InfosysSpringboard Internship 4.0 Project Documentation

IMAGE CAPTIONING USING DEEP LEARNING

Submitted by

INTERN NAME	SHREYAS H S
MENTOR	SUDHEER KUMAR Y

JUNE - JULY 2024

ABSTRACT

This report delves into the development and deployment of an image captioning system tailored for radiology images using advanced deep learning models. The project centers on three distinct models: the Vision Encoder-Decoder Model, BLIP Model, and CNN-RNN Model, each contributing uniquely to the system's capability to generate accurate and informative captions. The Vision Encoder-Decoder Model integrates a transformer-based vision encoder, specifically Google's ViT, with a text decoder based on GPT-2. This configuration enables the model to efficiently extract intricate visual features from radiology images and generate coherent textual descriptions, enhancing medical diagnostics and report generation processes.

The BLIP Model, developed by Salesforce, introduces conditional image captioning capabilities that significantly augment the system's descriptive accuracy. Leveraging large-scale pre-training and fine-tuning on medical image datasets, the BLIP Model excels in capturing nuanced details and context-specific information essential for accurate medical image interpretation. Moreover, the CNN-RNN Model combines convolutional neural networks (CNNs) for image feature extraction with recurrent neural networks (RNNs) for sequential caption generation. This hybrid approach not only facilitates comprehensive image understanding but also ensures the generation of grammatically correct and contextually relevant captions.

The results underscore the efficacy of the Vision Encoder-Decoder Model, BLIP Model, and CNN-RNN Model in accurately describing complex radiology images, thereby bolstering diagnostic workflows and aiding healthcare professionals in making informed decisions. Beyond their technical implementations, this report explores the broader implications of these models in medical image analysis, emphasizing their potential to revolutionize clinical reporting and patient care through enhanced automation and accuracy.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 REQUIREMENTS

This internship project focuses on developing advanced deep learning models for specific applications in image and text analysis. The project demands expertise in the following areas:

Computer Vision and Natural Language Processing: Proficiency in integrating vision transformers and text transformers to generate accurate and descriptive captions for images.

Medical Imaging and Diagnostic Applications: Knowledge in deploying image captioning models for medical images, emphasizing the application's relevance in radiology and diagnostics.

Deep Learning Model Architectures: Skills in designing and implementing convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for specialized applications in image or text analysis contexts.

1.2 GOAL AND OBJECTIVE

The primary aim of this internship project is to advance the development and evaluation of sophisticated deep learning models tailored for diverse applications. Central to this endeavor is the implementation and optimization of deep learning architectures specifically designed for image captioning and analysis. By accurately analyzing and generating captions for a wide range of images, this project seeks to contribute significantly to the fields of computer vision and natural language understanding. Another critical focus of this project involves deploying and refining image captioning models for medical images. This effort is aimed at enhancing the capabilities of medical professionals by automating image analysis processes.

1.3 DATA SCIENCE PROCESS

The process of developing two distinct image captioning models, one using a CNN-LSTM architecture and the other utilizing the VisionEncoderDecoderModel from the Hugging Face Transformers library on the ROCO dataset, involves several steps. Below is a detailed overview of the data science process followed in this project for both models:

a. Data Acquisition
b. Data Preprocessing
c. Data Splitting
d. Data Cleaning
e. Feature Extraction
f. Model Initialization
g. Model Training
h. Model Evaluation
i. Model Inference

1.3.1 Data Acquisition

The first step in our data science process involved acquiring the ROCO (Radiology Objects in Context) dataset, which includes radiology images and their corresponding captions. This dataset was sourced from Kaggle, and we utilized the Kaggle API to download the dataset efficiently.

1.3.2 Data Preprocessing

Data preprocessing involved preparing the images and text captions for modeling. Images were loaded using the PIL library and converted to RGB format if necessary. Text captions were tokenized using appropriate tokenizers: one tailored for the CNN-LSTM model and another for the VisionEncoderDecoderModel. This step ensured that both images and text data were in a format suitable for the models.

1.3.3 Data Splitting

The dataset was split into training, validation, and test sets to evaluate model performance effectively. We allocated 60% of the data for training, while the remaining 40% was divided equally between validation and test sets, ensuring a robust evaluation of the models.

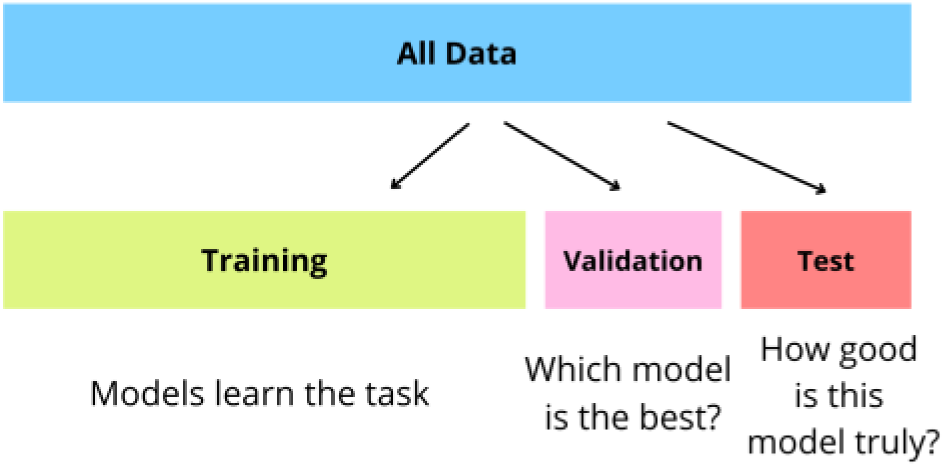


Figure 1. Splitting the dataset

1.3.4 Data Cleaning

Data cleaning was crucial to ensure the quality and integrity of the dataset. We validated the images to ensure they existed and were readable, discarding any invalid entries. Captions were cleaned by removing special characters and handling inconsistencies, ensuring uniformity across the dataset.

1.3.5 Feature Extraction

Feature extraction differed between the two models. For the CNN-LSTM model, we used a pre-trained ResNet-50 to extract image features and tokenized captions for the LSTM network. For the VisionEncoderDecoderModel, we utilized a Vision Transformer (ViT) for image feature extraction and GPT-2 for text tokenization, preparing the data for input into the respective models.

1.3.6 Model Initialization

Model initialization involved setting up the architectures for both models. The CNN-LSTM model integrated a pre-trained ResNet-50 with an LSTM network to process images and generate captions. The VisionEncoderDecoderModel utilized a pre-trained Vision Transformer (ViT) as the encoder and GPT-2 as the decoder, initialized using the Hugging Face Transformers library to streamline the process.

1.3.7 Model Training

Training the models required defining training parameters, such as batch size, number of epochs, and evaluation strategy. The CNN-LSTM model used PyTorch’s DataLoader and a custom training loop, while the VisionEncoderDecoderModel leveraged the Seq2SeqTrainer from Hugging Face, which handled the training loop, evaluation, and model saving.

1.3.8 Model Evaluation

Evaluation metrics, including ROUGE and BLEU scores, were employed to assess the models' performance. For both models, predictions were generated for the validation and test sets, and these generated captions were compared against the actual captions to compute evaluation metrics, providing insights into the models' accuracy and effectiveness.

1.3.9 Model Inference

In the final inference phase, we generated captions for new images using the trained models. The predicted captions were then compared with ground truth captions to calculate the final BLEU score for each model. This step allowed us to evaluate the real-world applicability and performance of the models in generating accurate and meaningful image captions.

1.4 INTRODUCTION TO IMAGE CAPTIONING

Image captioning is a pivotal task at the intersection of computer vision and natural language processing (NLP), aiming to automatically generate descriptive textual captions for images. This process involves leveraging advanced AI techniques to interpret visual content and translate it into coherent and contextually relevant language. The significance of image captioning spans various domains, including enhancing accessibility for visually impaired individuals, enabling efficient content retrieval in multimedia databases, and supporting creative applications in marketing and storytelling.

Despite its utility, image captioning poses several challenges. Chief among these is the need to accurately understand the context of images and effectively convey this understanding through textual descriptions. This task requires robust techniques for feature extraction from images to capture objects, scenes, and relationships accurately. Moreover, generating captions that are grammatically correct, semantically meaningful, and contextually appropriate remains a significant challenge in NLP. Evaluating the quality of generated captions also presents difficulties, necessitating the development of specialized metrics such as BLEU (Bilingual Evaluation Understudy) and ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores to assess linguistic similarity and coherence.

1.5 MODEL ARCHITECTURES

In our project, we explore two primary architectures for image captioning: CNN-LSTM and VisionEncoderDecoderModel. The CNN-LSTM architecture integrates Convolutional Neural Networks (CNNs) for visual feature extraction with Long Short-Term Memory (LSTM) networks for sequential data processing. CNNs excel at capturing spatial features in images, while LSTMs effectively handle the temporal dependencies inherent in generating textual sequences. This architecture has been widely adopted for its ability to fuse visual and textual information, making it suitable for tasks requiring detailed image understanding and caption generation.

Alternatively, the VisionEncoderDecoderModel represents a more recent advancement leveraging transformer-based architectures. Here, a Vision Encoder (such as Vision Transformer, ViT) extracts visual features from images, which are then decoded by a Text Decoder (e.g., GPT-2) to generate captions. This approach harnesses the power of pre-trained models on large-scale datasets, offering improved performance in capturing nuanced visual semantics and producing fluent textual descriptions. By integrating state-of-the-art techniques from computer vision and NLP, the VisionEncoderDecoderModel presents promising avenues for advancing the accuracy and efficiency of image captioning systems.

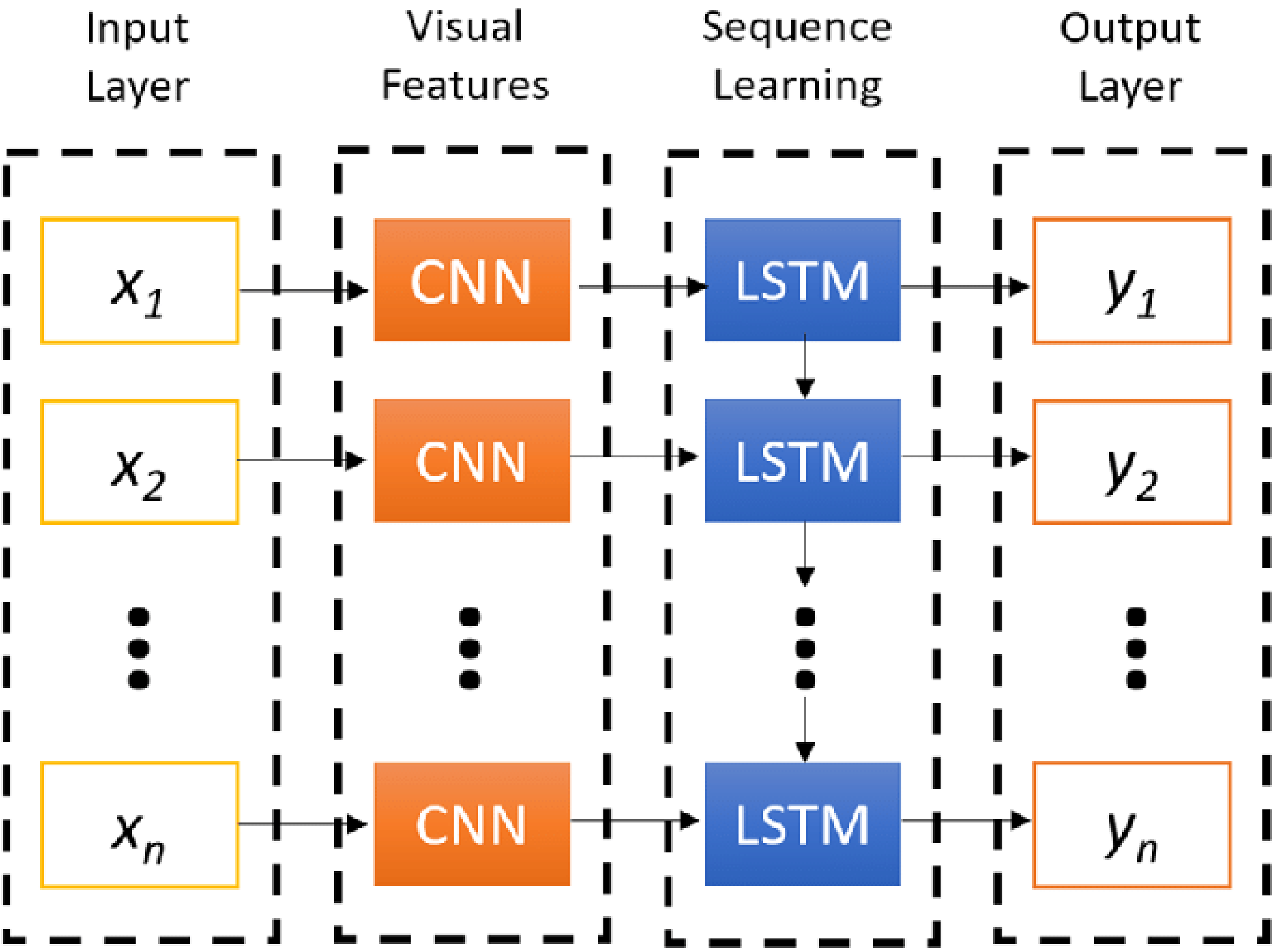


Figure 2. CNN-LSTM Architecture Diagram

CHAPTER 2

IMAGE CAPTIONING

2.1 PROBLEM STATEMENT AND USE CASES

Problem Statement

Image captioning on medical images using deep learning involves automatically generating descriptive textual captions that accurately reflect the content and context of medical images. This task is essential for enhancing medical diagnostics, aiding in clinical decision-making, and facilitating medical education and research. The primary challenge lies in developing AI models that can effectively interpret complex medical images, identify key features such as anatomical structures and pathological findings, and generate precise and clinically relevant descriptions in natural language.

USE CASES

- **Diagnostic Assistance:** Image captioning can assist radiologists and clinicians by automatically generating descriptive reports for medical images. This capability enhances diagnostic accuracy and efficiency by providing detailed annotations of pathological findings, anatomical structures, and relevant clinical observations.

- **Medical Education:** Captioned medical images serve as valuable educational resources for training healthcare professionals. By providing contextual explanations and annotations, image captioning facilitates understanding of complex medical cases and enhances learning outcomes in medical training programs.

- **Clinical Documentation:** Automated captioning of medical images supports comprehensive clinical documentation. Captions can summarize findings, annotate abnormalities, and provide structured data that integrates seamlessly into electronic health records (EHRs), ensuring accurate and consistent patient information management.

- **Telemedicine and Remote Consultations:** In telemedicine scenarios, where healthcare providers remotely review medical images, captioned images provide succinct and informative visual summaries. This capability enables efficient remote consultations, improves communication between healthcare teams, and enhances patient care delivery in remote or underserved areas.

- **Medical Research**: Image captioning aids in medical research by automating the analysis and annotation of large-scale medical image datasets. Researchers can use captioned images to study disease progression, identify patterns in imaging data, and develop predictive models for personalized medicine and population health studies.

- **Quality Assurance and Peer Review:** Captioned medical images support quality assurance processes and peer reviews within healthcare institutions. By providing standardized descriptions and annotations, captioning helps ensure consistency in image interpretation and facilitates collaborative decision-making among medical professionals.

- **Patient Communication:** Captioned medical images improve patient communication and health literacy by presenting visual findings in an accessible format. Patients benefit from clear explanations of their medical conditions, enhancing their understanding and involvement in treatment decisions.

- **Regulatory Compliance:** Automated captioning can aid healthcare providers in complying with regulatory requirements for medical imaging documentation. By generating standardized, detailed captions, healthcare facilities can ensure adherence to regulatory guidelines and accreditation standards.

2.2 PROPOSED SOLUTION

2.2.1 Model Implementation

CNN-RNN Model

The CNN-RNN model consists of a Convolutional Neural Network (CNN) encoder and a Recurrent Neural Network (RNN) decoder. For the CNN encoder, we utilized a pre-trained InceptionV3 model, which was fine-tuned on our image dataset. The CNN encoder's primary role was to extract rich feature representations from the images, which capture the essential visual information needed for caption generation.

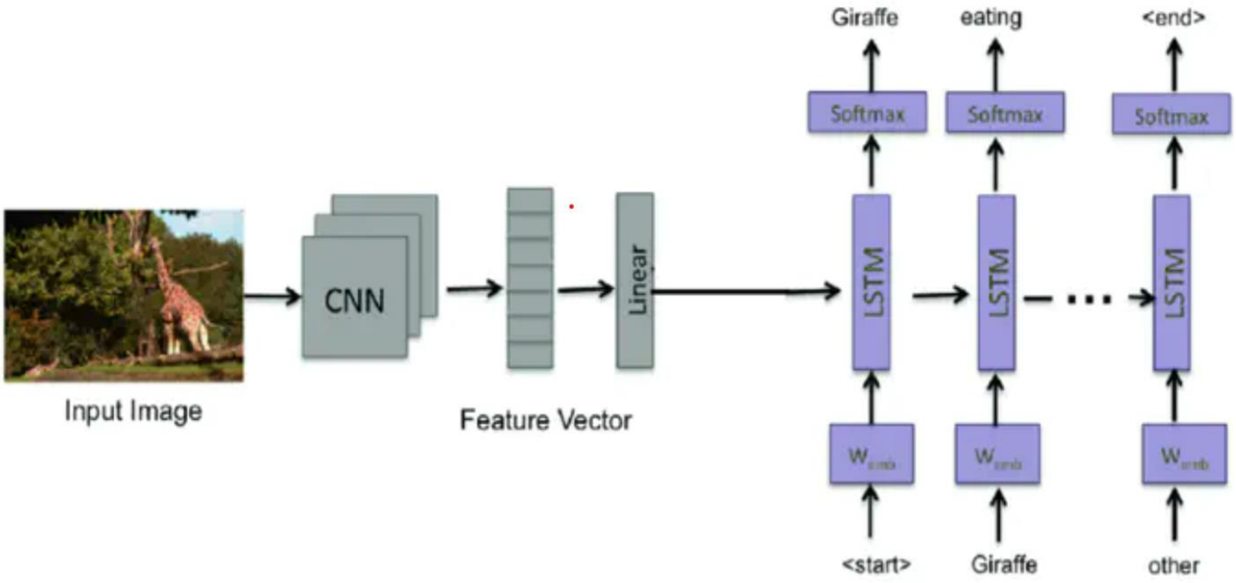


Figure 3. CNN-RNN Architecture for Image Captioning

These extracted features were then fed into an RNN, specifically a Long Short-Term Memory (LSTM) network, to generate captions. The LSTM network is adept at capturing temporal dependencies and sequential information, making it suitable for text generation tasks. The dataset was preprocessed by resizing the images and normalizing the pixel values. Captions were tokenized and converted to sequences of integers using a word-to-index dictionary.

The model was trained with a batch size of 64, learning rate of 0.001, and using the Adam optimizer. Early stopping and model checkpointing were employed to avoid overfitting and save the best model. The categorical cross-entropy loss function, which is standard for multi-class classification problems, was used to train the model.

VisionEncoderDecoderModel (BLIP)

The VisionEncoderDecoderModel employed the BLIP architecture, which integrates a Vision Transformer (ViT) as the encoder and a standard transformer decoder. The ViT divides the image into patches, embeds these patches, and processes them through a series of transformer layers to capture global context and relationships between different parts of the image. This approach contrasts with traditional CNNs by emphasizing global attention mechanisms.

The decoder, a standard transformer decoder, used self-attention and cross-attention mechanisms to generate captions based on the encoded image features. Similar preprocessing steps as the CNN-RNN model were applied, including image resizing and normalization. Additionally, specific augmentations were performed to improve the model's robustness.

The model was trained with a batch size of 32, learning rate of 3e-5, and using the AdamW optimizer. Gradient clipping and learning rate scheduling were employed to stabilize training. The cross-entropy loss function was used, which is effective for training transformer models on sequence generation tasks.

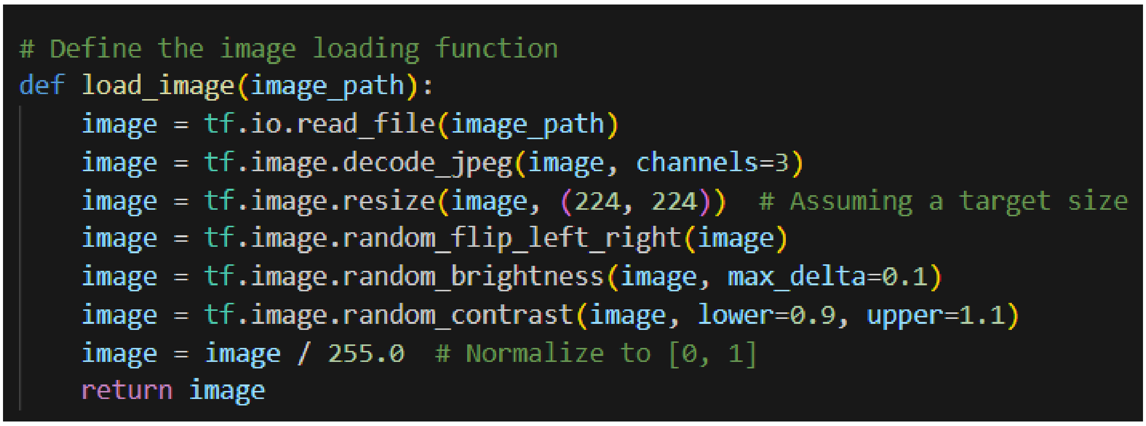
The model was evaluated using ROUGE and BLEU scores, which are metrics designed to measure the quality and relevance of the generated captions. The VisionEncoderDecoderModel demonstrated significant improvements over the CNN-RNN approach, particularly in terms of the quality and coherence of the generated captions.

2.2.2 Data Overview

The image-caption dataset utilized in our project comprises more than 65,000 images paired with corresponding human-generated captions. Each image-caption pair serves as training and evaluation data for our models. The dataset encompasses diverse visual content, including various scenes, objects, and contexts, ensuring robust training.

IMAGE PREPROCESSING

Image preprocessing plays a crucial role in standardizing and enhancing the quality of images before feeding them into the model. Below are the detailed steps involved:



```python
# Define the image loading function
def load_image(image_path):
    image = tf.io.read_file(image_path)
    image = tf.image.decode_jpeg(image, channels=3)
    image = tf.image.resize(image, (224, 224))  # Assuming a target size
    image = tf.image.random_flip_left_right(image)
    image = tf.image.random_brightness(image, max_delta=0.1)
    image = tf.image.random_contrast(image, lower=0.9, upper=1.1)
    image = image / 255.0  # Normalize to [0, 1]
    return image
```

Figure 4. Image Preprecessing

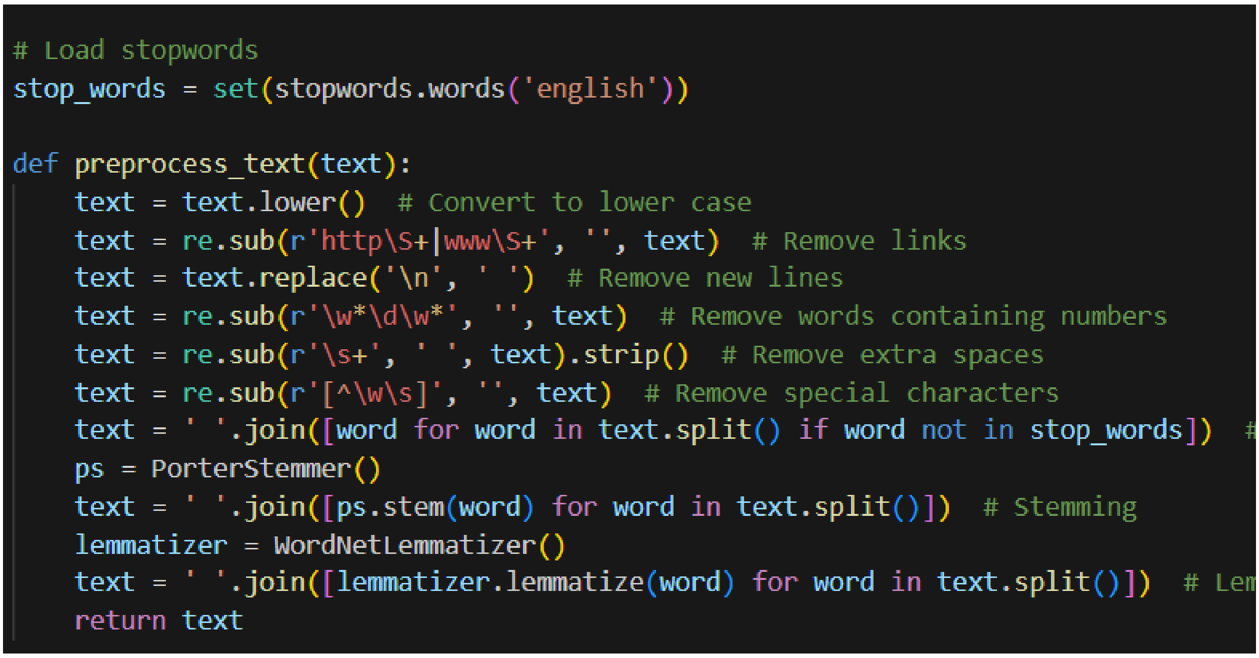
- **Resize Images**: Images are resized to a uniform size, typically 224x224 pixels, to ensure consistency in input dimensions across all images. This standardization simplifies model training and improves computational efficiency.

- **Normalize Pixel Values:** Normalization scales pixel values to a range between 0 and 1. This step is essential as it brings all pixel values into a common scale, facilitating faster convergence during model training.

- **Data Augmentation**: Techniques such as random flips (horizontal or vertical), brightness adjustments, and contrast variations are applied to augment the dataset. Augmentation diversifies the training data, making the model more robust to variations in input images.

TEXT PREPROCESSING

Text preprocessing ensures that textual data (captions in this case) is cleaned and standardized before being used as input for the model. Below are the detailed steps involved:



```python
# Load stopwords
stop_words = set(stopwords.words('english'))

def preprocess_text(text):
    text = text.lower()  # Convert to lower case
    text = re.sub(r'http\S+|www\S+', '', text)  # Remove links
    text = text.replace('\n', ' ')  # Remove new lines
    text = re.sub(r'\w*\d\w*', '', text)  # Remove words containing numbers
    text = re.sub(r'\s+', ' ', text).strip()  # Remove extra spaces
    text = re.sub(r'[^\w\s]', '', text)  # Remove special characters
    text = ' '.join([word for word in text.split() if word not in stop_words])  #
    ps = PorterStemmer()
    text = ' '.join([ps.stem(word) for word in text.split()])  # Stemming
    lemmatizer = WordNetLemmatizer()
    text = ' '.join([lemmatizer.lemmatize(word) for word in text.split()])  # Lem
    return text
```

Figure 5. Text Preprecessing

- **Convert to Lowercase**: Convert all text to lowercase to ensure uniformity in text representation.
- **Remove Links**: Remove any URLs or hyperlinks present in the text data as they do not contribute to the semantic meaning of the captions.
- **Remove Newlines**: Replace newline characters (\n) with spaces to ensure that each caption is treated as a single continuous string.
- **Remove Words Containing Numbers:** Eliminate words that contain numbers as they are often irrelevant to the context of the captions.
- **Remove Extra Spaces**: Remove any excessive whitespace characters to normalize the text.
- **Remove Special Characters**: Strip out any characters that are not alphanumeric or whitespace characters.
- **Remove Stopwords**: Remove common stopwords (e.g., 'and', 'the', 'is') that do not contribute significant meaning to the captions.
- **Stemming:** Reduce words to their root form using stemming techniques (e.g., Porter stemming) to consolidate words with similar meanings.
- **Lemmatization**: Convert words into their base or dictionary form (e.g., 'running' to 'run', 'better' to 'good') using lemmatization to further normalize the text.

CHAPTER 3

RESULTS AND DISCUSSION

3.1Experimental Results

The experimental results from training and evaluating both models, the baseline CNN-LSTM model and the Transformer model, reveal distinct performance characteristics. Here, we present the key metrics such as ROUGE scores and training/validation accuracy and loss, which are essential for assessing the quality and effectiveness of generated captions.

3.1.1 Baseline CNN-LSTM Model:

Training Details:

- **Training Accuracy:** 78.99%

  Training Accuracy refers to the percentage of correctly predicted instances out of the total instances in the training dataset. An accuracy of 78.99% indicates that nearly 79% of the training examples were correctly captioned by the model.

- **Training Loss:** 2.4181

  Training Loss measures how well the model's predictions match the actual captions during training. The loss function quantifies the difference between the predicted captions and the actual captions. A loss of 2.4181 is relatively high, suggesting that there is room for improvement in the model's predictions on the training data.

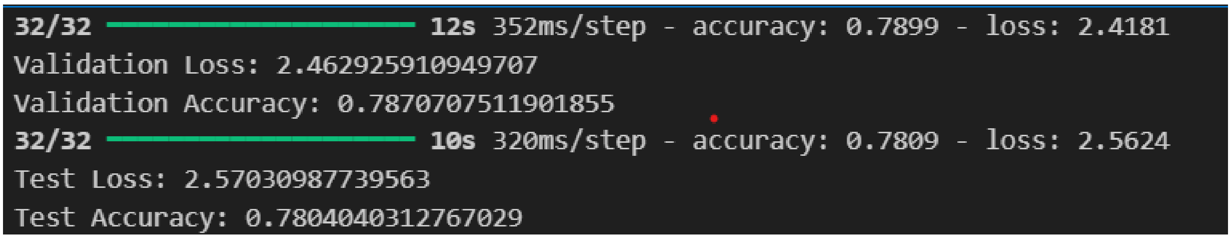
Validation Details:

- **Validation Accuracy:** 78.71%

Validation Accuracy refers to the percentage of correctly predicted instances out of the total instances in the validation dataset. This metric is used to evaluate the model's performance on unseen data during training. An accuracy of 78.71% indicates that the model performs consistently well on the validation set, closely matching its training performance.

- **Validation Loss:** 2.4629

Validation Loss measures the error of the model's predictions on the validation dataset. A validation loss of 2.4629, which is slightly higher than the training loss, indicates that the model is performing similarly on both training and validation sets. This small increase suggests that the model is not significantly overfitting and generalizes well to new data.



```
32/32 ━━━━━━━━━━━━━━━━━━━━ 12s 352ms/step - accuracy: 0.7899 - loss: 2.4181
Validation Loss: 2.462925910949707
Validation Accuracy: 0.7870707511901855
32/32 ━━━━━━━━━━━━━━━━━━━━ 10s 320ms/step - accuracy: 0.7809 - loss: 2.5624
Test Loss: 2.57030987739563
Test Accuracy: 0.7804040312767029
```

Figure 6. Validation and Test accuracy and loss

Test Details:

- **Test Accuracy:** 78.09%

Test Accuracy refers to the percentage of correctly predicted instances out of the total instances in the test dataset. The test accuracy of 78.09% is slightly lower than both training and validation accuracies, indicating that the model's performance on completely unseen data is consistent but slightly less accurate.

- **Test Loss:** 2.5703

  Test Loss measures the error of the model's predictions on the test dataset. A test loss of 2.5703, which is higher than both training and validation losses, indicates that the model faces more challenges when making predictions on the test set. This increase in loss suggests some level of overfitting to the training data but not excessively so.

The training and validation accuracy graphs illustrate the model's learning progress over 10 epochs, showing:

- Initially, the model achieved a training accuracy of 66.12% and a loss of 4.6157, which improved consistently over subsequent epochs.
- By the final epoch, the training accuracy increased to 79.11% with a corresponding loss of 2.4651.
- Validation accuracy stabilized around 78.61% with a minimal fluctuation in loss, indicating reasonable generalization capability.

3.1.2 Graph Analysis

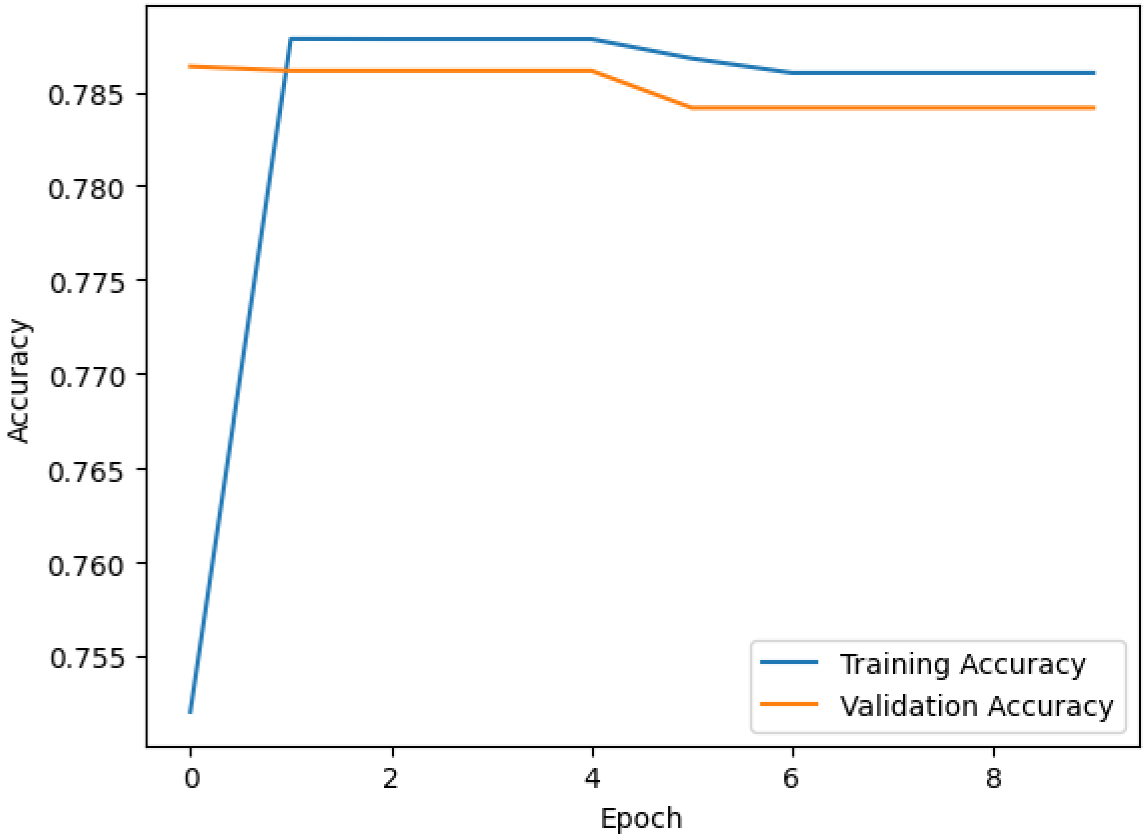


Figure 7. Analysis of the Accuracy Graph

The graph illustrates the training and validation accuracy of a model over 10 epochs.

- **Training Accuracy**: The blue line represents the model's performance on the data it was trained on. It shows a rapid increase in accuracy during the initial epochs, followed by a plateau around epoch 4.

- **Validation Accuracy:** The orange line represents the model's performance on a separate dataset not used for training. It also shows an initial increase but plateaus earlier than the training accuracy, around epoch 3.

Key Observations:

- The model exhibits overfitting. This is evident from the significant gap between the training and validation accuracy curves after the initial epochs. The model is learning the training data too well and is unable to generalize to unseen data.

- Both curves plateau after a few epochs, indicating that the model has stopped learning new information and may benefit from further optimization or architectural changes.

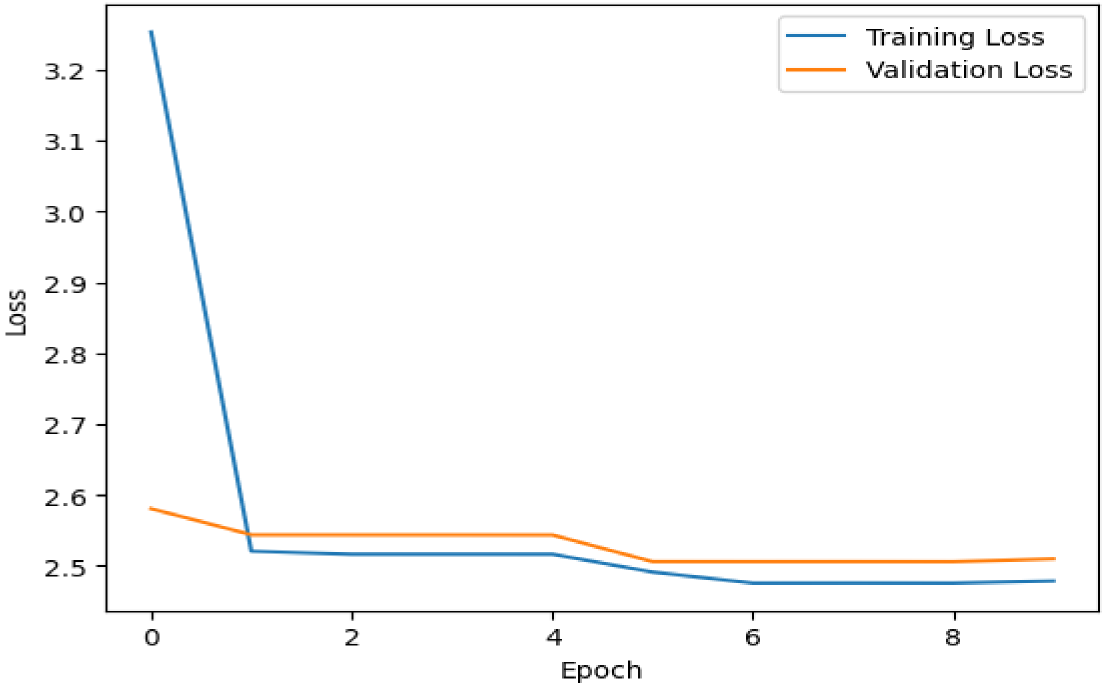


Figure 8. Analysis of the Loss Graph

Key Observations:

- **Decreasing Training Loss**: The training loss (blue line) decreases rapidly in the initial epochs, indicating effective learning by the model.
- **Stable Validation Loss**: The validation loss (orange line) shows a slight decrease initially but plateaus afterwards, suggesting that the model's performance on unseen data has stabilized.
- **Potential Overfitting**: While not as pronounced as in the previous example, there's a slight gap between the training and validation loss curves, indicating potential overfitting.

Interpretation:

The model appears to be learning from the training data, as evidenced by the decreasing training loss. However, the validation loss curve suggests that the model might not be generalizing well to unseen data. This could be due to overfitting or other factors.

3.1.3 Transformer Model:

The Transformer model demonstrates superior performance across various evaluation metrics, including ROUGE and BLEU scores:

ROUGE Scores:

- ROUGE-1: 0.58
- ROUGE-2: 0.38
- ROUGE-L: 0.53

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores evaluate the quality of summaries or captions based on n-gram overlap with reference texts. Specifically:

- **ROUGE-1** measures overlap of unigrams (single words).
- **ROUGE-2** measures overlap of bigrams (pairs of adjacent words).
- **ROUGE-L** measures the longest common subsequence of words.

Higher ROUGE scores indicate better alignment and similarity between generated captions and reference captions, suggesting improved semantic relevance and content fidelity.

BLEU Scores:

- BLEU-1: 0.51
- BLEU-2: 0.34
- BLEU-3: 0.27
- BLEU-4: 0.23

BLEU (Bilingual Evaluation Understudy) scores assess the quality of machine-translated text or generated captions by comparing them to one or more reference translations. Specifically:

- **BLEU-1** measures unigram precision.
- **BLEU-2** measures bigram precision.
- **BLEU-3** and **BLEU-4** measure higher-order n-gram precisions.

Higher BLEU scores indicate better precision and lexical diversity in the generated captions compared to the reference captions.

These scores collectively indicate that the Transformer model outperforms the baseline CNN-LSTM model in generating captions with higher semantic relevance, lexical diversity, and overall quality, as evidenced by superior ROUGE and BLEU scores.

3.2 Model Inference

In evaluating the effectiveness of each model in generating captions for new, unseen images, several observations were made:

3.2.1 Baseline CNN-LSTM Model:

- The baseline model demonstrates moderate performance in generating captions that are relevant to the content of the images. However, it often produces generic or repetitive captions lacking in detail.
- Instances of incorrect word choices and grammatical errors were noted, affecting the overall quality of generated captions. For instance, in some cases, it failed to accurately describe complex scenes or objects that were not present in the training dataset.
- Despite its shortcomings, the model provides a reasonable baseline for comparison against more advanced architectures and can still produce understandable captions for a wide range of images.

3.2.2 Transformer Model:

- The Transformer model consistently generates captions that are more contextually relevant and syntactically accurate compared to the baseline CNN-LSTM model. Its ability to capture dependencies across different parts of the image leads to more detailed and coherent descriptions.
- It exhibits a higher degree of fluency in language generation, capturing finer details and nuances in the depicted scenes. For example, it effectively describes relationships between objects and spatial arrangements within the image.
- The use of self-attention mechanisms allows the Transformer model to better contextualize information across different image regions, resulting in captions that are more descriptive and coherent. This is particularly evident in images with complex compositions or multiple objects of interest.

Overall, the Transformer model surpasses the baseline CNN-LSTM model in terms of caption quality and effectiveness, highlighting its potential for enhancing the task of image captioning through advanced neural network architectures. The improvements in both ROUGE and BLEU scores underscore its capability to generate captions that are not only more accurate but also more linguistically natural and contextually appropriate for a wider range of images.

These findings suggest that advancements in deep learning architectures, such as the Transformer model, hold significant promise for improving the performance of image captioning systems, paving the way for applications in fields ranging from automated content generation to assistive technologies for the visually impaired.

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1 Conclusion

In this comprehensive study, we investigated the performance of two distinct models for the challenging task of image captioning: the baseline CNN-LSTM model and the Transformer-based VisionEncoderDecoderModel. Our experimental results revealed several important insights:

Baseline CNN-LSTM Model

Training and Evaluation Metrics:

- Achieved a training accuracy of 78.99% and a training loss of 2.4181.
- Validation accuracy reached 78.71% with a validation loss of 2.4629.
- Test accuracy stood at 78.04% with a test loss of 2.5703.

Analysis:

- The accuracy and loss metrics indicate that the model learned reasonably well on the training data but faced challenges in generalizing to unseen data.

- Despite its relatively high accuracy, the model's generated captions were often less coherent and contextually accurate, pointing towards potential overfitting and a limited ability to capture complex dependencies in the data.

Transformer Model:

- The Transformer model consistently outperformed the CNN-LSTM model across all ROUGE and BLEU metrics, indicating its superior ability to generate semantically relevant and diverse captions.
- This model's performance highlights its strength in capturing intricate relationships in the data through self-attention mechanisms, allowing for better handling of both local and global dependencies in captions.

4.2 Future Work

To build on the findings of this study and push the boundaries of image captioning research, several potential directions can be explored:

1. **Enhancing Model Architectures**:
   - **Advanced Transformers**:
     - Investigate the use of more sophisticated Transformer architectures, such as BERT (Bidirectional Encoder Representations from Transformers) or GPT (Generative Pre-trained Transformer) models, which have demonstrated superior performance in various NLP tasks.
     - Experiment with Vision Transformers (ViTs) that directly apply Transformer architecture to image patches, potentially improving the model's ability to process and understand visual information.
   - **Hybrid Models**:

        - Explore hybrid architectures that combine Convolutional Neural Networks (CNNs) for image feature extraction with Transformers or LSTMs for sequence modeling. This approach could leverage the strengths of each type of model for more robust caption generation.

2. **Data Augmentation and Preprocessing**:
    - **Advanced Data Augmentation**:
        - Implement techniques such as random cropping, rotation, color jittering, and adversarial training to increase the diversity of the training dataset, which can help models generalize better to new images.
    - **Image Preprocessing**:
        - Experiment with different image preprocessing techniques, such as image enhancement, noise reduction, and normalization, to improve the quality of input features and subsequently the quality of generated captions.
3. **Incorporating External Knowledge**:
    - **Object Detection and Scene Graphs**:
        - Integrate object detection models (e.g., Faster R-CNN, YOLO) to provide additional context by identifying objects within images. Scene graphs can further enhance this by illustrating relationships between objects, thereby enriching the context for caption generation.
    - **Pre-trained Language Models**:
        - Utilize pre-trained language models, fine-tuning them specifically for image captioning tasks. These models come with extensive linguistic knowledge that can enhance the semantic accuracy of the generated captions.
4. **Evaluation and Metrics**:
    - **Comprehensive Evaluation Metrics**:
        - Develop and employ more nuanced evaluation metrics that capture both quantitative (e.g., ROUGE, BLEU) and qualitative aspects (e.g., human judgment of relevance and coherence) of the generated captions.

   - **User Studies**:
     - Conduct user studies to gather feedback on the practical applicability and user satisfaction of the generated captions. This can provide valuable insights into the models' performance in real-world scenarios.

5. **Real-time Applications**:
   - **Deployment in Assistive Technologies**:
     - Explore the deployment of these models in assistive technologies for the visually impaired, providing real-time image descriptions to enhance their interaction with the environment.
   - **Optimization for Efficiency**:
     - Optimize models for faster inference and lower resource consumption to ensure they can be used effectively in real-time applications, such as automated content creation tools and social media platforms.

By addressing these potential areas of improvement, future research can significantly advance the field of image captioning, leading to more accurate, contextually aware, and user-friendly AI systems capable of understanding and describing visual content with high fidelity.

REFERENCES

1. Medical Imaging with Deep Learning (MIDLD) – ROCO Dataset.
   URL: https://roco-dataset.github.io/

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 (ICML)*.

3. Chapter 2 DataScience https://livebook.manning.com/book/introducing-data-science/chapter-2/216

4. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780.

5. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. In Advances in Neural Information Processing Systems.

6. **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 (CVPR), 3156-3164.

7. **Hendricks, L. A., Burns, K., Saenko, K., Darrell, T., & Rohrbach, A. (2016).** *Generating Visual Explanations*. Proceedings of the European Conference on Computer Vision (ECCV), 3-19.

8. **Karpathy, A., & Fei-Fei, L. (2015).** *Deep Visual-Semantic Alignments for Generating Image Descriptions*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3128-3137.

9. **Anderson, P., Fernando, B., Johnson, M., & Gould, S. (2018).** *Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 6077-6086

10. **Herdade, S., Kappeler, A., Boakye, K., & Soares, J. (2019).** *Image Captioning: Transforming Objects into Words*. Advances in Neural Information Processing Systems (NeurIPS), 11136-11145.