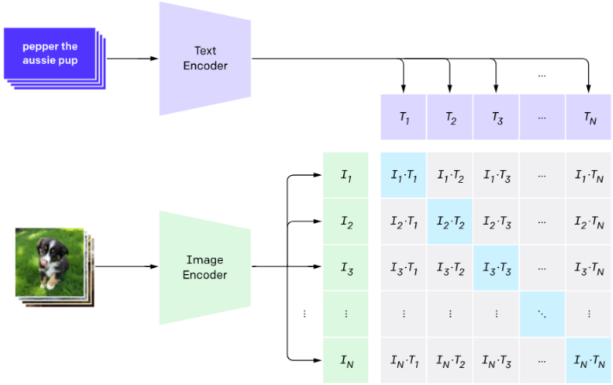
After several iterations of tuning and re-running, the model showed noticeable improvement. Initially, the image captions were entirely unrelated when the number of epochs was low. However, upon refining the hyperparameters and conducting further training, the model began to perform better. Examining the above images, it's clear that the model was able to identify the color of the dog, although it failed to accurately count the number of dogs. The caption for the second image, generated by the final VGG16 model, recognized all the objects in the image. However, it failed to capture the attributes of the objects, such as the color of the shirt, which was inaccurately depicted.

Main Model: CLIPCap (GPT2 Integration)

In contrast to the baseline model, the CLIPCap model harnesses the potential of OpenAI's CLIP (**Contrastive Language-Image Pretraining**) model, integrating it with advanced language models, such as GPT-2.

Image Model: CLIP is **pretrained** on a large-scale dataset containing images and their associated textual descriptions. This dataset is created by collecting images from the web and their corresponding text, such as captions, titles, or descriptive paragraphs. It is approximately trained on 400,000,000 (image, text) pairs. An (image, text) pair might be a picture and its caption.

This model uses a **contrastive learning** approach to understand the semantic relationship between images and text. During training, the model is presented with a batch of images and their corresponding textual descriptions. The goal is to learn to identify the correct image-text pair among a set of incorrect (negative) pairs. This is achieved by maximizing the similarity between the correct image-text pair and minimizing the similarity between the incorrect pairs. One of the unique capabilities of the CLIP model is its ability to perform **zero-shot learning**. This means that the model can generalize to new tasks without requiring any fine-tuning or task-specific training. For instance, given a set of image categories and their textual descriptions, the CLIP model can classify images into these categories even if it has not been explicitly trained to do so. This capability arises from the shared image-text embedding space learned during pretraining (see the figure below), which enables the model to transfer knowledge between vision and language tasks.



	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

Figure 2: Contrastive learning

We then use this CLIP model to extract visual features from the input images. The pretrained vision encoder was used to generate high-level visual features, which are semantically meaningful and can be used for caption generation.

Language Model: GPT2 or Generative Pre-trained Transformer 2 is Transformer based architecture with self-attention mechanism. This allows the model to weigh the importance of each word in the input sequence relative to the current word being processed. Unlike recurrent neural networks (RNNs), transformers can process input sequences in parallel rather than sequentially, making them more efficient and effective at handling long-range dependencies. We used GPT2- medium from Hugging Face, which is pretrained on a large-scale dataset called Web Text, which is derived from web pages collected by crawling the internet. Web Text contains a diverse range of content, including articles, blog posts, and web pages, covering various topics and styles. Fig 3 below represents the high-level view of how CLIPCap with GPT2 model works.

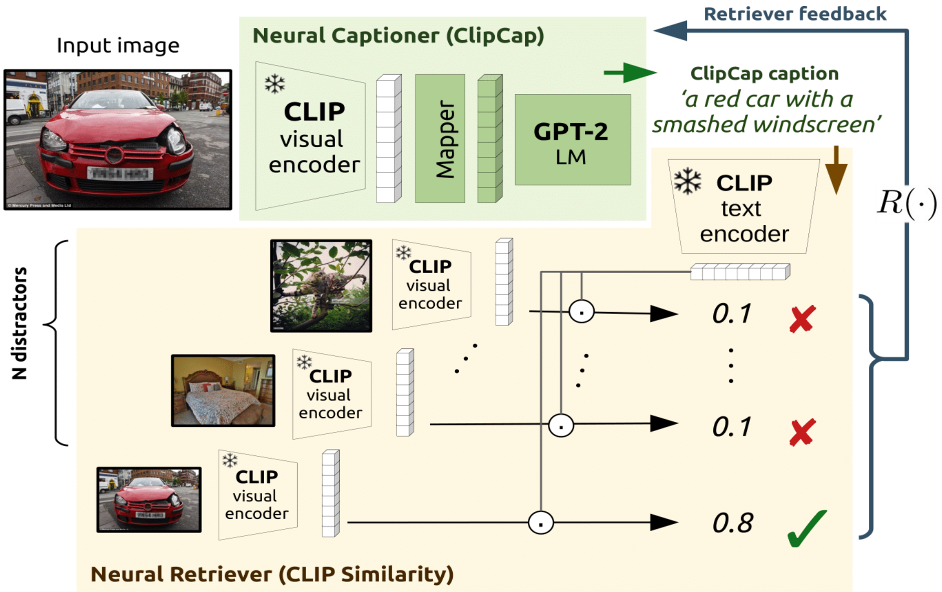


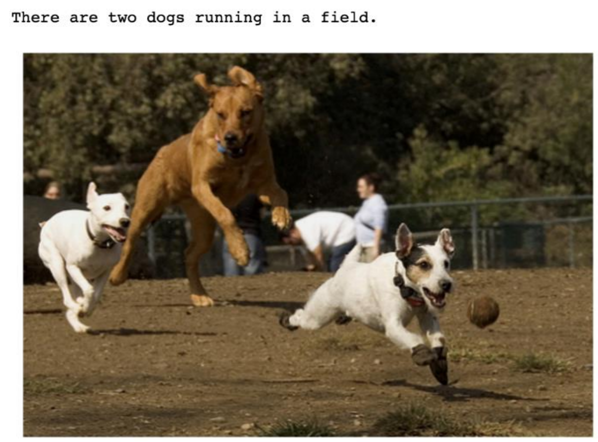
Figure 3: Overview of CLIPCap Architecture

Training Details:

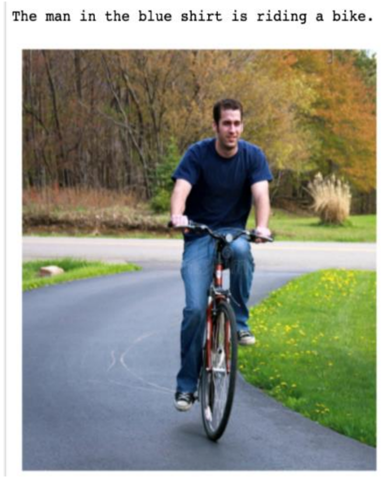
Learning_rate	$3e^{-3}$
optimizer	Adam
criterion	Cross Entropy
Epochs	150

dropout	0.1
max_len (maximum length of the generated text)	40
num_layers(Transformer encoder)	6
n_heads (no. of heads in multihead attention)	16

The hyperparameters specified above were utilized in the training of the CLIPCap model on the Flickr8K dataset. With a transformer encoder comprising six layers, this model is capable of learning intricate and abstract input data representations. The model demonstrated a high degree of accuracy, generating captions for images that were near perfect. This is largely attributed to the zero-shot learning capability of the model.



```
There are two dogs running in a field.
```



```
The man in the blue shirt is riding a bike.
```

Upon comparing it with the VGG+LSTM model, we can observe that the CLIPCap model produces remarkably better captions. However, it's worth noting that the model failed to accurately count the number of dogs in the first image and didn't capture the color of the dogs as accurately as the baseline model. On the contrary, in the case of the second image, the model successfully discerned all the details and generated captions that were more accurate and detailed than those from the baseline model.

Results

The provided table presents BLEU scores for two distinct models: VGG16+LSTM and CLIPCap. These scores are used to evaluate the performance of each model in a task, likely related to language translation or text generation. For each BLEU metric (BLEU-1, BLEU-2, BLEU-3, BLEU-4), the CLIPCap model outperforms the VGG16+LSTM model.

	VGG16+LSTM	**CLIPCap**
BLEU-1	0.51	0.61
BLEU-2	0.27	0.46
BLEU-3	0.19	0.38
BLEU-4	0.09	0.28

In summary, the CLIPCap model consistently outperforms the VGG16+LSTM model across all BLEU scores. These results clearly suggest that CLIPCap generally produces outputs that are more similar to the reference translations or texts, taking into account both individual words and their order.

Interesting Scenarios:

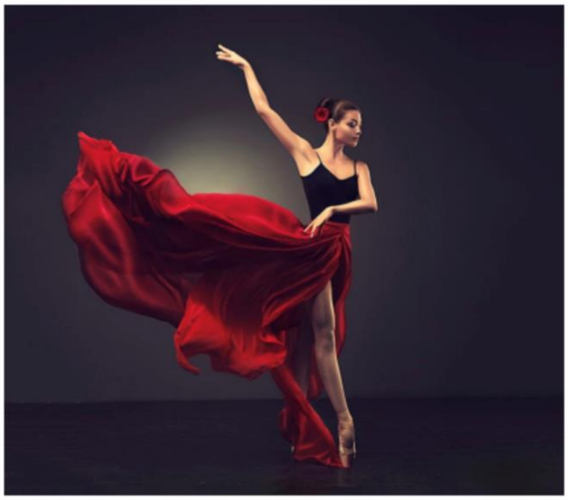
Case 1: In certain instances, the captions produced by both models were strikingly similar. This typically occurred in situations such as those involving straightforward and uncomplicated image scenes, commonly observed objects, and high-resolution images. The image sample provided below serves as an illustration of this circumstance.

```
dog is running on grass
```

```
One dog is running through a field.
```

Case 2: In certain situations, the CLIPCap model significantly surpassed the performance of the VGG+LSTM model. As demonstrated by the image samples provided below, this superior performance could be attributed to a couple of factors: the handling of abstract concepts and the model's capacity for zero-shot learning. The training data for CLIPCap is more varied, and the model's architectural design enables it to generalize to classes unseen during training.

```
two men are standing in front of the sun
```



```
"Lady in red dress dancing on the street. "
```

Discussion and conclusions

Both the VGG16+LSTM and CLIPCap models have shown promising results in various image captioning tasks, such as aiding the visually impaired, enhancing social media content, and facilitating image indexing. However, these two models represent distinct phases in the progression of image captioning techniques. VGG16+LSTM serves as a fundamental benchmark in the field, while CLIPCap symbolizes an advanced method that leverages pretrained models and sophisticated language models for superior performance.

A key factor contributing to this performance difference could be the training approach. CLIPCap benefits from being trained on an extensive dataset of text-image pairs, which allows it to gain a comprehensive understanding of both language and visual concepts.

Furthermore, the CLIP model's inherent design for simultaneous understanding of images and text enables it to capture the relationships between visual and textual information. This multimodal comprehension can be instrumental in producing more precise and contextually appropriate text when provided with a visual input.

Statement of individual contribution

Rishabh Agrawal: Refined the pre-existing VGG16+LSTM model through the application of various optimizers and precise hyperparameter adjustments. composed the VGG16 image-model code to develop the baseline model. Wrote the VGG16 image-model code to create the basic model. Clearly explained the architecture and approach used for the VGG16+LSTM model.

Vansh Porwal: Developed code for the Language model (LSTM) in the basic model. Handled data preparation, training, and improvement for the CLIPCap Model. Tweaked settings and evaluated the CLIP model's performance. Thoroughly described the approach used for implementing and training the CLIPCap Model.

Chiranjeevi Konduru: Orchestrated the setup of the comprehensive VGG16+LSTM and CLIP models on the Google Cloud Platform for project execution. Oversee the image data preprocessing for the baseline model. Developed the integration code for the CLIP model with GPT2. Authored the project report, including results and a comparative analysis by highlighting the key differences in the results produced by the models.

References

VGG16+LSTM (https://github.com/anunay999/image_captioning_vgg16)

CLIP: Connecting text and Images (https://openai.com/research/clip)

CLIP_prefix_caption (https://github.com/rmokady/CLIP_prefix_caption)

Flicker8k Dataset (https://paperswithcode.com/dataset/flickr-8k)

Fig 3 source: *Cross-Domain Image Captioning with Discriminative Finetuning* (Roberto Dessì and Michele Bevilacqua and Eleonora Gualdoni, 2023)