**Image Captioning using CNN + BiLSTM with Attention Mechanism**

**Abstract**

**Image captioning is the task of generating a textual description for a given image. It bridges computer vision and natural language processing, aiming to mimic the human capability of understanding a scene and articulating it in natural language. This report presents the development of an image captioning model that leverages pre-trained CNNs for feature extraction and combines them with Bidirectional LSTMs and an attention mechanism for robust caption generation. The model architecture is inspired by recent advances in deep learning-based sequence modeling.**

**1. Introduction**

**Image captioning is a challenging AI task that requires both visual recognition and natural language understanding. It has applications in assisting the visually impaired, automated content creation, and enhancing accessibility in digital platforms.**

**In this project, we implemented an encoder-decoder architecture:**

* **The encoder extracts visual features from images using a pre-trained CNN (VGG16 in this case).**
* **The decoder processes these features with a sequence model to generate meaningful captions word-by-word.**

**We integrate an attention mechanism to dynamically focus on different parts of the image during caption generation, which significantly improves contextual relevance.**

**2. Dataset and Preprocessing**

* **Dataset Used: Flickr8k**
* **Image Preprocessing: Images are resized and passed through VGG16 (excluding the final classification layer) to obtain a 4096-dimensional feature vector.**
* **Text Preprocessing: Captions are tokenized, converted to lowercase, stripped of punctuation, and then padded to a maximum length. Special tokens such as <start> and <end> are used to mark the beginning and end of a sentence.**
* **Vocabulary Creation: A tokenizer was fitted on all captions to create a word-to-index and index-to-word dictionary.**

**3. Model Architecture**

**Encoder**

* **Input: 4096-dimensional vector from VGG16.**
* **Dropout (0.5): Helps prevent overfitting.**
* **Dense (256): Reduces feature vector dimensionality.**
* **RepeatVector: Repeats image vector to match caption sequence length.**
* **BiLSTM: Processes the repeated image vector to extract sequential context from both directions.**

**Textual Sequence Processor**

* **Input: Sequence of word indices.**
* **Embedding Layer: Maps words to 256-dimensional vectors.**
* **Dropout: Regularization.**
* **BiLSTM: Captures both past and future word contexts in the sequence.**

**Attention Mechanism**

* **Dot product is computed between image features and caption sequence features to get attention scores.**
* **Softmax is applied to normalize attention weights.**
* **A context vector is created using tf.einsum, which performs weighted sum of caption features.**

**Decoder**

* **The context vector and the original image encoding are concatenated.**
* **Dense (256): Transforms the combined vector.**
* **Final Dense: Applies softmax activation to predict the next word.**

**Model Compilation**

* **Loss Function: Categorical Crossentropy**
* **Optimizer: Adam**

**4. Training Details**

* **The model was trained using teacher forcing, where the true previous word is fed at each time step.**
* **Evaluation metric used is BLEU Score to quantify caption similarity to reference.**
* **Trained over multiple epochs with early stopping to avoid overfitting.**

**5. Flow Diagram of the Architecture**

**Image features (4096D) -> Dense -> Repeat -> BiLSTM**

**↓**

**Dot product (Attention)**

**↓**

**Caption input -> Embedding -> BiLSTM -> Attention weighted vector**

**↓**

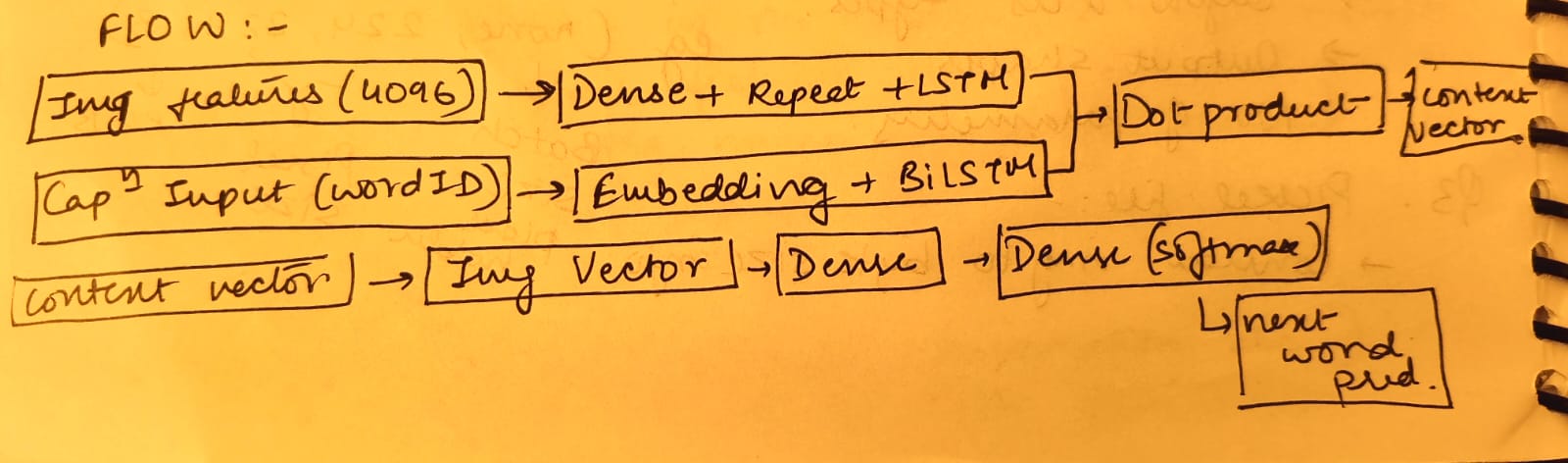
**Concatenate with image vector**

**↓**

**Dense -> Softmax**

**↓**

**Next Word Prediction**



**6. Pseudocode for the Model**

**1. Input image features from CNN (shape: 4096)**

**2. Apply Dropout and Dense layer to project features to 256 dimensions**

**3. Repeat this vector to match the max caption length**

**4. Pass repeated vector through BiLSTM (encoder)**

**5. Input caption sequence as word indices**

**6. Pass through Embedding layer (embedding\_dim = 256)**

**7. Apply Dropout and BiLSTM (decoder)**

**8. Calculate dot-product attention between encoder and decoder outputs**

**9. Apply softmax to get attention weights**

**10. Compute context vector as weighted sum of decoder outputs using attention weights**

**11. Reduce context vector by summing along time axis**

**12. Concatenate context vector with original image feature vector**

**13. Pass through Dense layers to generate probability distribution over vocabulary**

**14. Use softmax to select the most probable word**

**7. Evaluation and Results**

* **The model generated reasonably accurate captions for test images.**
* **BLEU scores were used to validate performance.**
* **Attention mechanism allowed better alignment between visual content and words generated.**

**Example Output:**

* **Image: A dog playing with a frisbee.**
* **Predicted Caption: "A dog catches a frisbee in the park."**

**8. Tools and Libraries**

* **TensorFlow / Keras**
* **Numpy / Pandas / Matplotlib**
* **NLTK for BLEU score**
* **Pre-trained VGG16 from Keras applications**

**9. Challenges Faced**

* **Sequence padding/truncation inconsistencies.**
* **Overfitting on small datasets.**
* **Aligning temporal sequences with image encodings.**

**10. Conclusion**

**This project demonstrates how combining CNNs with BiLSTM and attention mechanisms allows for generating context-aware image captions. The encoder-decoder design, attention logic, and sequential learning provide a solid foundation for future enhancements such as Transformer-based models.**

**11. Future Work**

* **Using larger datasets like MS COCO.**
* **Integrating Transformer-based decoders.**
* **Improving attention granularity with multi-head attention.**
* **Applying the model in multilingual settings.**

**12. References**

1. **Show, Attend and Tell – Xu et al.**
2. **VGGNet – Simonyan & Zisserman**
3. **Keras Documentation**
4. **BLEU Score – Papineni et al.**

**Submitted by:** *Aarya Tiwari*  
**Roll No:** *E23CSEU0724*  
**Course:** *ARITIFICIAL INTELLIGENCE*