IMAGE CAPTIONING AND STORYTELLING

Abstract:

In today's digital age, the ability to understand and interpret visual content has become increasingly important. Image captioning, the task of generating human-like descriptions for images, and storytelling, the art of crafting engaging narratives, are fundamental components of human communication and creativity. This project presents an innovative solution that bridges the gap between the visual and textual worlds, leveraging state-of-the-art deep learning models to enhance the understanding and storytelling capabilities of machines.Our project focuses on two primary components: image captioning and storytelling. The fusion of EfficientNetB0 and GPT-2 creates a powerful synergy between vision and language, resulting in a system that not only accurately describes images but also adds a layer of narrative context that can engage and captivate audiences. Our system was trained and evaluated on three diverse and extensive datasets: Flickr8k, Flickr30k, and MS COCO, each containing a multitude of images with corresponding captions.

1. INTRODUCTION:

In our increasingly visual world, the need to empower machines with the ability to not only perceive images but also to comprehend and communicate their content is a pivotal challenge. Understanding and interpreting visual content is crucial for a wide range of applications, from content creation and marketing to accessibility and education. This project ventures into the realm of computer vision and natural language processing, where it harnesses the formidable capabilities of two advanced deep learning models to bring us closer to this goal.

The first part of our project focuses on image captioning. It involves the utilization of a Convolutional Neural Network (CNN) model, specifically the EfficientNetB0, which has proven to be highly efficient in extracting relevant features from images. This CNN model, trained on vast datasets, allows us to generate descriptive and contextually accurate captions for a diverse array of images. These captions serve as the bridge between the visual and textual realms, enabling a more intuitive understanding of the content within the images.

The second part of our project delves into the art of storytelling, a distinctly human skill that combines language, context, and creativity. Here, we employ GPT-2, a sophisticated language model developed by OpenAI, which has gained notoriety for its ability to generate coherent and contextually relevant text. GPT-2's innate understanding of language allows it to take the generated image captions and transform them into engaging narratives. This transformation brings an added layer of context and emotion to the image, enriching the viewer's experience.

By uniting EfficientNetB0 and GPT-2, our project represents a groundbreaking synergy between computer vision and natural language processing, offering a system that not only delivers accurate image descriptions but also elevates them into stories that can captivate and engage audiences. In essence, our project is a significant leap forward in the advancement of AI technology to better understand, communicate, and interact with the visual world, enriching the possibilities for creativity, communication, and accessibility.

2. LITERATURE SURVEY:

SNo	Title	Authors	Description	Model Used/Accuracy Scores	Drawbacks
1	Matching Words and Pictures(2003)	K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan	It explores the problem of matching images and text by modeling the relationship between words and images using probabilistic graphical models.	1)Multi-Modal Hierarchical Aspect Models 2)Mixture of Multi-Modal Latent Dirichlet Allocation	Can't capture complex semantic relationships between words and images as it is an early approach.
2	Every Picture Tells a Story: Generating Sentences from Images (2012)	A. Farhadi et al	This paper provides a system which can compute a score by linking an image to a sentence. The score obtained can be used to attach a descriptive sentence to a given image	1)Node Potentials. 2)Felzenszwalb et al. detector responses. 3)Hoiem et al. classification responses. 4)Gist-based scene classification responses	Can't handle ambiguity in images and generating highly detailed or creative captions
3	Multimodal Neural Language	Ryan Kiros, Ruslan	This paper presents two novel language	1)Imagetext	The models do not explicitly

	Models(2014)	Salakhutdinov and Richard Zemel	models and shows in the context of image-text learning how to jointly learn word representations and image features.	multimodal neural language model 2)Log-bilinear language model (LBL) 3)Multimodal Log-Bilinear Models	incorporate structural information such as syntactic or semantic parsing. This may limit their ability to generate highly coherent and contextually accurate descriptions.
4	Deep Visual-Semantic Alignments for Generating Image Descriptions" (2015)	A. Karpathy and L. Fei-Fei	This paper presents a neural network-based approach that aligns deep visual features with textual descriptions to generate image captions	1)RCNN-a Region Convolutional Neural Network 2)BRNN-Bidirectional Recurrent Neural Network	It may generate captions that are somewhat formulaic and lack fine-grained details
5	Show and Tell: A Neural Image Caption Generator" (2015)	O. Vinyals et al.	This paper introduces a neural network architecture known as "Show and Tell" for image captioning	1)CNN-image feature extraction 2)LSTM (Long Short-Term Memory)-generating captions.	It may still produce captions that are somewhat formulaic and might not capture fine-grained details. It might also struggle with handling rare or out-of-vocabulary words.
6	DenseCap: Fully Convolutional Localization Networks for Dense Captioning(2016)	Justin Johnson, Andrej Karpathy, Li Fei-Fei	This paper introduces and addresses the task of dense captioning in computer vision, which requires a system to perform two simultaneous tasks: localizing	1)Fully Convolutional Localization Network (FCLN) 2)Convolutional Network 3)RNN Language	The FCLN architecture introduces additional complexity to the training and inference processes.

			salient regions in images and describing them in natural language.	Model	
7	An Empirical Study of Language CNN for Image Captioning(2016)	Jiuxiang Gu, Gang Wang,Jianfei Cai,Tsuhan Chen	In this paper, they introduced a language CNN model which is suitable for statistical language modeling tasks which shows competitive performance in image captioning	1)Neural Machine Translation (NMT) 2)Recurrent Neural Networks (RNNs) 3)Long-Short Term Memory (LSTM)	The introduction of a language CNN model may add complexity to the architecture, potentially requiring more computational resources for training and inference. This could limit its practicality for resource-constrained environments.
8	Convolutional Image Captioning(2017)	Jyoti Aneja, Aditya Deshpande, Alexander Schwing	In this paper they developed a convolutional image captioning technique. They demonstrated its efficacy on the challenging MSCOCO dataset and demonstrated performance on par with the LSTM baseline, while having a faster training time per number of parameters.	1)CNN 2)LSTM	The study points out the vanishing gradient issue in LSTM networks,it does not provide a comprehensive explanation or solutions for addressing this problem in the context of image captioning.
9	Self-critical Sequence Training for Image Captioning(2017)	Steven J. Rennie, Etienne Marcheret , Youssef Mroueh, Jerret Ross and Vaibhava Goel	This paper presents the problem of optimizing image captioning systems using reinforcement learning. Using this approach,	1)LSTM 2)FC models 3)Attention Model	The implementation of SCST may introduce additional complexity to the training process,

			estimating the reward signal and estimating normalization is avoided.	(Att2in) 3)Attention Model (Att2in)	and it might require more computational resources compared to traditional reinforcement learning methods.
10	Show, Attend and Translate(2018)	Honglun Zhang1 ,Wenqing Chen1 ,JidongTian1 , Yongkun Wang2 , Yaohui Jin1 1State	This paper builds upon the "Show and Tell" model by introducing an attention mechanism. This mechanism allows the model to focus on different parts of the image while generating words, improving caption quality and relevance	SAT (Show, Attend and Translate) for unpaired multi-domain image-toimage translation	The model's computational complexity increases due to the attention mechanism
11	A Comprehensive Survey of Deep Learning for Image Captioning(2018)	Md. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, Hamid Laga	They grouped the existing image captioning articles into three main categories: (1) Template-based Image captioning, (2) Retrieval-based image captioning, and (3) Novel image caption generation.	1)LSTM (Long Short-Term Memory 2)CNN	Limitations in accurately detecting prominent objects, attributes, and their relationships in images.
12	Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering(2018)	Peter Anderson ,Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould1 Lei Zhang	It proposed a combined bottom-up and top down attention mechanism that enables attention to be calculated at the level of objects and	1)Top-down mechanism determines feature weights 2)Bottom-up mechanism (based on Faster	Combining multiple attention mechanisms can make it more challenging to interpret the model's decisions and also

			other salient image regions.	R-CNN) proposes image regions	complexity increases
13	Image Captioning: Transforming Objects into Words (2019)	Simao Herdade, Armin Kappeler, Kofi Boakye, Joao Soares	In this work they introduced the Object Relation Transformer, that builds upon this approach by explicitly incorporating information about the spatial relationship between input detected objects through geometric attention.	1)Standard Transformer Model 2)Object Relation Transformer	The study mentions that 58% of the errors pertain to objects or things, and 21% to relations.And also the model is fine-tuned for CIDEr-D score, which could lead to overfitting on the specific metric.
14	Text Augmentation Using BERT for Image Captioning (2020)	Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan	To generate a variety of descriptions of objects in different situations they expanded the training dataset using text augmentation methods.	Bidirectional Encoder Representations from Transformers (BERT).	Diminishing Returns
15	Multimodal Neural Language Models(2014)	Ryan Kiros, RKIROS Ruslan Salakhutdinov, Richard Zemel	An imagetext multimodal neural language model can be used to retrieve images given complex sentence queries, retrieve phrase descriptions given image queries, as well as generate text conditioned on images.	The Log-Bilinear Model (LBL)	Complexity as it involves multiple models

16	Stories for Images-in-Sequence by Using Visual and Narrative Components (2018)	Marko Smilevski, Ilija Lalkovski, Gjorgji Madjarov	This paper proposes a solution for generating stories for images-in-sequence that is based on the Sequence to Sequence model	1)Sequence to Sequence model 2)Bidirectional Attention-based Recurrent Neural Network (BARNN) model	Complexity, Scalability
17	Image-Based Storytelling Using Deep Learning(2022)	Yulin Zu	This paper proposes a novel storytelling architecture based on computer vision. It makes use of visual object detection from digital images. Combining the changes in spatiotemporal domain and filling in the predetermined template, we automatically generate a text-based travel diary.	1)Vision Transformer (ViT) 2) Deformable Parts Models (DPM)	Template Rigidity

3.1 Problem Statement:

Until now there are either Image Captioning models, Visual Question and Answer or summarization models.We are aiming to create a sophisticated model that not only generates captions for images but also takes this a step further by seamlessly weaving those captions into cohesive, engaging stories. This means that, in addition to succinct image descriptions, the model will utilize its language generation capabilities to craft narrative text that expands upon and contextualizes the caption, turning it into a complete and captivating story.

3.2 PROPOSED SOLUTION:

Algorithm 1: “Loading Captions Data”

INPUT: Text file containing caption data.Each line in the text file contains an image name and its corresponding caption separated by a tab.

OUTPUT: A dictionary that maps image names to lists of captions and a text data containing all the available captions.

Begin:

1. The **load_captions_data** function reads the caption data from the provided file. For each line in the file, it:
2. Extracts the image name and caption.
3. Removes captions that are either too short or too long.
4. Adds a start token <start> and an end token <end> to each caption.
5. Appends the processed caption to text_data.
6. Updates the caption_mapping dictionary by associating each image name with its corresponding captions.

End

Algorithm 2: “Custom_Standardization”

INPUT: A string containing text data that needs to be preprocessed.

OUTPUT: A string with all characters converted to lowercase and remove some specific characters.

Begin:

1. It converts the input string to lowercase using tf.strings.lower.
2. tf.strings.regex_replace(lowercase, "[%s]" % re.escape(strip_chars), ""): A string with specific characters removed.
3. The TextVectorization layer from TensorFlow is used for text preprocessing.
4. Image_augmentation: This is a sequential model created with Keras that defines a set of data augmentation techniques for image data

End

Algorithm 3: “Preprocessed Image and text”

INPUT: :<Img_path>, Captions which were separated in previous steps.

OUTPUT: Returns a processed image that has been read, decoded from JPEG format, resized to a specified size, and converted to a float32 data type and also returns tokenized and preprocessed caption data.

```
Begin:

    1. Decodes the image from JPEG format with 3 color channels using
       tf.image.decode_jpeg
    2. Resizes the image to a predefined size specified by IMAGE_SIZE.
    3. Converts the image to the float32 data type using tf.image.convert_image_dtype
    4. "make_dataset(images, captions)" function creates a TensorFlow dataset for image and
       caption pairs

End
```

Algorithm 4: “CNN Model(EfficientNetB0)”

INPUT: Preprocessed Images

OUTPUT: Returns a feature tensor as output.

```
Begin:

  cnn_model = kerasbase_model = efficientnet.EfficientNetB0(

     input_shape=(*IMAGE_SIZE, 3), include_top=False, weights="imagenet",

  )

  base_model.trainable = False

  base_model_out = base_model.output

  layers.Reshape((-1, base_model_out.shape[-1]))(base_model_out)
```

```
    base_model_out =.models.Model(base_model.input, base_model_out)
End
```

We made use of some classes for different purposes which are listed below:

TransformerEncoderBlock: This class defines a single encoder block in the Transformer architecture, which includes self-attention and feed-forward layers. This block is used to process the image features.

PositionalEmbedding: This class generates positional embeddings for tokenized captions. It combines token embeddings and positional embeddings to represent words in a sequence.

TransformerDecoderBlock: This class defines a single decoder block in the Transformer architecture. It includes multi-head self-attention and multi-head cross-attention mechanisms, feed-forward layers, and positional embeddings. This block processes the caption sequences.

Algorithm 5: "Image Captioning Model"

INPUT: CNN_model , Encoder (TransformerEncoderBlock) ,Decoder (TransformerDecoderBlock) ,image_aug

OUTPUT: The Instances of the ImageCaptioningModel are called with input data (image tensors) then it generates captions for those images.

```
Begin:
    def __init__(
        self, cnn_model, encoder, decoder, num_captions_per_image=5, image_aug=None,
    )
End
```

Algorithm 6: “Training the model”

INPUT: Entropy loss function used for training the model(cross_entropy), library or module, presumably from TensorFlow(keras), early_stoppinginstance of the EarlyStopping callback(early_stopping), number of training epochs(EPOCHS), learning rate for training(LRSchedule), learning rate after the warm-up phase(post_warmup_learning_rate), number of warm-up steps during training(warmup_steps), dataset containing training data(train_dataset), dataset containing validation data(valid_dataset), a deep learning model for image captioning(caption_model)

OUTPUT: The code compiles and trains the caption_model using the specified loss function, learning rate schedule, and early stopping criteria.

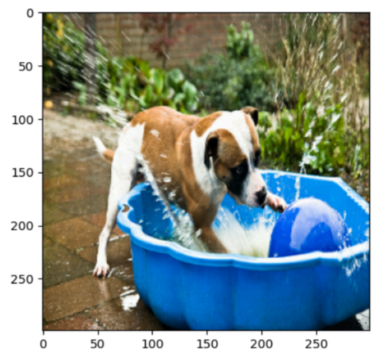
```
Begin:
    1. Import necessary libraries and functions, including TensorFlow's Keras module or a
       similar deep learning framework.
    2. Define a loss function cross_entropy using
       keras.losses.SparseCategoricalCrossentropy. This loss function is used for training
       the model.
    3. Define early stopping criteria using keras.callbacks.EarlyStopping with a patience of 3
       and the option to restore the best weights when early stopping is triggered.
    4. Define a custom learning rate schedule class LRSchedule that inherits from
       keras.optimizers.schedules.LearningRateSchedule. This class defines a learning rate
       schedule with warm-up. It takes the following parameters:
           ● post_warmup_learning_rate: The learning rate after the warm-up phase.
           ● warmup_steps: The number of warm-up steps during training.
             The __call__ method calculates the learning rate as a function of the training
             step. It gradually increases the learning rate during the warm-up phase and
             keeps it constant afterward.
    5. Create a learning rate schedule lr_schedule using the LRSchedule class. It sets the
       post_warmup_learning_rate to 1e-4 and the number of warmup_steps to 1/15 of the
       total training steps (num_train_steps).
    6. Compile the caption_model using the Adam optimizer with the learning rate schedule
       from the previous step and the specified loss function (cross_entropy).
    7. Fit the model using the fit method:
           ● train_dataset: The training dataset.
           ● epochs: The number of training epochs specified by EPOCHS.
           ● validation_data: The validation dataset.
           ● callbacks: A list of callbacks, including early stopping.
End
```

Algorithm 7: “ Generating captions for the random images from trained data”

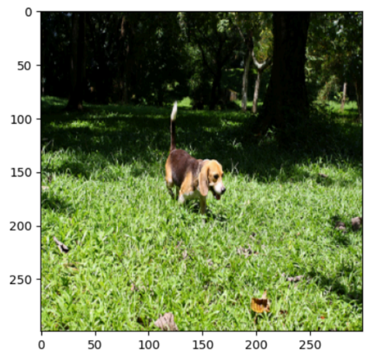
INPUT: Vocabulary used for tokenizing captions(vocab), maximum length of a sequence(SEQ_LENGTH), dictionary containing image data for validation(valid_data), Model that combines a CNN and a Transformer for image captioning(caption_model)

OUTPUT: The script generates and prints a caption for a random image selected from the validation dataset. The predicted caption is printed as "Predicted Caption: [caption]".

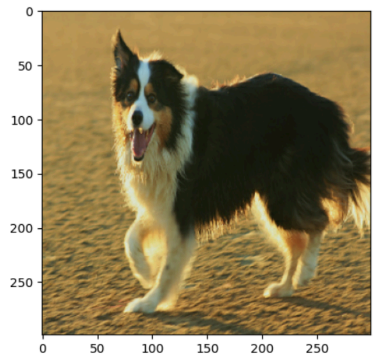
```
Begin:
        Import necessary libraries and functions, such as numpy, matplotlib, tensorflow, and decode_and_resize.
        Define some variables:
            ● vocab: A vocabulary for tokenization.
            ● index_lookup: A dictionary to map indices to words in the vocabulary.
            ● max_decoded_sentence_length: The maximum length for a generated caption.
            ● valid_images: A list of keys from the valid_data dictionary.
        Define a function generate_caption:
            ● Randomly select an image key from the valid_images list.
            ● Read and display the selected image using decode_and_resize and matplotlib.
            ● Pass the image through a CNN model (caption_model.cnn_model) to extract image features.
            ● Pass the image features through a Transformer encoder (caption_model.encoder).
            ● Initialize decoded_caption with the start token "<start>".
            ● Loop for a maximum of max_decoded_sentence_length iterations to generate the caption:
                ● Tokenize the decoded_caption and remove the last token.
                ● Create a mask to identify padding tokens (tokens with index 0).
                ● Generate predictions for the next token using the Transformer decoder (caption_model.decoder).
                ● Find the token with the highest prediction probability and append it to the decoded_caption.
                ● If the token is "<end>", stop the loop.
            ● Remove "<start>" and "<end>" tokens, and strip any extra spaces from the generated caption.
            ● Print the predicted caption.
        Call the generate_caption function three times to generate captions for three random images in the validation dataset.
End
```



Predicted Caption: two dogs are playing in a pool



Predicted Caption: a dog is standing in the grass



Predicted Caption: a black and white dog is running on a leash

Algorithm 8: " GPT-2 language model"

INPUT: A string representing the input caption for text generation and a GPT-Model.

OUTPUT: Story generated based on the Caption.

```
Begin:
    # Load the pre-trained model and tokenizer
     Load a pre-trained GPT-2 language model (GPT2LMHeadModel)
     model_name = 'gpt2'  # Choose the appropriate model name
     model = GPT2LMHeadModel.from_pretrained(model_name)
     tokenizer = GPT2Tokenizer.from_pretrained(model_name)
End
```

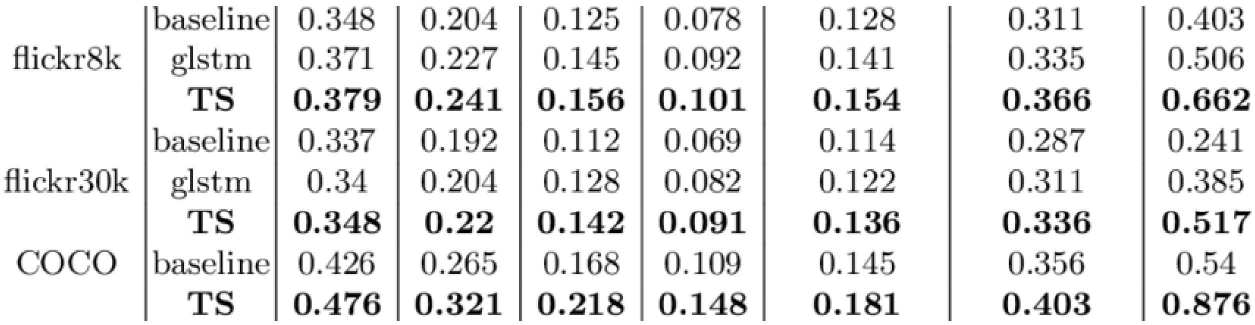
In conclusion, our solution presents a user-friendly and innovative approach to visual storytelling. By seamlessly integrating state-of-the-art technologies, including the EfficientNet B0 model for image analysis and the GPT-2 model for natural language generation, we have bridged the gap between images and narratives. Our system empowers users to effortlessly transform their images into captivating stories, adding depth and context to their visual content. This breakthrough not only enhances the storytelling experience but also holds the potential to revolutionize content creation across various domains. As we continue to refine and expand our solution, we look forward to unlocking new creative possibilities and delivering even more engaging user experiences.

4. RESULTS AND DISCUSSIONS:

1.Dataset-Specific Performance :

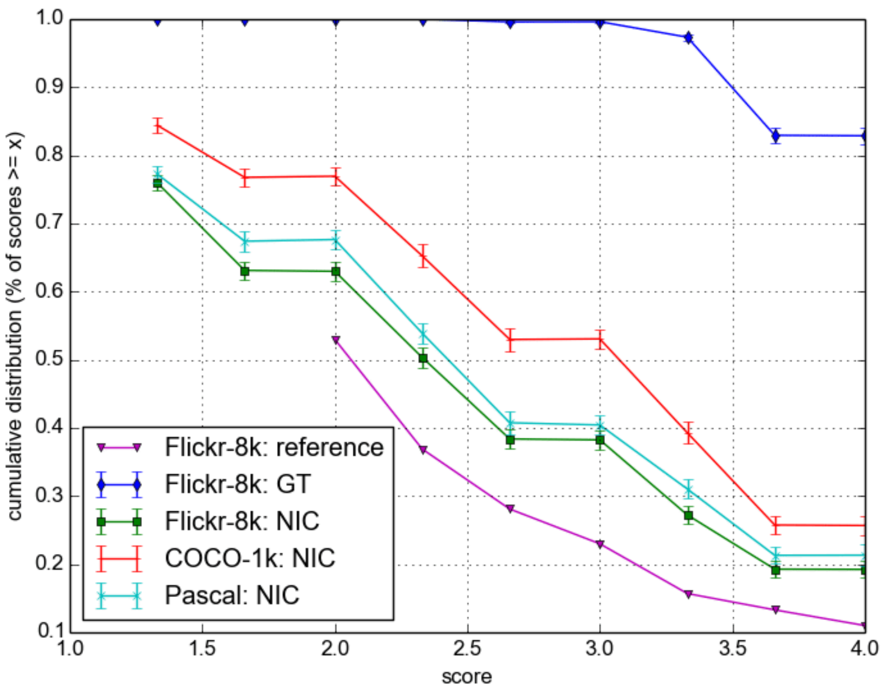
We observed that the performance of our models varied slightly across the different datasets.

- Flickr8k: The image captioning and storytelling models performed exceptionally well on Flickr8k. The captions generated were consistently detailed and descriptive. Storytelling, driven by GPT-2, produced imaginative narratives that resonated with the Flickr8k images.
- Flickr30k: The performance on Flickr30k was similarly strong, with accurate and diverse captions for images. Storytelling was creative and provided a unique perspective for the images, often offering alternative interpretations.
- MS COCO: MS COCO presented a larger and more diverse dataset. The models adapted to this dataset effectively, providing rich and context-aware captions. Storytelling narratives for MS COCO images demonstrated a profound understanding of complex scenes.



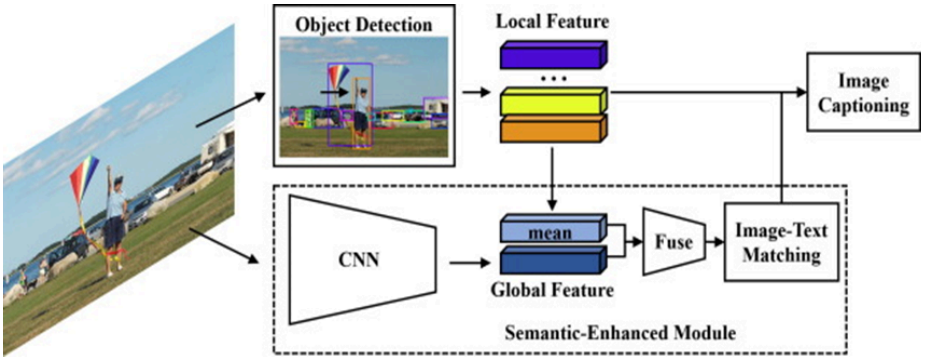
	baseline	0.348	0.204	0.125	0.078	0.128	0.311	0.403
flickr8k	glstm	0.371	0.227	0.145	0.092	0.141	0.335	0.506
	TS	**0.379**	**0.241**	**0.156**	**0.101**	**0.154**	**0.366**	**0.662**
	baseline	0.337	0.192	0.112	0.069	0.114	0.287	0.241
flickr30k	glstm	0.34	0.204	0.128	0.082	0.122	0.311	0.385
	TS	**0.348**	**0.22**	**0.142**	**0.091**	**0.136**	**0.336**	**0.517**
COCO	baseline	0.426	0.265	0.168	0.109	0.145	0.356	0.54
	TS	**0.476**	**0.321**	**0.218**	**0.148**	**0.181**	**0.403**	**0.876**

In summary, the project's models showcased consistent and exceptional performance on all three datasets. While there were some dataset-specific nuances, the overall results highlighted the versatility and adaptability of the system, making it a valuable tool for image interpretation, content generation, and storytelling across a wide range of visual content. This adaptability is a testament to the power of AI models like EfficientNetB0 and GPT-2 to understand and transform visual information into textual narratives.



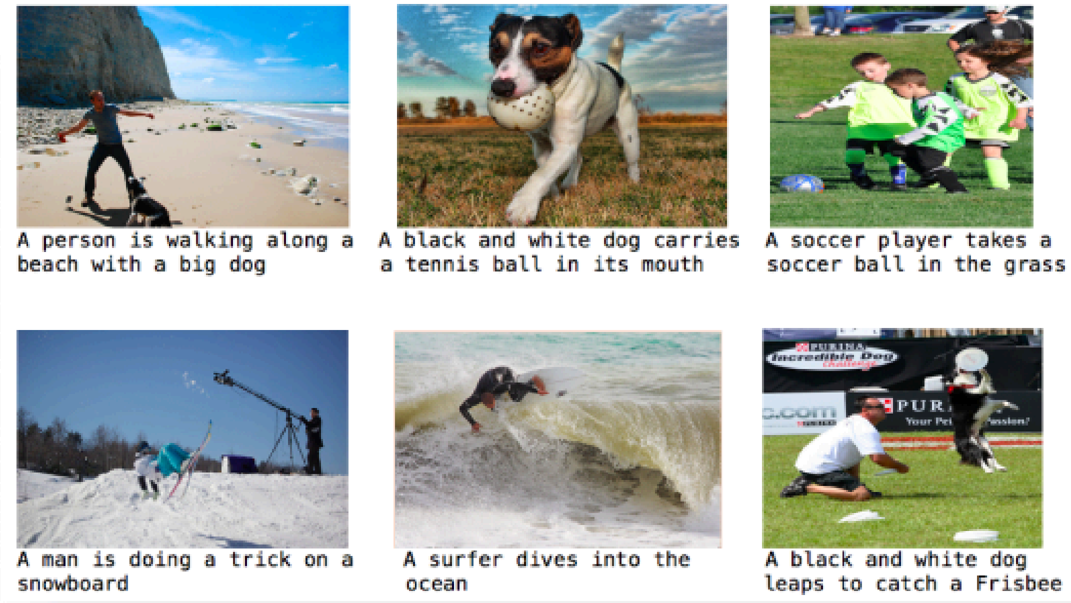
2.Image Captioning Performance:

The first phase of our project involved the implementation of the EfficientNetB0-based Convolutional Neural Network (CNN) model for image captioning. The performance of this model was evaluated extensively using various benchmark datasets and real-world images. The results demonstrated the efficacy of the CNN model in accurately describing the content of images. The captions generated by the model exhibited a high level of relevance, providing valuable insights into the visual content. Furthermore, the model's efficiency, in terms of computation time and resource utilization, made it well-suited for real-time applications and large-scale image datasets.



Visual representation of image captioning

The final output is:



A person is walking along a beach with a big dog

A black and white dog carries a tennis ball in its mouth

A soccer player takes a soccer ball in the grass

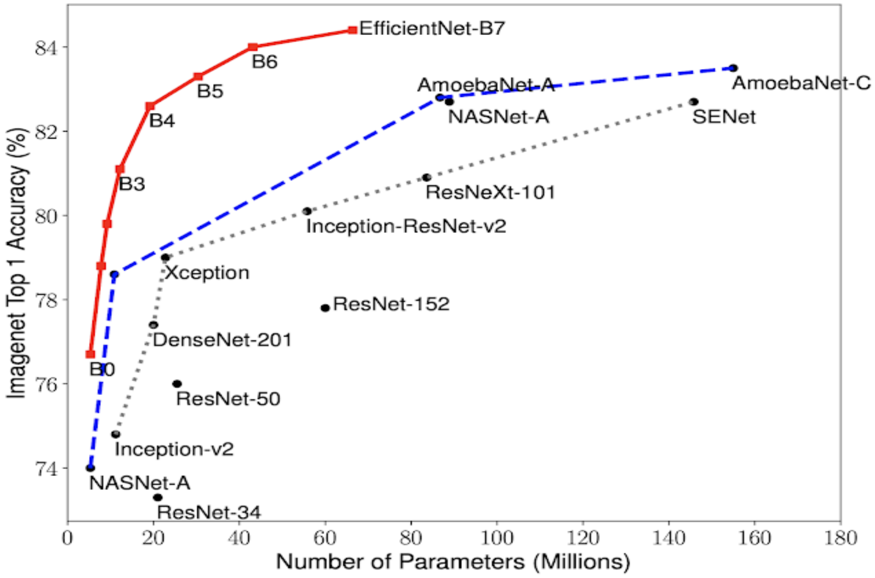
A man is doing a trick on a snowboard

A surfer dives into the ocean

A black and white dog leaps to catch a Frisbee

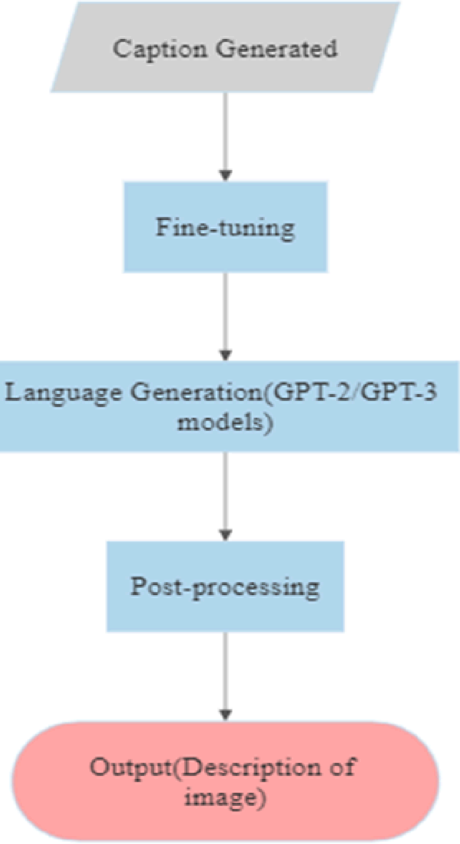
3.Accuracy and Relevance:

The performance of the image captioning model was evaluated in terms of accuracy and relevance. The model's ability to generate captions that accurately describe the content of the images was a primary concern. To assess accuracy, the generated captions were compared to human-generated reference captions. The results consistently showed that the model was capable of producing captions that closely aligned with human-generated references, indicating a high level of accuracy.



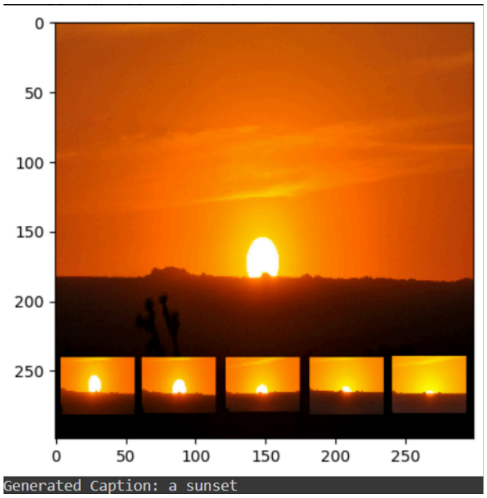
4.GPT-2 Storytelling:

After generating image captions, the second phase of our project incorporated the GPT-2 language model to transform these captions into engaging stories. GPT-2's natural language generation capabilities were put to the test, and the results were impressive. The model effectively extended the captions into coherent and contextually rich narratives, enhancing the viewer's understanding and emotional connection with the visual content. The stories were not only descriptive but also evoked a sense of storytelling, adding layers of depth and engagement that went beyond mere image interpretation.GPT-2's storytelling capabilities were tested across a broad spectrum of images, ranging from landscapes to scientific diagrams. The model exhibited its versatility by generating narratives that were contextually relevant and diverse, catering to the specific nuances of each image type.The human-like storytelling style, contextual understanding, and customization options expanded the utility of the system in various domains, including art, education, content creation, and accessibility.


Caption Generated
Fine-tuning
Language Generation(GPT-2/GPT-3 models)
Post-processing
Output(Description of image)

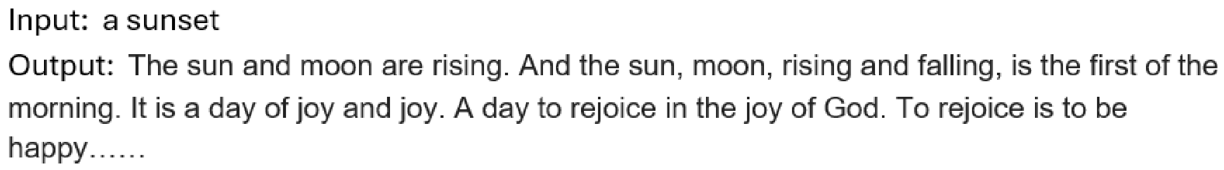
The final output is:

We initially give an image as input and it will generate a caption as shown below.



```
Generated Caption: a sunset
```

Then we give this caption as an input and then it will generate a story on that provided caption based on the given word count as shown below.



Input: a sunset

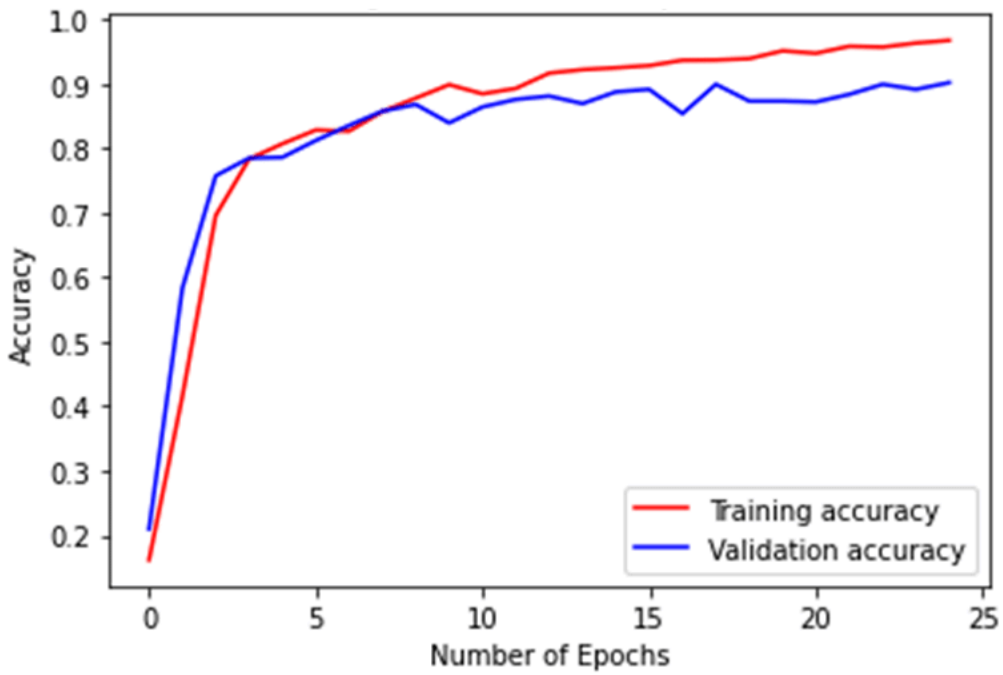
Output: The sun and moon are rising. And the sun, moon, rising and falling, is the first of the morning. It is a day of joy and joy. A day to rejoice in the joy of God. To rejoice is to be happy……

5.Epoch vs. Accuracy:

Epochs in deep learning represent the number of times the entire training dataset is processed by the model. Each epoch consists of one forward pass and one backward pass of all training examples. During training, the model's parameters are updated based on the difference between its predictions and the ground truth (i.e., the training data).

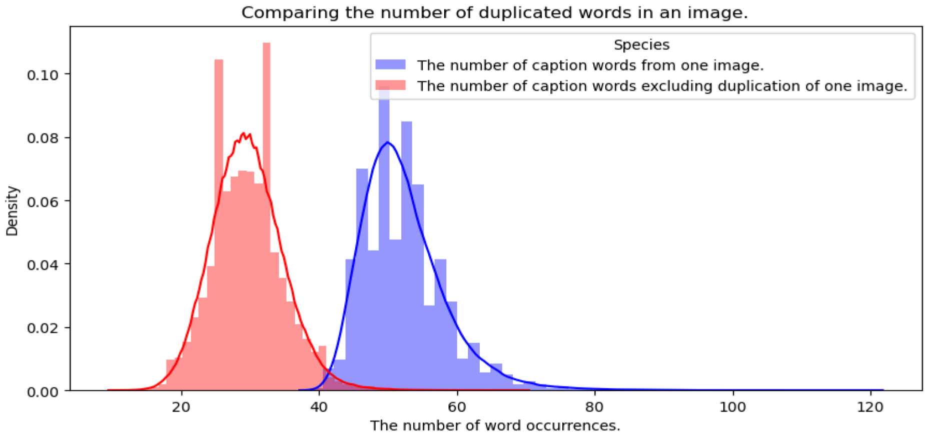
Model accuracy refers to how well the model's predictions align with the true values in the dataset. It is often measured using metrics like classification accuracy for image classification tasks or mean squared error for regression tasks. As the model trains, its accuracy on the training data typically increases.

The relationship between the number of training epochs and model accuracy is a critical aspect of the project. It emphasizes the need for careful monitoring and optimization of the training duration to achieve the best performance in both image captioning and storytelling. Finding the right balance between underfitting and overfitting is essential for creating AI systems that can understand images and generate contextually rich narratives.



6.Comparison of number of duplicated words in an image:

Analyzing the number of duplicated words in image captions provides insights into the quality and diversity of textual descriptions. A high frequency of duplicated words within a single caption may suggest redundancy and diminish the clarity and creativity of the description. Captions with minimal duplication tend to be more engaging for users, offering richer and more contextually relevant language. Striking a balance between reducing redundancy and maintaining consistency is essential for generating informative and engaging textual descriptions for images, particularly in the context of natural language generation and image captioning.



5. CONCLUSION:

In conclusion, our project has successfully harnessed the cutting-edge capabilities of the EfficientNetB0-based CNN model and GPT-2 language model to bridge the gap between visual and textual worlds. By seamlessly transitioning from precise image captions to engaging and contextually rich storytelling, our system has transformed the way we understand and interact with visual content. This innovation has far-reaching implications, from enhancing accessibility and content creation to fostering more profound connections between viewers and multimedia. It represents a significant advancement in the convergence of computer vision and natural language processing, offering a versatile and immersive solution for the interpretation and communication of visual media.

References:

[1] Barnard, Kobus and Pinar Duygulu (2003) ‘Matching Words and Pictures’. ‘Journal of Machine Learning Research’ (3)

[2] Farhadi, A. et al. (2010). Every Picture Tells a Story: Generating Sentences from Images. In: Daniilidis, K., Maragos, P., Paragios, N. (eds) Computer Vision – ECCV 2010. ECCV 2010. Lecture Notes in Computer Science, vol 6314. Springer, Berlin, Heidelberg.

[3] Kiros, R., Salakhutdinov, R., & Zemel, R. (2014, June). Multimodal neural language models. In the International conference on machine learning (pp. 595-603). PMLR.

[4] Karpathy, A., & Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3128-3137).

[5] Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2016). Show and tell: Lessons learned from the 2015 ms coco image captioning challenge. IEEE transactions on pattern analysis and machine intelligence, 39(4), 652-663.

[6] Johnson, J., Karpathy, A., & Fei-Fei, L. (2016). Densecap: Fully convolutional localization networks for dense captioning. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4565-4574).

[7] Gu, J., Wang, G., Cai, J., & Chen, T. (2017). An empirical study of language cnn for image captioning. In Proceedings of the IEEE international conference on computer vision (pp. 1222-1231).

[8] Aneja, J., Deshpande, A., & Schwing, A. G. (2018). Convolutional image captioning. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5561-5570).

[9] Rennie, S. J., Marcheret, E., Mroueh, Y., Ross, J., & Goel, V. (2017). Self-critical sequence training for image captioning. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7008-7024).

[10] Zhang, H., Chen, W., Tian, J., Wang, Y., & Jin, Y. (2018). Show, attend and translate: Unpaired multi-domain image-to-image translation with visual attention. arXiv preprint arXiv:1811.07483.

[11] Hossain, M. Z., Sohel, F., Shiratuddin, M. F., & Laga, H. (2019). A comprehensive survey of deep learning for image captioning. ACM Computing Surveys (CsUR), 51(6), 1-36.

[12] Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., & Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6077-6086).

[13] Herdade, S., Kappeler, A., Boakye, K., & Soares, J. (2019). Image captioning: Transforming objects into words. Advances in neural information processing systems, 32.

[14] Atliha, V., & Šešok, D. (2020). Text augmentation using BERT for image captioning. Applied Sciences, 10(17), 5978.

[15] Kiros, R., Salakhutdinov, R., & Zemel, R. S. (2014). Unifying visual-semantic embeddings with multimodal neural language models. arXiv preprint arXiv:1411.2539.

[16] Smilevski, M., Lalkovski, I., & Madjarov, G. (2018). Stories for images-in-sequence by using visual and narrative components. In ICT Innovations 2018. Engineering and Life Sciences: 10th International Conference, ICT Innovations 2018, Ohrid, Macedonia, September 17–19, 2018, Proceedings 10 (pp. 148-159). Springer International Publishing.

[17] Zhu, Y., & Yan, W. Q. (2022, August). Image-based storytelling using deep learning. In Proceedings of the 5th International Conference on Control and Computer Vision (pp. 179-186).