Introduction

- **Objective**: This is a deep learning project to generating image captions using a combined Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) architecture. The aim is to automatically generate descriptive captions for images, enhancing the understanding of the visual content.

- **Architecture**: The architecture consists of three main components: an image feature extraction pathway, a sequential text feature extraction pathway, and a decoder that combines these features to produce captions.

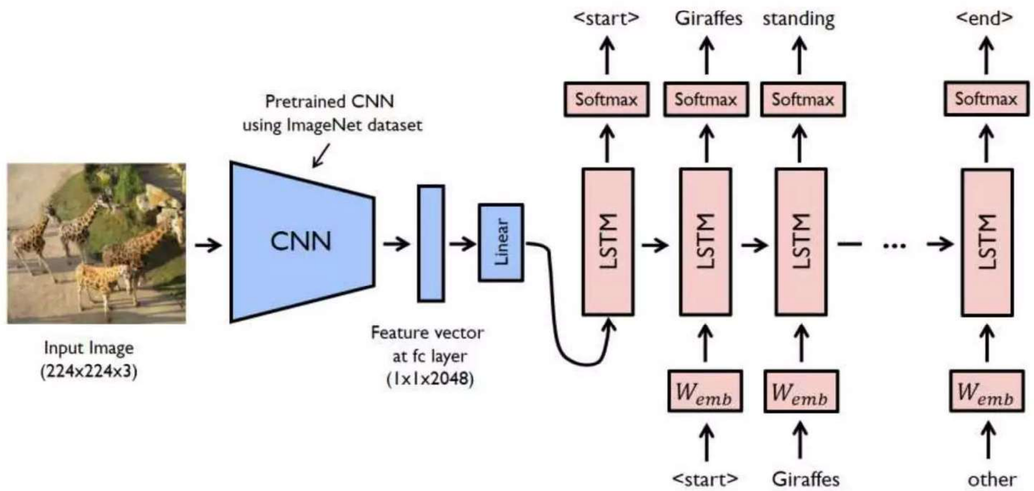
- **Dataset**: Flickr30k dataset, a diverse collection of images paired with human-generated captions. This dataset enables the training and evaluation of the image caption generator, facilitating the learning of meaningful relationships between images and their corresponding textual descriptions.

Proposed Methodology

1. **Image Feature Extraction**
    - First the process involves resizing the images into 224x224 pixels and preprocessing the images for input to the model.
    - Then we extract image features using the **VGG16** model, skipping the last layer since that is used for classification.
    - Extracted features are then stored in a dictionary and serialized to a ‘features.pkl' file using pickle library.

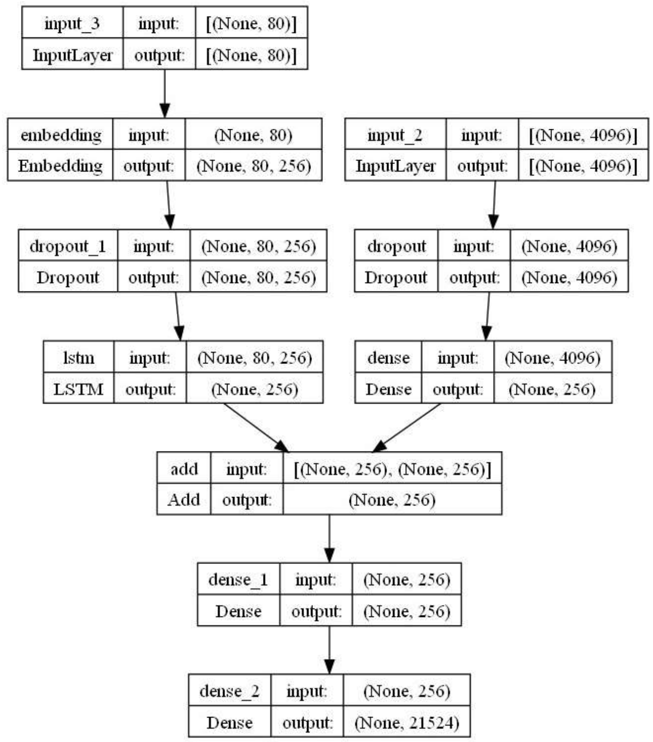
2. **Caption Preprocessing**
    - Converting captions to lowercase, removing non-alphabetic characters (except spaces), extra spaces are then replaced with a single space.
    - Special tokens, 'startseq' and 'endseq', are added to indicate the beginning and end of captions during model training.
    - The Tokenizer() is fitted on preprocessed captions to convert text data into numerical sequences, which will also be used to build vocabulary.
    - The size of the vocabulary is determined by counting unique words in the tokenizer's word index.
    - The fitted tokenizer is serialized and saved as 'tokenizer.pkl', allowing consistent preprocessing of new data with the same vocabulary.
    - Also, maximum length (in words) across all preprocessed captions is calculated, which will help in padding.
    - A list of image names is created, the dataset is divided into training and testing, with an 80-20 split ratio.

3. Model Architecture



- **CNN Encoder**
  - The first part of the model is a CNN that takes in an input image and extracts high-level features from it. These features are then transformed into a fixed-length vector representation.
- **RNN Decoder**
  - The second part of the model is an RNN. It takes the fixed-length vector representation from the CNN and generates a sequence of words one at a time to form a coherent caption. At each step, the RNN generates the next word based on its previous output and a hidden state that maintains context.
- **Embedding Layer**
  - Words in the caption are usually represented as embeddings, which are learned representations of words in a continuous vector space. The embeddings help capture semantic relationships between words.
- **Loss Function**
  - The model is trained using a loss function that measures the dissimilarity between the generated caption and the ground truth caption.

4. Implementing The Model



- **Encoder (right side)**
  - Extracted images features of dimension 4096 are input to the encoder.
  - A dropout layer with a 40% dropout rate is applied to the image features.
  - The dropout output is connected to a Dense layer with 256 units and ReLU activation.
- **Sequence Feature Layer (left side)**
  - Text data sequences, with a maximum length calculated earlier, serve as input.
  - An embedding layer converts tokenized sequences into dense 256-dimensional vectors.
  - A dropout layer with a 40% dropout rate is applied to the embedded sequences.
  - An LSTM layer with 256 units processes the sequences.
- **Decoder**
  - The outputs of the image encoder and sequence feature layer are element-wise added.
  - The result is passed to a Dense layer with 256 units and ReLU activation.
  - A final Dense layer with a softmax activation generates predicted token probabilities from a vocabulary

5. Generating Captions

- First we convert the integer index back to its corresponding word in the vocabulary of the tokenizer.
- The we predict the caption using the trained model, tokenizer, and maximum caption length.
- It starts with an initial input text 'startseq' and iteratively predicts the next word in the caption until it encounters the 'endseq' token or reaches the maximum caption length.

```
generate_caption('8212843678.jpg')
```

```
--------------------Actual--------------------
startseq a biker demonstrates his talents in the air by placing only his left arm on the bike seat endseq
startseq a motocross rider performs a trick in the air at a stadium as the sun sets endseq
startseq dirt bike racer in the air with one hand on the dirt bike and body in air endseq
startseq a professional motocross rider is performing a midair stunt on his bike endseq
startseq a stuntman jumping a motorcycle in a stadium endseq
--------------------Predicted--------------------
startseq a man is riding a bicycle on a tightrope endseq
```


```
generate_caption('8204629082.jpg')
```

```
--------------------Actual--------------------
startseq a man is singing on stage wearing a white shirt and a black vest and black pants while holding a guitar endseq
startseq an older gentleman wearing a white shirt and black vest is playing a guitar while singing endseq
startseq a man in a black vest is playing a guitar and another man playing a guitar beside him endseq
startseq a balding man is playing guitar and singing endseq
startseq a band of older men perform live on stage endseq
--------------------Predicted--------------------
startseq a man in a suit is playing a guitar endseq
```

6. Evaluation

- **BLEU Score**

  - Bilingual Evaluation Understudy is a widely used metric to evaluate the quality of machine-generated text, including image captions.
  - It compares the generated caption with reference captions provided by human annotators and calculates the precision of n-grams (typically up to 4-grams) in the generated caption compared to the reference captions.
  - For **BLEU-1**, only unigram matches are considered (weight = 1.0).
  - For **BLEU-2**, bigram matches are considered with equal importance (weight = 0.5 for both unigrams and bigrams).
  - BLEU score is a value between 0 and 1. A score of 1 means the generated text perfectly matches the reference text, while a score of 0 means no overlap. So, **higher BLEU scores indicate better similarity between generated and reference text.**

```
print("BLEU-1: %f" % corpus_bleu(actual[:3178], predicted[:3178], weights=(1.0, 0, 0, 0)))
print("BLEU-2: %f" % corpus_bleu(actual[:3178], predicted[:3178], weights=(0.5, 0.5, 0, 0)))
```

```
BLEU-1: 0.623424
BLEU-2: 0.412801
```

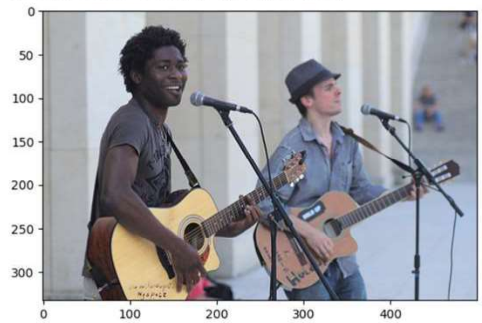
- **Training**

  - GPU Used: Nvidia RTX 3060 Laptop
  - Training Time: 3 hours for 30 epochs

Result

```
generate_caption('7983388093.jpg')
```

```
--------------------Actual--------------------
startseq two men one africanamerican and one white play acoustic guitars and sing into microphones in an outdoor setting endseq
startseq two men are singing and playing guitars outside a cement structure with spectators sitting on steps on their left endseq
startseq two men one black and one white play their guitars and sing into microphones as they stand outdoors endseq
startseq a black man and a white man with acoustic guitars are standing in front of microphones endseq
startseq two males one white and one black singing and playing the guitar endseq
-------------------Predicted-------------------
startseq a man in a black shirt plays a guitar endseq
```




```
generate_caption('8131954252.jpg')
```

```
--------------------Actual--------------------
startseq a woman in black bicycle gear and a white helmet pedals hard uphill on her bike endseq
startseq a cyclist wearing sunglasses and a silver bicycle helmet competing in a race endseq
startseq a woman wearing red and black biking gear biking up a hill endseq
startseq a bike racer following the racing trail uphill endseq
startseq a woman cycling up a hill endseq
-------------------Predicted-------------------
startseq a man in a blue shirt is riding a bike endseq
```


```
startseq a woman in a black shirt and sunglasses is taking a picture endseq
```

Conclusion

In this study, we successfully explored the utilization of a combined CNN (VGG16) and RNN (LSTM) model for the purpose of generating image captions. We were able to achieve a commendable BLEU-1 score of 0.6234. This score indicates a meaningful level of accuracy in terms of matching generated captions with human-written references. The project highlights the potential of merging convolutional and recurrent neural networks to tackle the complex task of image captioning.

References:

- Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator.