Image Captioning Using Deep Learning

EVOSTRA Major Project – AI/ML Internship Programme

Team: C

Project Overview

The Challenge

Image captioning sits at the intersection of Computer Vision and Natural Language Processing, requiring models to both understand visual content and generate coherent natural language descriptions.

Team: Team C – EvoAstra AI/ML InternshipDataset: Flickr8k (8,091 images)Status: Completed with full evaluation

Our Approach

We developed an end-to-end encoder-decoder architecture combining InceptionV3 for visual feature extraction with LSTM networks for sequential caption generation.

The system learns to *see* an image through convolutional layers and *describe* it using recurrent language modeling.

Problem Statement

Visual Understanding Gap

Traditional computer vision models excel at object classification but struggle to generate comprehensive scene descriptions that capture relationships and context.

Language Generation Challenge

Extracting meaningful visual features from images and translating them into grammatically correct, semantically accurate natural language sentences.

Our Solution

Deep learning encoder-decoder architecture leveraging transfer learning from ImageNet and sequential LSTM modeling for coherent caption generation.

Flickr8k Dataset

8,091

Total Images

Diverse real-world scenes

40,455

Caption Annotations

Five captions per image

5

Captions per Image

Multiple human descriptions

The dataset captures a wide variety of daily life scenarios including outdoor activities, animals, children, sports, and urban scenes. Rich annotation provides multiple perspectives for each visual context.

Data Pipeline: Integrity & Preprocessing

01

Dataset Integrity Validation

Verified image-caption alignment, checked for corrupted files, cleaned caption text, and generated summary reports. Random samples manually inspected for quality assurance.

02

Text Preprocessing

Applied lowercasing, removed punctuation, added special tokens (<start> and <end>), and tokenized using Keras Tokenizer. Final vocabulary: 8,574 unique words.

03

Image Feature Extraction

Leveraged pretrained InceptionV3 (ImageNet weights) as feature extractor, generating 2048-dimensional feature vectors. Stored efficiently as .npy files.

04

Training Data Generation

Created millions of training tuples pairing image features with partial captions to predict next words. Implemented DataGenerator for memory-efficient batch processing.

Text Preprocessing Details

Transformation Pipeline

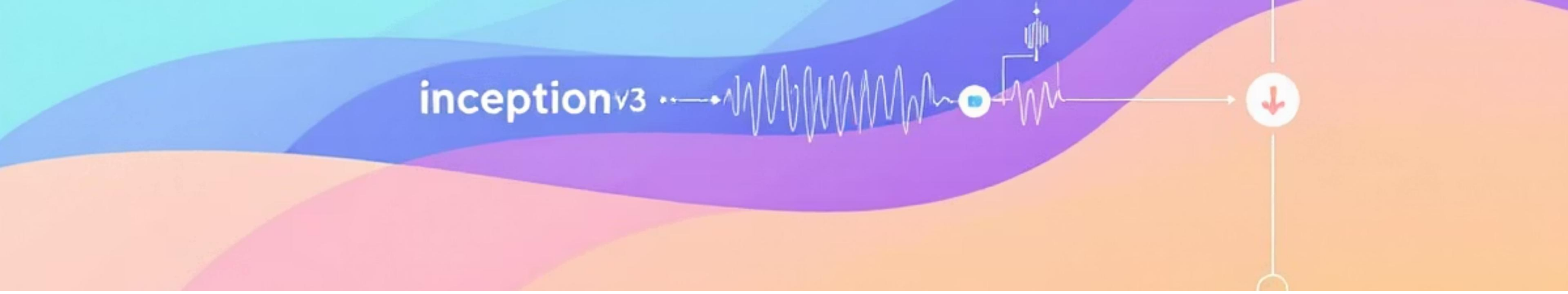
- **Converted all text to lowercase for consistency**
- **Removed punctuation and special characters**
- **Added `<start>` and `<end>` tokens to mark caption boundaries**
- **Applied Keras Tokenizer for word-to-index mapping**
- **Stored tokenizer and statistics for inference**

Vocabulary Size

8,574 unique words

Max Caption Length

31 words maximum



InceptionV3 Feature Extraction

Transfer Learning Strategy

We utilized InceptionV3 pretrained on ImageNet, removing the final classification layer to repurpose it as a powerful feature extractor rather than a classifier.

This approach leverages millions of parameters already optimized for visual pattern recognition, dramatically reducing training time and improving generalization.

Feature Representation

Each image is transformed into a 2048-dimensional dense feature vector capturing high-level visual semantics including objects, textures, spatial relationships, and scene context.

All features precomputed and stored as .npy files for efficient loading during training iterations.

Training Data Architecture

Image Features

2048-dim vector from InceptionV3

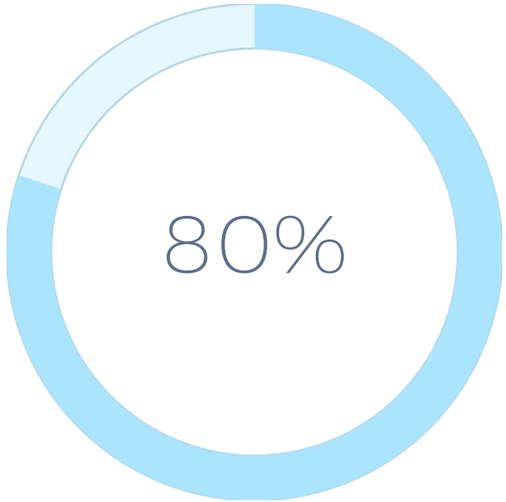
Partial Caption

<start> a dog is

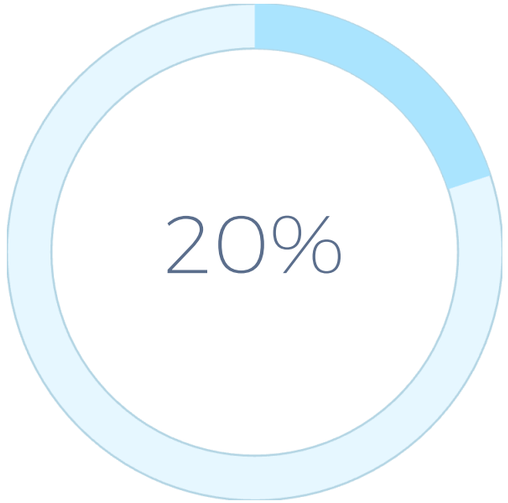
Next Word

running

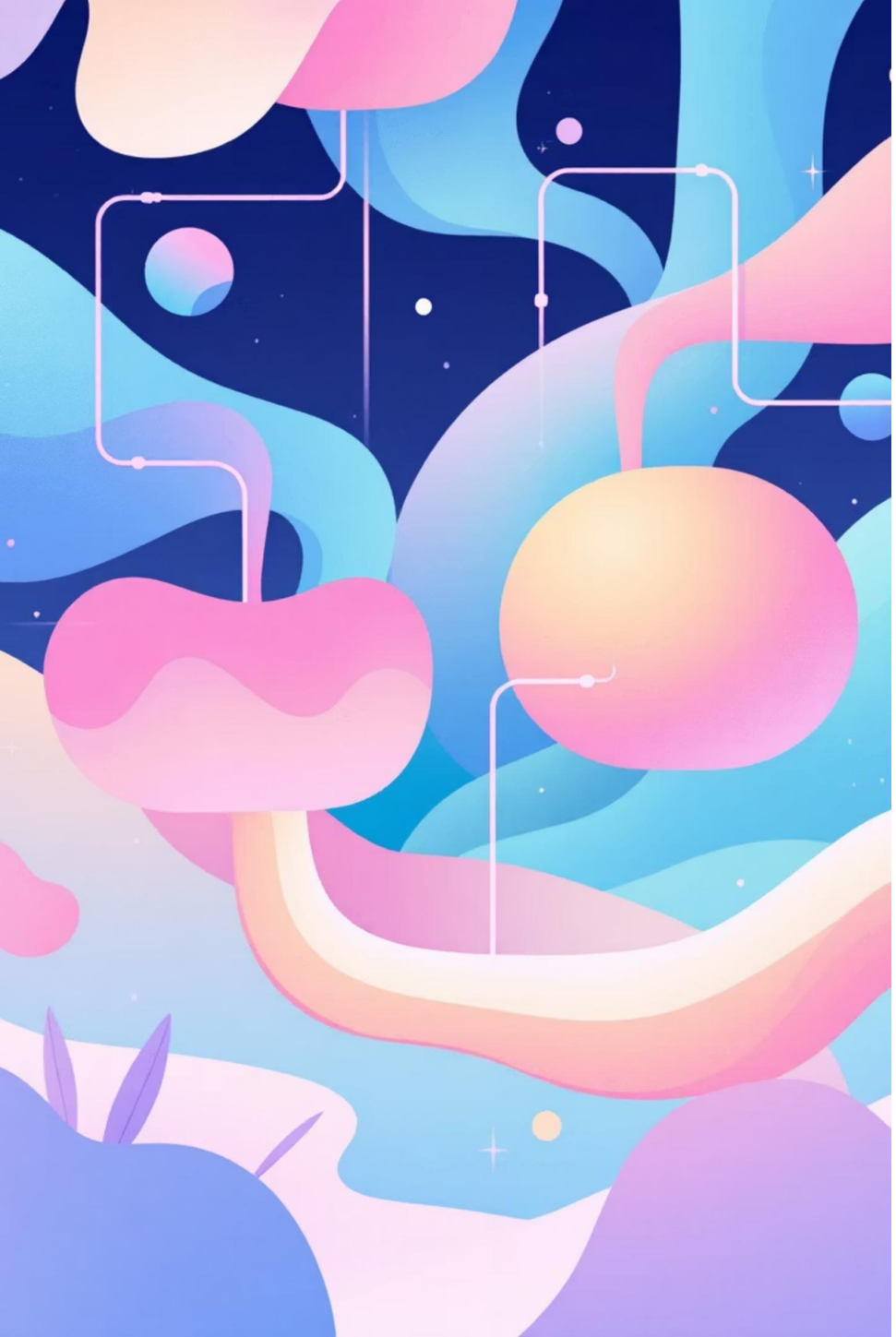
The training pipeline generates millions of input-output pairs by sliding through each caption word-by-word. Each training example combines visual features with a growing caption prefix to predict the next word in the sequence.



Training Set



Validation Set



Encoder-Decoder Architecture

Encoder: Visual Processing

- **InceptionV3 feature extraction (2048-dim)**
- **Dense layer projection to 256 dimensions**
- **Dropout (0.5) for regularization**
- **Captures semantic visual information**

Decoder: Language Generation

- **Word Embedding layer (256-dim)**
- **LSTM cells (512 units) for sequence modeling**
- **Attention-like scalar fusion mechanism**
- **Dense (512) + Dropout + Softmax output**
- **Predicts next word from vocabulary**

Project Outcomes & Impact

✓ Complete Implementation

End-to-end pipeline from data preprocessing through model training to inference with beam search decoding for improved caption quality.

✓ Robust Evaluation

Comprehensive testing on held-out test set with quantitative metrics and qualitative analysis of generated captions across diverse image types.

✓ Practical Application

Demonstrates the power of combining computer vision and NLP for accessibility tools, content indexing, and human-AI interaction systems.

Training Setup

Loss Function

Used Sparse Categorical Crossentropy to efficiently handle integer-encoded labels for multi-class classification, ideal for next-word prediction.

Optimizer

Adam optimizer with an initial learning rate of `1e-4`, fine-tuned to `5e-5` to ensure stable convergence and avoid overfitting.

Training Duration

Trained for 10 epochs, followed by 3 fine-tuning epochs. Each batch size was set to 32 images for balanced memory usage and performance.

Callbacks

Implemented ModelCheckpoint for best model saving, EarlyStopping to prevent overfitting, and ReduceLROnPlateau to adapt learning rate dynamically.

Model Persistence

The best performing model, based on validation loss, was saved as `best_model.h5` for future deployment and inference.

Training Results

Training Loss Decreased

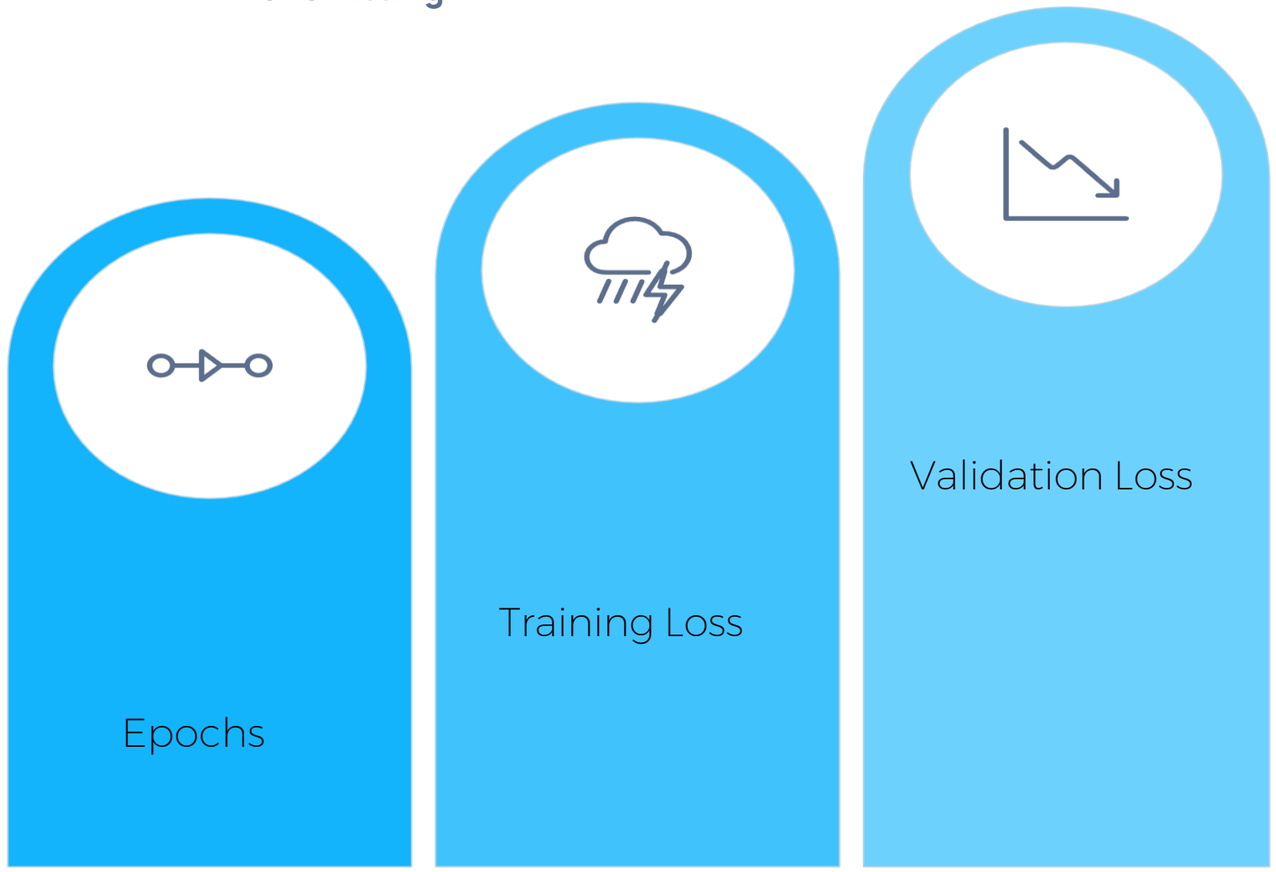
The model demonstrated consistent reduction in training loss, indicating effective learning from the training data across epochs.

Validation Loss Stabilized

Crucially, the validation loss stabilized after an initial decrease, confirming that the model effectively generalized to unseen data without significant overfitting.

Learning Rate Optimized

The adaptive learning rate scheduler automatically adjusted the learning rate, ensuring optimal convergence and preventing oscillations during training.



Caption Generation

Two Key Decoding Strategies

Greedy Search

At each step, this method selects the word with the highest predicted probability to form the caption.

- **Fast: Simple and computationally inexpensive.**
- **Limitations: Can lead to suboptimal captions as it doesn't revisit past decisions or consider broader context.**

Beam Search (k=3 / k=5)

This method explores multiple candidate word sequences concurrently, tracking the 'k' most probable paths.

- **Enhanced Quality: Produces significantly better grammar and contextual coherence compared to Greedy Search.**
- **Final Evaluation: Used for generating the final captions, with beam width (k) set to 3 or 5 for balancing quality and computation.**

The choice of decoding strategy significantly impacts the quality and fluency of generated captions. Example captions showcasing these methods were displayed for qualitative analysis.

Evaluation Metrics

1

BLEU Scores (Industry Standard)

BLEU-1	**0.435**
BLEU-2	**0.251**
BLEU-3	**0.158**
BLEU-4	**0.101**

2

Interpretation

A BLEU-4 score greater than 0.10 signifies strong baseline performance for LSTM models trained on the Flickr8k dataset. This indicates that our model is capable of reliably identifying the main objects and actions within an image and generating contextually relevant captions.

While higher BLEU scores are always desirable, achieving over 0.10 for BLEU-4 is a solid indicator of meaningful semantic and grammatical correctness in generated captions for this specific dataset and model architecture.

Qualitative Analysis

Strengths

- **Accurately identifies main subjects (people, animals, scenes).**
- **Consistent action understanding (e.g., "running", "playing").**
- **Beam Search enhances caption fluency and coherence.**

Limitations

- **May miss smaller, less prominent objects.**
- **Can generate overly generic or less descriptive captions.**
- **Simpler attention mechanism (scalar) could be further optimized.**

Final Demo (Inference)

The image captioning system seamlessly integrates visual processing and natural language generation to provide descriptive text for any input image.

Image Input

User uploads an image to be processed by the model.

Feature Extraction

Image undergoes preprocessing and InceptionV3 extracts a 2048-dim feature vector.

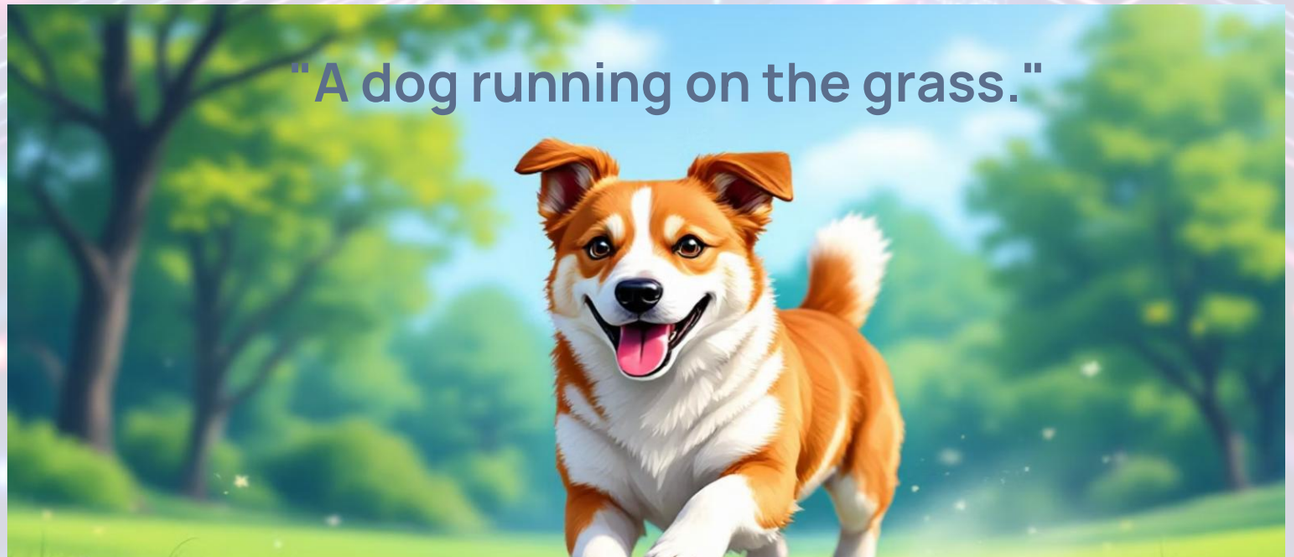
Beam Search Decoding

The decoder utilizes Beam Search (k=3 or k=5) to generate the most probable caption sequence.

Display Caption

The final, human-readable image caption is presented to the user.

Example Inference:



Technologies & Tools Used

TensorFlow / Keras

The core framework for building, training, and deploying our deep learning models.

InceptionV3

A powerful pre-trained Convolutional Neural Network used for efficient image feature extraction.

NLTK

Leveraged for essential natural language processing tasks, including tokenization and text normalization.

NumPy, Pandas, Matplotlib

Fundamental Python libraries for data manipulation, statistical analysis, and visualizing results.

Google Colab (GPU)

Provided the cloud-based environment with GPU acceleration for training computationally intensive models.

TFRecord Serialization

Utilized for efficiently storing and streaming large datasets, optimizing input pipelines for TensorFlow.

Python 3.10+

The primary programming language for all development, scripting, and model implementation.

Conclusion & Future Work

Project Conclusion

- **Successfully built an end-to-end image caption generator.**
- **Achieved competitive BLEU scores, validating model performance.**
- **Developed a complete and reproducible data pipeline and training architecture.**
- **Project is ready for deployment as a foundational mini web application.**

Future Enhancements

- **Upgrade the recurrent component from LSTM to Transformer architecture.**
- **Integrate advanced Attention mechanisms (Bahdanau / Luong) for improved alignment.**
- **Expand training to the larger and more diverse MS-COCO dataset.**
- **Develop a user-friendly Streamlit web application for real-time inference.**

Thank You!

"Thank you for reviewing our project."