Assignment 3

**Image Captioning**



Anna Roy - 24886234

[Ezilaan Irraivan](https://canvas.uts.edu.au/groups/214888/users/169389) - 24695379

Hema Vajravelu - 24613503

[Rohit Sharma](https://canvas.uts.edu.au/groups/214888/users/149243) - 24590960

[**94691 - Deep Learning**](https://canvas.uts.edu.au/courses/31218)

**Master of Data Science and Innovation**

**University of Technology of Sydney**

Table of Contents

[1. Introduction 1](#_Toc167485574)

[2. Data Understanding 3](#_Toc167485575)

[3. Data Preparation 5](#_Toc167485576)

[4. Modelling 7](#_Toc167485577)

[**a.** **Resnet50 Model – Hema Vajravelu** 9](#_Toc167485578)

[**b.** **VGG16 Model – Anna Roy** 10](#_Toc167485579)

[**c.** **GoogleNet Model – Rohit Sharma** 11](#_Toc167485580)

[**d.** **MobileNetV2 Model – Ezilaan Irraivan** 12](#_Toc167485581)

[5. Evaluation 14](#_Toc167485582)

[**a.** **Evaluation Metrics** 14](#_Toc167485583)

[**b.** **Results and Analysis** 15](#_Toc167485584)

[**c.** **Data Privacy and Ethical Concerns** 16](#_Toc167485585)

[6. Conclusion 17](#_Toc167485586)

[7. References 18](#_Toc167485587)

# Introduction

In the modern digital landscape, images have become an integral mode of communication, capturing moments, conveying emotions, and narrating stories. However, while humans effortlessly decode visual content, teaching machines to comprehend and describe images presents a formidable challenge. Image captioning, an automated process of creating textual descriptions for images, stands at the intersection of computer vision and natural language processing. It presents an exciting opportunity to bridge the cognitive divide between visual content and language, enriching our engagement with visual data.

This project delves into developing and evaluating an image captioning model implemented using PyTorch, applied to the Flickr8k dataset. Comprising 8,000 images, each paired with five captions, this dataset is a benchmark for sentence-based image description tasks. We aim to craft a model capable of analysing input images and producing accurate and contextually relevant textual descriptions, mirroring the cognitive process of a human observer in perceiving and articulating image content.

We adopt a systematic approach, exploring a range of architectural designs, data preprocessing techniques, evaluation metrics, and training strategies to develop a robust image captioning solution. The model architecture comprises a feature extraction component, leveraging a pre-trained convolutional neural network (CNN) to extract salient visual features, alongside a sequence generation module employing a recurrent neural network (RNN) with long short-term memory (LSTM) units to generate captions based on these features.

Evaluation of the model entails using metrics such as BLEU (Bilingual Evaluation Understudy) scores, which gauge the resemblance between the generated captions and the ground truth references provided in the dataset. Additionally, we thoroughly analyse the model's performance, pinpointing limitations and offering recommendations for further enhancement.

Our efforts aim to advance computer vision and natural language processing by addressing the intrinsic obstacles of comprehending and describing images. We envision that robust image captioning systems will enrich human-computer interaction and find utility in diverse applications such as assistive technology, content indexing, and multimedia retrieval.

# Data Understanding

The dataset utilized in this project is the Flickr8k dataset, which is widely recognized as a benchmark dataset for sentence-based image description tasks. It comprises a diverse collection of 8,000 images, each with five descriptive captions. This dataset is a valuable aid for training and assessing image captioning models, offering diverse visual content coupled with numerous textual descriptions.

Each image in the Flickr8k dataset is associated with five captions, capturing different perspectives and interpretations of the visual content. This diversity in captions allows for robust training of image captioning models, enabling them to generate varied and contextually relevant descriptions for a given image.

The dataset includes images with a broad spectrum of scenes, objects, and activities, enabling the evaluation of the image captioning model's ability to generalize across various visual scenarios. Additionally, the dataset's large size and multiple captions per image contribute to its effectiveness as a benchmark for evaluating the quality and diversity of generated captions.



*Figure 1: Sample image from the datas**et*

**Data Limitations**

* **Limited Diversity:** The dataset contains only 8,000 images, which is relatively small compared to other datasets like MS COCO. This limits the model's ability to generalize to a wide range of images.
* **Caption Diversity:** While each image has five captions, the diversity and complexity of these captions might not be sufficient to cover all possible descriptions of the image.
* **Quality of Annotations:** Human-generated captions can vary in quality and detail, which might introduce variability in the training process.
* **Subjectivity:** The captions are subjective and may vary significantly in style and detail due to the different perspectives of annotators.
* **Bias:** Since the dataset was collected from Flickr, it may contain cultural and contextual biases based on the user base of the platform.

**Features in the Dataset**

The dataset consists of two primary features:

* **Images:** The visual content of the dataset. Each image is represented as a high-resolution RGB image, typically resized to a standard dimension for model training.
* **Captions:** Text descriptions associated with each image. Each image has five different captions, which are tokenized and converted into sequences of word indices for model training.

**Significance of Features**

* **Images:** The images provide the visual input for the image captioning models. The ability to extract meaningful features from images is crucial for generating relevant and accurate captions.
* **Captions:** The captions serve as the ground truth for training and evaluating the models. They help the model learn the mapping between visual features and textual descriptions, enabling it to generate coherent and contextually appropriate captions.

# Data Preparation

**Data Cleaning and Pre-processing**

To prepare the Flickr8k dataset for modelling, several data cleaning and pre-processing steps were necessary. These steps ensured that the data was in a suitable format for training and evaluating the image captioning models.

**Steps Taken for Data Preparation**

* **Loading the Data:** Images and captions were loaded into the environment. The images were stored in directories, and the captions were provided in a text file with each line containing an image identifier followed by its corresponding caption.
* **Data Cleaning**
* **Text Normalization:** The captions were converted to lowercase to maintain consistency.
* **Removing Punctuation:** Punctuation marks were removed from the captions to simplify the tokenization process.
* **Tokenization:** Captions were tokenized into individual words. This involved splitting each caption into a list of words.
* **Vocabulary Creation**
* **Word to Index Mapping:** A dictionary was created mapping each unique word to a unique index. This allowed the conversion of words in captions to integer indices.
* **Padding Sequences:** Captions were padded to ensure all sequences had the same length, which is required for batch processing in neural networks. Padding was added to the end of shorter sequences to match the length of the longest caption.
* **Splitting the Data**

The dataset was split into training, validation, and testing sets based on predefined image lists. This ensured that the same images were consistently used for training and evaluation across different experiments.

* **Image Pre-processing**
* **Resizing:** All images were resized to a standard size (224x224 pixels) to ensure consistency across the dataset.
* **Normalization:** Pixel values were normalized to the range [0, 1] by dividing by 255. This helped in speeding up the training process and improving convergence.
* **Feature Extraction:** Pre-trained CNN models (*ResNet, VGG16, GoogLeNet, MobileNetV2*) were used to extract feature vectors from the images. The final fully connected layer was removed, and the output of the last convolutional layer was used as the feature vector for each image.
* **Feature Engineering**
* **Image Features:** As mentioned, image features were extracted using pre-trained CNN models. These feature vectors served as the input to the caption generation model.
* **Text Features:** Captions were converted to sequences of word indices. Additionally, start and end tokens (<start> and <end>) were added to each caption to help the model learn the beginning and end of a sentence.
* **Embedding Layer:** An embedding layer was used to convert word indices to dense vectors, which captured the semantic meaning of words.

# Modelling

The primary focus of this project was to develop image captioning models using a combination of Convolutional Neural Networks (CNNs) for feature extraction and Recurrent Neural Networks (RNNs) for caption generation.

The following algorithms for the encoders and architectures were used

* ***ResNet (Residual Networks)***
* ***VGG16 (Visual Geometry Group)***
* ***GoogLeNet (Inception v1)***
* ***MobileNetV2***
* ***LSTM (Long Short-Term Memory)***

**Rationale Behind Selecting These Algorithms**

* **CNNs for Feature Extraction:**
* **ResNet:** ResNet is known for its deep architecture with residual connections that mitigate the vanishing gradient problem, making it highly effective for extracting complex features from images.
* **VGG16:** VGG16 is a simpler architecture compared to ResNet but is effective in extracting spatial features due to its deep convolutional layers.
* **GoogLeNet:** GoogLeNet introduces the Inception module, which captures multi-scale features efficiently, making it a versatile choice for image feature extraction.
* **MobileNetV2:** MobileNetV2 is designed for efficient computation with depth-wise separable convolutions, making it suitable for lightweight models while maintaining high accuracy.
* **RNNs for Caption Generation:**
* **LSTM:** LSTMs are a type of RNN designed to handle long-term dependencies in sequential data, making them suitable for generating coherent and contextually appropriate captions from image features.

**Parameter Tuning and Model Selection Process**

* **Feature Extraction Using Pre-trained CNNs**
* Pre-trained versions of ResNet, VGG16, GoogLeNet, and MobileNetV2 were used, with the final fully connected layers removed to extract feature vectors from images.
* The output of the last convolutional layer of each CNN served as the input feature vector for the caption generation model.
* **Caption Generation with LSTM**
* An embedding layer was used to convert word indices to dense vectors.
* The LSTM network was configured with one or more layers, where each layer had a specified number of units (e.g., 256 or 512).
* Dropout was applied to prevent overfitting, with dropout rates typically set between 0.3 and 0.5.
* **Parameter Tuning**
* **Batch Size:** Different batch sizes (e.g., 32, 64) were experimented with to balance between memory usage and training speed.
* **Learning Rate:** Various learning rates (e.g., 0.001, 0.0001) were tested to find the optimal rate for model convergence.
* **Optimizer:** The Adam optimizer was chosen for its adaptive learning rate properties, which help in faster convergence.
* **Sequence Length:** The maximum caption length was set based on the longest caption in the dataset, with shorter captions padded accordingly.
* **Model Selection**
* Models were evaluated using BLEU scores (BLEU-1, BLEU-2, BLEU-3, BLEU-4) on the validation set to assess their performance.
* The model with the best performance metrics on the validation set was selected for final evaluation on the test set.

**Considerations during Modelling**

* **Over-fitting:** Regularization techniques such as dropout were used to prevent overfitting.
* **Sequence Padding:** Captions were padded to ensure uniform sequence lengths, which is required for batch processing in RNNs.
* ***Computational Efficiency:*** *MobileNetV2 was considered for its lightweight architecture, making it suitable for deployment in resource-constrained environments.*
* **Model Interpretability:** Attention mechanisms could be explored in future work to improve interpretability and performance by allowing the model to focus on specific parts of the image when generating captions.

The combination of CNNs for feature extraction and LSTMs for caption generation proved effective for the image captioning task. The chosen architectures and hyperparameters were selected based on their ability to capture and process complex visual and textual information, with the ultimate goal of generating accurate and contextually appropriate captions.

Parameter tuning and model selections were guided by performance metrics and visual inspection, ensuring the development of a robust image captioning system.

## **Resnet50 Model – Hema Vajravelu**

ResNet50 is a deep convolutional neural network with 50 layers, known for its residual learning framework which helps in training deeper networks effectively by solving the vanishing gradient problem. The encoder part of the model uses ResNet50 to extract high-level image features.

**Algorithm and Hyper parameters**

* **Encoder:** ResNet50
* **Layers:** 50
* **Pre-trained:** True
* **Embedding Size:** 256
* **Linear Layer:** Fully connected layer with batch normalization
* **Decoder**: LSTM
* **Embedding Size:** 256
* **Hidden Size:** 512
* **Number of Layers:** 2
* **Dropout:** 0.5

**Performance Analysis of Resnet Model**

* **BLEU-1:** ***0.1379*** indicates the accuracy of unigrams (single words), showing a moderate ability to predict individual words correctly.
* **BLEU-2:** ***0.0341*** for bigrams suggests that the model struggles more with predicting pairs of words correctly, which affects the fluency of the generated captions.
* **BLEU-3:** ***0.0225*** and **BLEU-4**: ***0.0192*** further indicate that the model's performance diminishes with longer n-grams, highlighting issues in capturing more extended sequences and context.

ResNet50’s architecture is powerful for feature extraction due to its deep layers and residual connections. However, the relatively low BLEU scores, especially for higher n-grams, suggest that while the model can identify individual objects and attributes in images, it struggles with forming coherent and contextually accurate sentences. This could be due to the complexity of language generation which requires understanding relationships between objects and actions depicted in images.

## **VGG16 Model – Anna Roy**

VGG16 is a convolutional neural network with 16 layers known for its simplicity and effectiveness. It uses smaller convolutional filters (3x3) which capture more fine-grained details in the images.

**Algorithm and Hyper-parameters**

* **Encoder:** VGG16
* **Pre-trained:** True
* **Embedding Size:** 256
* **Linear Layer:** Fully connected layer with batch normalization
* **Decoder:** LSTM
* **Embedding Size:** 256
* **Hidden Size:** 512
* **Number of Layers:** 1

**Performance Analysis**

* **BLEU-1:** ***0.1617*** is the highest among the models, showing better performance in predicting individual words.
* **BLEU-2:** ***0.0371*, BLEU-3:** ***0.0246***, and **BLEU-4:** ***0.0208*** scores, although still low, are higher than those of ResNet50, indicating better performance in predicting longer n-grams.

The higher BLEU scores for VGG16 suggest that its architecture might be better suited for capturing detailed features and patterns in images that are relevant for language generation. This model's ability to predict words and short phrases more accurately suggests it may be capturing some contextual relationships better than ResNet50. However, the drop-off in scores for higher n-grams still indicates challenges in forming longer, coherent sentences.

## **GoogleNet Model – Rohit Sharma**

GoogLeNet, also known as Inception v1, introduces the Inception module which allows the network to capture multi-scale information by using multiple convolutional filters of different sizes simultaneously.

**Algorithm and Hyper parameters**

* **Encoder:** GoogLeNet
* **Pre-trained:** True
* **Embedding Size:** 256
* **Linear Layer:** Fully connected layer with batch normalization and dropout
* **Decoder:** LSTM
* **Embedding Size:** 256
* **Hidden Size:** 512
* **Number of Layers:** 2
* **Dropout:** 0.5

**Performance Analysis**

* **BLEU-1:** ***0.1438***, **BLEU-2:** ***0.0344***, **BLEU-3:** ***0.0228***, and **BLEU-4:** ***0.0191*** scores indicate performance comparable to ResNet50, with a slight edge in unigram prediction but similar challenges with longer n-grams.

GoogLeNet's architecture is designed to handle varying scales of features, which helps in recognizing objects and attributes at different levels of detail. However, similar to ResNet50, it seems to struggle with longer n-grams, indicating difficulties in generating coherent and contextually accurate longer sentences. This could be due to the complexity of the language modelling task which requires understanding the broader context within an image.

## **MobileNetV2 Model – Ezilaan Irraivan**

MobileNetV2 is a lightweight convolutional neural network designed for mobile and embedded vision applications. It achieves a good balance between accuracy and computational efficiency, making it suitable for resource-constrained environments. It gives a reduction in operations so it is highly applicable with machines with limited processing power.

**Algorithm and Hyper parameters**

* **Encoder:** MobileNetV2
* **Pre-trained:** True
* **Embedding Size:** 512
* **Linear Layer:** Fully connected layer with batch normalization and dropout
* **Activation Layer:** Tanh
* **Decoder:** LSTM
* **Embedding Size:** 512
* **Hidden Size:** 512
* **Number of Layers:** 1
* **Dropout:** 0.2

**Performance Analysis**

* **BLEU-1:** ***0.1469***, **BLEU-2:** ***0.0346***, **BLEU-3:** ***0.0222***, **BLEU-4: *0.0183***. These scores indicate a performance slightly better than ResNet50, particularly in unigram prediction (BLEU-1), but similar challenges with longer n-grams.

MobileNetV2's architecture is optimized for efficiency, allowing it to process images quickly and with relatively low computational resources. The model's performance, as indicated by the BLEU scores, is comparable to that of ResNet50 and GoogLeNet, with similar challenges in generating longer n-grams.

While MobileNetV2 may not perform as well as VGG16 in predicting individual words or short phrases, its efficiency and compactness make it a strong candidate for deployment in mobile or edge devices where resource constraints are a concern.

The trade-off between accuracy and efficiency is often crucial in real-world applications, and MobileNetv2's performance makes it a viable option for such scenarios.

**Overall Outcome**

* **Performance Comparison:** VGG16 performs the best in predicting individual words (BLEU-1) among the four models, closely followed by MobileNetV2. However, all models face challenges with longer n-grams, indicating difficulties in generating coherent and contextually accurate longer sentences.
* **Architecture Considerations**: ResNet50 and GoogLeNet offer more complex architectures with capabilities for capturing multi-scale information, while VGG16 is simpler but effective. MobileNet provides a lightweight alternative suitable for resource-constrained environments.
* **Project Specificity:** *The choice of model chosen on the specific requirements of the project, such as computational resources, need for detailed feature extraction,* ***and the importance of contextual understanding in generating captions seems to be better in Google Net, though VGG16 performs great in capturing individual words.***

# Evaluation

## **Evaluation Metrics**

**BLEU Scores**

BLEU (Bilingual Evaluation Understudy) scores are widely used for evaluating the quality of text generated by machine learning models, especially in tasks like machine translation and image captioning. BLEU scores measure how close the generated captions are to the reference (ground truth) captions. They compare the n-grams (contiguous sequences of n items) of the generated captions with those of the reference captions and provide a score based on the overlap.

For this project, the following BLEU scores were used:

* ***BLEU-1: Measures the precision of unigrams (single words) in the generated captions.***
* ***BLEU-2: Measures the precision of bigrams (two-word sequences).***
* ***BLEU-3: Measures the precision of trigrams (three-word sequences).***
* ***BLEU-4: Measures the precision of four-grams (four-word sequences).***

These metrics were chosen for the following reasons:

* **Relevance to Project Goals:** The primary goal of the project is to generate accurate and contextually relevant captions for images. BLEU scores provide a quantitative measure of how well the generated captions match the reference captions, which directly relates to the project's objective of creating high-quality image descriptions.
* **Standardized Metric:** BLEU scores are a standardized and widely accepted metric in the field of natural language processing (NLP) and image captioning, allowing for comparison with other models and previous work.
* **Multi-Gram Analysis:** By evaluating n-grams of different lengths (from unigrams to four-grams), BLEU scores capture both the overall content and the syntactic structure of the generated captions. This helps in assessing not just the presence of correct words but also the fluency and coherence of the captions.
* **Simplicity and Interpretability:** BLEU scores are relatively simple to compute and interpret, providing clear insights into the performance of the model.

## **Results and Analysis**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **BLEU-1** | **BLEU-2** | **BLEU-3** | **BLEU-4** |
| ***Resnet50*** | 0.1379 | 0.0341 | 0.0225 | 0.0192 |
| ***VGG16*** | 0.1617 | 0.0371 | 0.0246 | 0.0208 |
| ***GoogleNet*** | 0.1438 | 0.0344 | 0.0228 | 0.0191 |
| ***MobileNetV2*** | 0.1469 | 0.0346 | 0.0222 | 0.0183 |

Table 1 - BLEU Scores of all the Models

**Key Insights**

* **Trade-off between Accuracy and Efficiency:** VGG16, while more accurate in generating captions, may be less efficient for deployment on mobile or edge devices due to its larger size and computational requirements. MobileNetV2, on the other hand, offers a good balance between accuracy and efficiency, making it suitable for resource-constrained environments. But for project’s objective, it can be observed [*Figure 2*] that GoogleNet generates better captions than the rest.



*Figure 2- Sample Image Captioned from GoogleNet Model*

* **Importance of Architecture:** The choice of architecture plays a crucial role in the model's performance for image captioning tasks. Models like VGG16, with simpler and deeper architectures, tend to perform better in capturing detailed features and contextual relationships.
* **Challenges in Language Generation:** Generating coherent and contextually accurate captions remains a challenge for all models, indicating the complexity of the language modelling task and the need for further research and improvements in this area. the impact and benefits of the final model on the business use cases.

## **Data Privacy and Ethical Concerns**

Data privacy and ethical considerations were integral to the project's design and implementation. By ensuring data anonymization, mitigating biases, adhering to ethical guidelines, and maintaining transparency, the project aimed to uphold the highest standards of data privacy and ethical conduct. These steps not only protected the privacy of individuals depicted in the images but also contributed to the development of fair and responsible machine learning models. With the context of the Flickr8k dataset, it is an open-source dataset which was released under a creative common license to be used by anyone, ensuring the privacy concerns.

# Conclusion

The project successfully developed and evaluated image captioning models using the Flickr8k dataset, demonstrating the effectiveness of deep learning techniques in generating accurate and contextually relevant image descriptions. The models were assessed using BLEU scores, providing quantitative measures of performance. The project also prioritized data privacy and ethical considerations, ensuring responsible and fair use of data and technology.

**Future Work and Recommendations**

* **Fine-tuning Models:**

Fine-tuning the models on specific datasets, such as running for more epochs or adjusting the optimizer or using different pre-processing techniques such as advanced tokenization could improve their performance in generating more accurate and coherent captions.

* **Ensemble Methods:**

Exploring ensemble methods, combining predictions from multiple models, could potentially improve the overall caption prediction performance.

* **Advanced Architectures:**

Investigating more advanced architectures specifically designed for image captioning, such as Transformer-based models, could lead to better caption generation results.

# References

* Dr.Mir Kabir, Lectures, Lab Exercises for Neural Networks(2024),

<https://colab.research.google.com/drive/1HdE5OijBx8CtCSiqXBsQPqkPfoPv9_9I>

<https://colab.research.google.com/drive/1iYloBODmTBp29aTqyMkiCgILAUAJG3nxC>

<https://colab.research.google.com/drive/1HdE5OijBx8CtCSiqXBsQPqkPfoPv9_9I>

* Suresh, K.R., Jarapala, A. & Sudeep, P.V. Image Captioning Encoder–Decoder Models Using CNN-RNN Architectures: A Comparative Study. Circuits Syst Signal Process 41, 5719–5742 (2022). <https://doi.org/10.1007/s00034-022-02050-2>
* Aryavir07. (2021). Image-Captioning-Using-CNN-and-LSTM.

https://github.com/Aryavir07/Image-Captioning-Using-CNN-and-LSTM

* Neural Networks and Deep Learning - <http://neuralnetworksanddeeplearning.com/chap2.html>
* Lasiman, D. (n.d.). Image Captioning. https://daniel.lasiman.com/post/image-captioning/
* Vinod, S. (2018). a-PyTorch-Tutorial-to-Image-Captioning. GitHub. https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Image-Captioning
* Zhou, J. (2018). Image Captioning with Keras: Teaching Computers to Describe Pictures. Towards Data Science. <https://towardsdatascience.com/image-captioning-with-keras-teaching-computers-to-describe-pictures-c88a46a311b8>
* Brownlee, J. (2019). Develop a Deep Learning Caption Generation Model in Python. Machine Learning Mastery. <https://machinelearningmastery.com/develop-a-deep-learning-caption-generation-model-in-python/>
* Manakov, A. (2019). How to Build an Image Captioning Model in PyTorch. Towards Data Science. <https://towardsdatascience.com/how-to-build-an-image-captioning-model-in-pytorch-29b9d8fe2f8c>