Image Captioning Model Report

G. Vineeth

July 14, 2024

1 Introduction

Image captioning is the task of generating descriptive textual captions for images. This report outlines a deep learning model designed for image captioning using MobileNetV2 for feature extraction and a combination of LSTM and attention mechanisms for sequence generation. The model is built to convert images into meaningful textual descriptions.

2 Dataset and Preparation

2.1 Paths and Loading

2.1.1 Image Paths

The dataset is organized into distinct sets for training, validation, and testing, each stored in separate directories for easy access and management.

2.1.2 Annotation Paths

Each dataset split (training, validation, testing) includes JSON files containing annotations detailing object labels and bounding boxes for the images.

2.1.3 Google Drive Integration

Google Drive is mounted to facilitate seamless access to the dataset for processing and analysis.

2.2 Image and Feature Extraction

2.2.1 Image Loading

Images are loaded from their directories and resized to 224x224 pixels to ensure uniformity. They are then normalized to standardize pixel values and converted into numerical arrays for further processing.

2.2.2 Feature Extraction

Images undergo feature extraction using MobileNetV2 with average pooling. This convolutional neural network architecture generates compact, high-level representations of each image, preserving essential information while reducing dimensionality.

2.3 Annotations and Captions

2.3.1 Flattening

Annotation data in JSON files is flattened to simplify its hierarchical structure into a list-based format, enhancing manageability for training and evaluation.

2.3.2 Preprocessing

Captions associated with images are preprocessed for sequence generation models. Special tokens are added at the beginning and end of each caption to denote sequence start and end. These tokens aid in training the model to generate accurate and coherent textual descriptions from image features.

3 Tokenisation and Sequence Generation

3.1 Tokenization

Text Conversion: Tokenization is the process of converting text captions into numerical sequences that can be fed into machine learning models. This involves using a tokenizer, which maps each unique word or token in the text to a specific index in a vocabulary.

Special Tokens: During tokenization, special tokens are introduced to help manage the sequence generation process. These include:

- **Start Token:** Added at the beginning of each caption sequence to signal the start of the text generation process.
- **End Token:** Added at the end of each caption sequence to indicate the termination of the text generation process.

Vocabulary Size: The tokenizer builds a vocabulary based on the unique words or tokens found in the dataset. The size of this vocabulary is determined, which represents the total number of unique tokens that the model will recognize and use. This vocabulary size plays a crucial role in defining the model's input and output space.

3.2 Sequence Creation

Input Sequences: Once the captions are tokenized, they are transformed into sequences of word indices. To ensure that all sequences are of uniform length, padding is applied where necessary. Padding involves adding a specific value (usually zeros) to shorter sequences so that they match the length of the longest sequence in the dataset. This step is important for maintaining consistency in the input data when training the model.

Output Sequences: For training purposes, the tokenized sequences are split into input-output pairs. This means that each sequence is divided into an input sequence (which includes the start token and the words up to a certain point) and an output sequence (which starts from the next word and continues to the end token). This structure allows the model to learn to predict the next word in the sequence based on the preceding words, which is essential for generating coherent text from image features.

4 Model Architecture

4.0.1 Feature Extractor

- **Input:** The feature extractor takes as input the feature vectors derived from the images using MobileNetV2. These vectors encapsulate the high-level content of the images and are used as a basis for generating descriptive text.
- **Processing:**
  - **Dropout Layers:** Dropout is employed to prevent overfitting by randomly setting a fraction of the input units to zero during training. This helps the model generalize better by avoiding reliance on any particular set of features.
  - **Dense Layers:** Dense (fully connected) layers process the feature vectors to transform them into a format suitable for use with the sequence model. The features are often repeated or reshaped to match the length of the word sequences. This ensures that each word in the sequence has a corresponding feature representation.

4.0.2 Sequence Model

- **Input:** This component takes sequences of words (captions) as input. Each word in the sequence is typically represented by an integer index and then converted into a dense vector through an embedding layer.
- **Processing:**

- **Embedding Layer:** Converts integer word indices into dense, continuous-valued vectors. These embeddings capture semantic information about the words and help the model understand their meanings in the context of the sequence.
- **Dropout Layers:** Similar to the feature extractor, dropout is used here to prevent overfitting. It randomly omits some of the input units during training to encourage the model to learn more robust features.
- **LSTM Layers:** Long Short-Term Memory (LSTM) layers are used to model sequential dependencies in the word sequences. LSTMs can remember information for long periods, making them well-suited for handling sequential data like sentences. They process the embedded word vectors and maintain a state that captures the context of the sequence.

4.0.3 Attention Mechanism

- **Attention Scores:** The attention mechanism calculates scores to determine how much focus each word in the sequence should have on different parts of the image feature vectors. This is typically done using dot products between the sequence vectors (hidden states of the LSTM) and the image feature vectors. The scores indicate the relevance of different parts of the image to each word in the sequence.
- **Context Vector:** The context vector is a weighted sum of the image feature vectors, where the weights are derived from the attention scores. This vector represents the image content that is most relevant to the current word in the sequence and helps the model generate more accurate and contextually appropriate descriptions.

4.0.4 Decoder

- **Processing:**
  - **Combination of Context Vector and Sequence Output:** The decoder combines the context vector (representing the relevant image features) with the output of the sequence model (the LSTM's hidden states). This combination helps in producing more informed predictions for the next word in the sequence.
  - **LSTM Layer:** An additional LSTM layer processes the combined information from the context vector and the sequence model output. This layer helps to maintain the sequence's contextual flow while integrating the relevant image features.
  - **Output:**
    - **Dense Layer:** The final dense layer uses the output of the LSTM to predict the next word in the sequence. This layer typically has a softmax activation function, which outputs a probability distribution over the vocabulary, allowing the model to select the most likely next word based on the context provided by the image features and previous words.
- The model architecture integrates various components to effectively generate textual descriptions from images. The feature extractor captures high-level image content, the sequence model processes and understands the sequence of words, the attention mechanism highlights relevant parts of the image for each word, and the decoder generates the final output sequence by combining these elements.

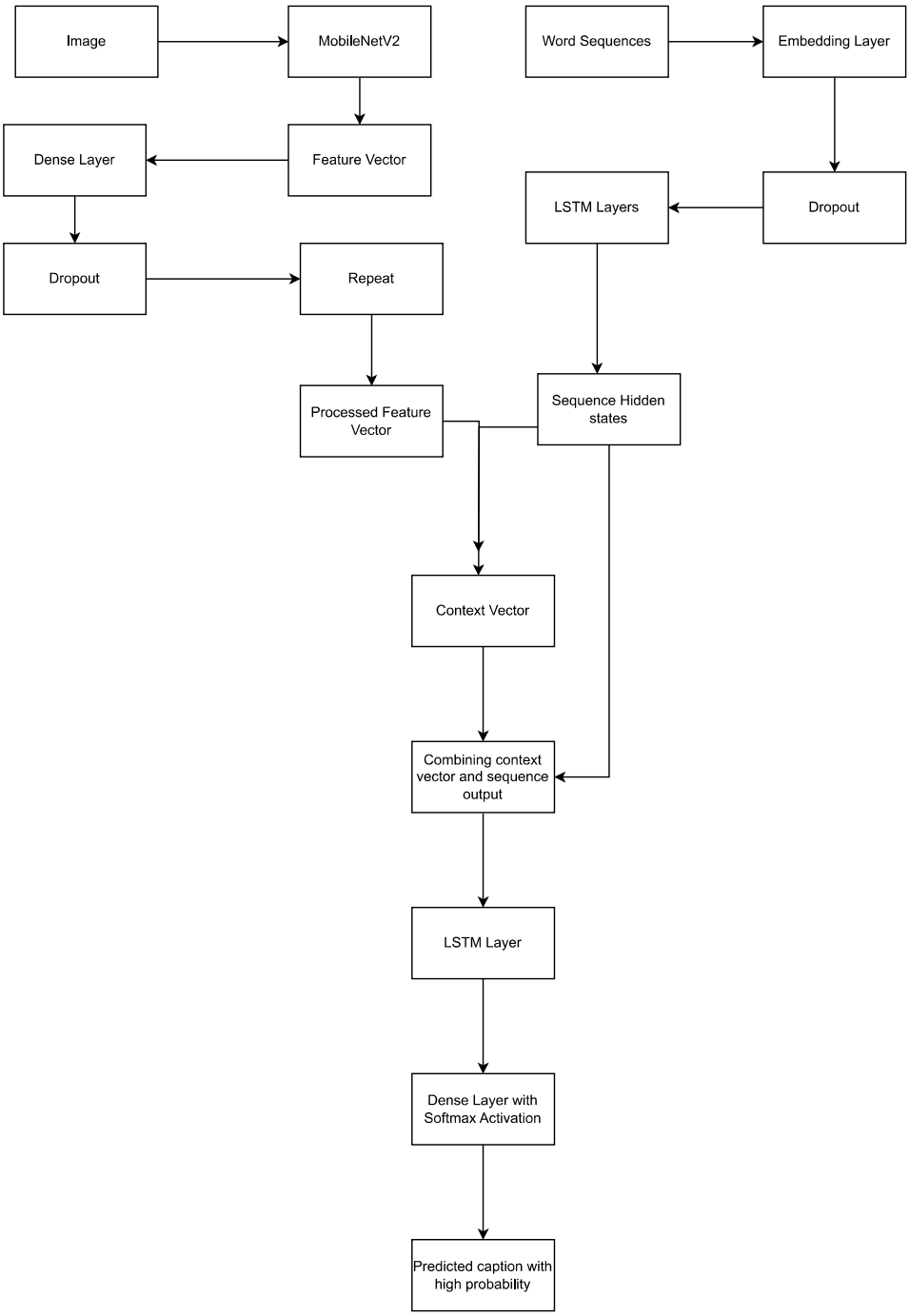

Fig : Model Architecture

5 Training and Caption Generation

5.0.1 Model Training

Training Process:

- **Epochs:** The model undergoes training over a defined number of epochs, which in this case is set to 30. An epoch represents one complete pass through the entire training dataset.

- **Training Data:** During each epoch, the model learns from the training data by adjusting its weights based on the loss function. This involves using backpropagation and optimization techniques to minimize the prediction error.
- **Validation:** Throughout the training process, the model's performance is periodically evaluated on a separate validation set. This helps in monitoring the model's generalization capability and detecting any overfitting or underfitting issues. The validation data is not used for training but to assess how well the model is performing on unseen data.

5.0.2 Caption Generation

Beam Search:

- **Purpose:** Beam search is an advanced search algorithm used for generating sequences, such as image captions. It aims to find the most probable sequence of words that best describes an image.
- **Process:** Instead of generating just one sequence, beam search maintains a set of the most promising sequences (beams) at each step. It expands each beam by considering all possible next words and keeps only the top sequences with the highest probability scores. This approach balances between exploring different possible sequences and exploiting the most likely sequences, thus improving the quality of the generated captions.

Post-processing:

- **Cleaning Captions:** After generating captions using beam search, post-processing is performed to refine the output. This step involves removing redundant phrases that may have been generated due to repeated or unnecessary words.
- **Punctuation and Grammar:** Proper punctuation and grammatical correctness are ensured. This involves adding or correcting punctuation marks, capitalizing the first letter of sentences, and making sure the caption adheres to grammatical rules. This step is crucial for making the captions more readable and natural.

.

6 Evaluation

6.0.1 Metrics

BLEU Score:

- **Definition:** The BLEU (Bilingual Evaluation Understudy) score is a metric used to evaluate the quality of text generation, particularly in the context of machine translation and image captioning. It compares the generated captions against a set of reference captions (ground truth) to determine how closely they match.
- **Calculation:** The score is calculated based on the precision of n-grams (sequences of n words) in the generated captions relative to the reference captions.

ROUGE Score:

- **Definition:** ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics for evaluating the quality of generated text by comparing it to reference texts. It focuses on the overlap of n-grams between the generated captions and reference captions.
- **Calculation:** The ROUGE score typically includes measures like ROUGE-N (for n-grams), ROUGE-L (for the longest common subsequence), and ROUGE-W (for weighted n-grams).

7 Conclusion

- The image captioning model demonstrates the capability to generate descriptive captions for images by combining feature extraction with sequence-to-sequence learning and attention mechanisms. The model effectively learns to understand and describe the content of images, converting visual information into meaningful textual descriptions.
- The evaluation metrics, including BLEU and ROUGE scores, provide quantitative insights into the model's effectiveness in generating captions that closely resemble the reference captions. These metrics highlight areas where the model excels and areas for potential improvement, guiding future enhancements in model architecture and training strategies.