Image Captioning using CNN and LSTM on Flickr8k Dataset

Computer Vision Project

2024PCS0026 – Hritik Som
2024PCS0031 – Neetish Bamotra
2024PCS0041 – Uma Shankar Sharma
2024PCS0043 – Yash Agarwal
2024PCS0123 – Ayushman Jadon

Department of Computer Science and Engineering

November 2025

Guided By: (Dr. Vinit Jakhetiya)

Outline

1. Introduction
2. Objective
3. Dataset
4. Preprocessing
5. Architecture
6. Implementation
7. Training and Results
8. Challenges and Learnings
9. Conclusion and Future Work

What is Image Captioning?

- Automatically generating natural language descriptions for images.
- Combines Computer Vision (for visual feature extraction) and NLP (for text generation).
- Applications: accessibility tools, image search, robotics, and media analysis.

Project Objective

- Develop a model that generates captions for unseen images.
- Use CNN (for feature extraction) and LSTM (for text decoding).
- Evaluate qualitative and quantitative performance.

Flickr8k Dataset

- Contains 8,000 images, each annotated with 5 captions.
- Commonly used benchmark for small-scale captioning systems.
- Split: 6000 training, 1000 validation, 1000 testing.

Image Preprocessing

- Resize images to 224x224 to match CNN input requirements.
- Normalize pixel values between 0 and 1.
- Extract CNN features and save them to disk.

Caption Preprocessing

- Lowercase all text and remove punctuation.
- Tokenize words and build vocabulary.
- Add `<start>` and `<end>` tokens.
- Pad all captions to a fixed sequence length.

Model Overview

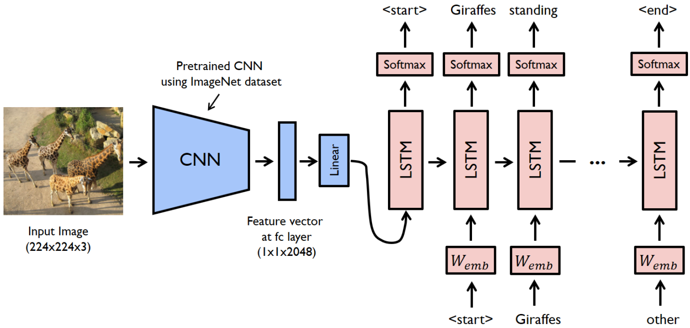


Figure: CNN (Feature Extractor) + LSTM (Decoder) Architecture

CNN: Feature Extractor

- Used pretrained DenseNet201 (transfer learning).
- Removed classification layers; extracted 1920-dimensional features.
- Added Dense layer to reduce dimensionality before LSTM input.

LSTM: Decoder Network

- Embedding layer (256 units) converts words into vectors.
- LSTM (256 units) learns temporal dependencies.
- Dense layer predicts next word using softmax activation.

Libraries and Tools

- Python, NumPy, Pandas
- TensorFlow / Keras for deep learning
- NLTK for text preprocessing
- Matplotlib for visualization

Training Details

- Optimizer: Adam (lr = 0.001)
- Batch size: 64
- Epochs: 50 with EarlyStopping
- Loss: Categorical cross-entropy

Training Loss Curve

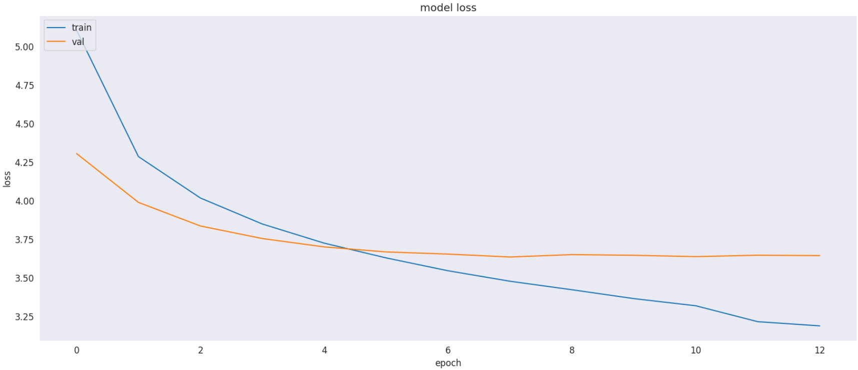
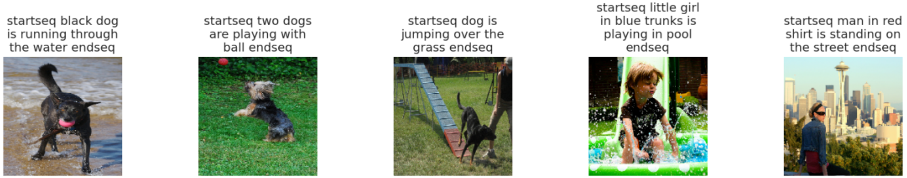


Figure: Train vs Validation Loss Across Epochs

Results



startseq black dog is running through the water endseq

startseq two dogs are playing with ball endseq

startseq dog is jumping over the grass endseq

startseq little girl in blue trunks is playing in pool endseq

startseq man in red shirt is standing on the street endseq

Challenges Faced

- Dataset too small for generalization.
- High memory usage due to LSTM sequences.

Key Learnings

- Feature extraction saves GPU time.
- Text cleaning and tokenization impact performance heavily.
- Transfer learning significantly improves results.

Conclusion

- Built a complete image captioning pipeline using CNN + LSTM.
- Achieved reasonable accuracy and coherent sentence generation.
- Demonstrated integration of vision and language models.

Future Enhancements

- Integrate attention mechanisms.
- Fine-tune CNN backbone.
- Switch to Transformer decoders for better sequence modeling.

References

- Flickr8k Dataset by Hodosh et al., 2013.

Acknowledgements

- Faculty Guide: Dr. Vinit Jakhetiya
- Institute: IIT Jammu
- Contact: vinit.jakhetiya@iitjammu.ac.in

Thank You! Questions?