TEAM D

Final Project Submission

Evoastra Ventures Pvt. Ltd.

Deep Learning Research Division

Image Caption Generator

Bridging Computer Vision and NLP to narrate images using Deep Learning

PROJECT LEADERSHIP

Team Lead
Rakshada Renapurkar

Co-lead
Subham Maharana

TEAM MEMBERS

- Anurag Ojha
- K Sai Kiran
- Kota Aravind Kumar Reddy
- Nazim Nazir

Generated Caption: "A person surfing on a big blue wave."

Input: Image (Pixels) → Output: Text (Sequence)

THE PROBLEM STATEMENT

What is Image Captioning?

Image captioning is the core task of automatically generating a natural language description for a given image. It bridges the gap between vision and language by translating visual content into coherent text.

Why is it Challenging?

Visual Understanding (CV)

The model must detect objects, actions, attributes, and their spatial relationships within the image scene.

Language Generation (NLP)

The system must formulate grammatically correct and semantically meaningful sentences from extracted features.

Multi-Modal Fusion

Effectively mapping high-dimensional visual data to sequential text data in a single end-to-end differentiable model.

CONTEXT & IMPACT

Motivation & Real-World Applications

Why does image captioning matter? Beyond technical novelty, it bridges the digital divide and powers intelligent systems.

01

Accessibility & Inclusion

Empowers visually impaired users by converting visual scenes into descriptive speech, enabling independent navigation of the web and physical world.

02

Smart Search & Indexing

Enhances digital asset management (DAM) by automatically tagging images based on content, allowing users to search visual databases using natural language queries.

03

Content Moderation

Provides context-aware analysis for social media platforms to identify inappropriate or sensitive content that standard object detection models might miss.

04

Autonomous Agents

Enables robots and digital assistants to "see" and describe their environment, facilitating better human-robot interaction and situational awareness.

ARCHITECTURE

System Architecture

End-to-End Encoder-Decoder Model with Attention Mechanism

Feature Extraction

The **CNN Encoder** (InceptionV3) processes the image to extract high-level visual features. We remove the final classification layer to get a rich feature map representation instead of a class label.

Transfer Learning Spatial Features

Attention Mechanism

Before generating each word, the model computes a **weighted sum** of image features. It learns to "look" at specific regions of the image that are relevant to the current word being generated.

Soft Attention Dynamic Context

Sequence Generation

The **LSTM Decoder** takes the context vector and the previously generated word to predict the next word in the sequence. This continues until an <end> token is produced.

Word-by-Word Long-term Memory

MS COCO Dataset

MICROSOFT COMMON OBJECTS IN CONTEXT

A large-scale object detection, segmentation, and captioning dataset. It is the industry standard benchmark for training image captioning models due to its diversity and scene complexity.

117K
TOTAL IMAGES

80
OBJECT CATEGORIES

5
CAPTIONS PER IMAGE

WHY MS COCO?

Non-iconic views | Complex scenes

CAPTION 1

"A man holding a tennis racket ready to serve the ball on a court."

CAPTION 2

"A close up of a pizza with cheese, basil and tomato sauce on a wooden table."

CAPTION 3

"A herd of elephants walking across a dry grassy field during the day."

CAPTION 4

"A laptop computer sitting on top of a wooden desk next to a notebook."

PIPELINE STAGE 1

Data Preparation Process

Before training, raw images and captions undergo rigorous preprocessing to ensure compatibility with the CNN-LSTM architecture.

1. IMAGE PREP

- Resize to 299x299 (InceptionV3 input)
- Normalize pixel values (-1 to 1)
- Augmentation (Optional)

2. TEXT CLEANING

- Lowercase conversion
- Remove punctuation & numbers
- Filter single characters

3. TOKENIZATION

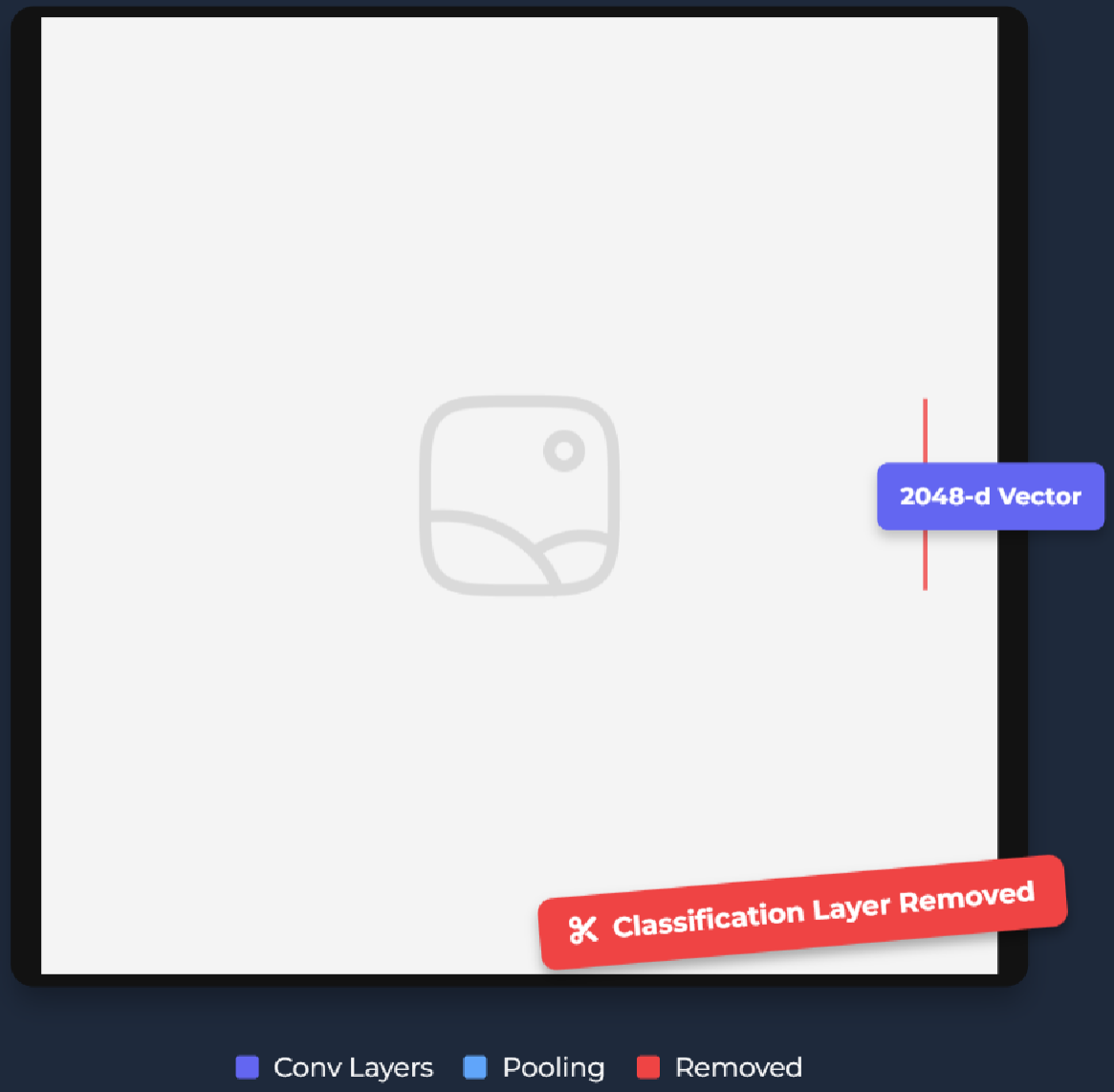
- Add <start>, <end> tokens
- Map words to integers
- Pad sequences to fixed length

4. SPLITTING

- Create Train/Val/Test sets
- Map Image ID to Captions
- Create Generator batches

Training Set (80%) | Validation Set (10%) | Test Set (10%)

ARCHITECTURE DIAGRAM



FEATURE ENCODER

Using InceptionV3 for Vision

We utilize a pre-trained InceptionV3 model as a **feature extractor**. By removing the final classification layer, we convert the image into a dense vector representation that encodes visual content.

INPUT SIZE	FEATURE VECTOR
299 x 299	**2048 dim**
WEIGHTS	PARAMETERS
ImageNet	**~24 Million**

Why Transfer Learning?

- **Faster Training:** Leverages learned features
- **Better Accuracy:** Robust object detection
- **Efficiency:** Reduced computational cost

DECODER MECHANISM

Caption Generation

LSTM-Based Language Model

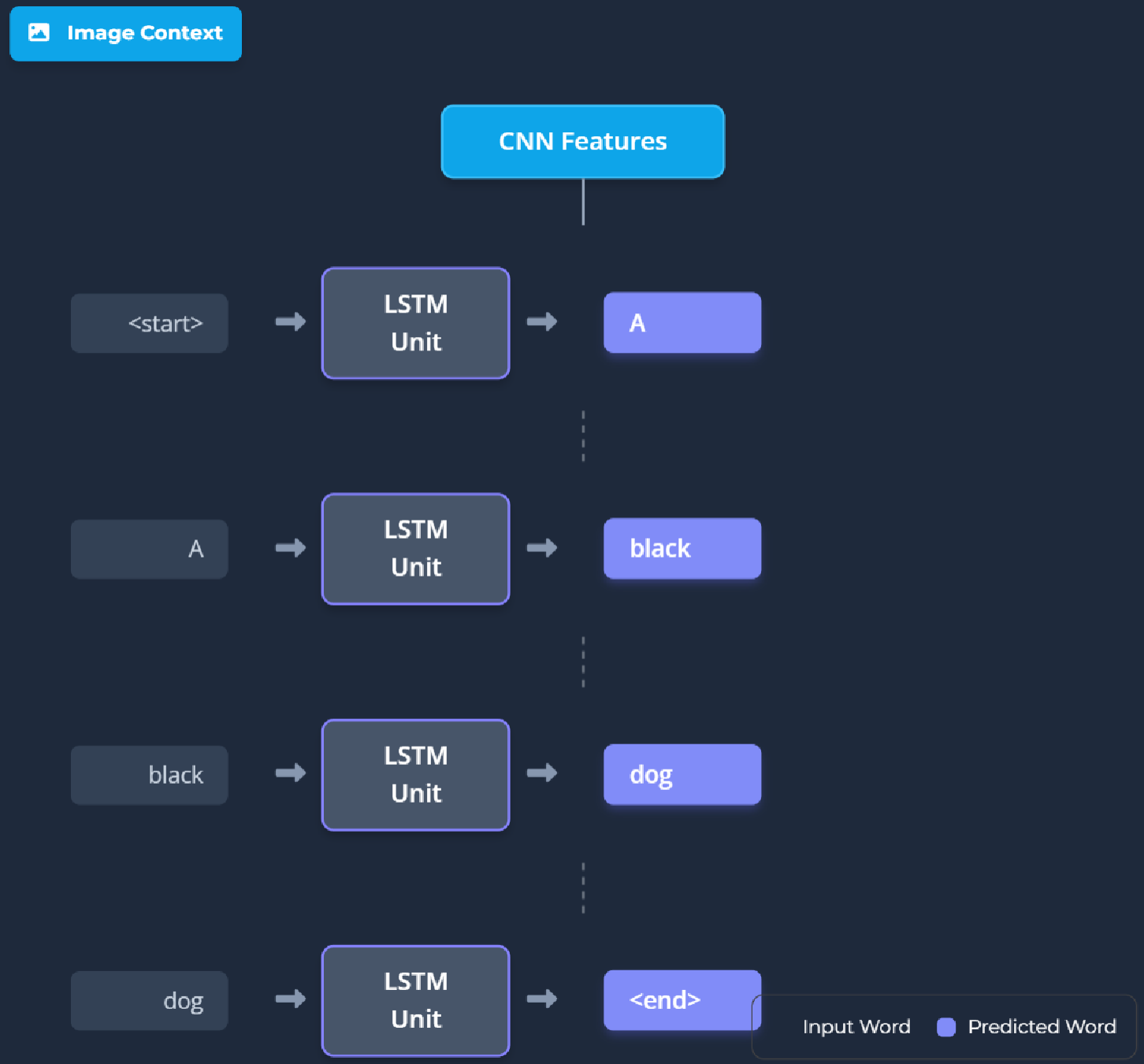
The LSTM decoder acts as the "mouth" of the system. It receives the visual context vector and generates the sentence word by word, learning both grammar and image relevance.

Input Integration
Takes the CNN feature vector as the initial state and the previous word as input for the next step.

Teacher Forcing
During training, the ground-truth word is fed as input for the next step instead of the model's own prediction.

Sequence Learning
Predicts the probability distribution over the vocabulary for the next word until an <end> token is reached.

```
def decoder_step(features, input_word):
  context = lstm(features)
  pred = dense(context, input_word)
  return pred
```



MECHANISM DEEP DIVE

Attention Mechanism

"Look where you speak"

Instead of compressing the entire image into a single static vector, the **Attention Mechanism** allows the decoder to focus on different parts of the image at every step of caption generation.

Dynamic Focus

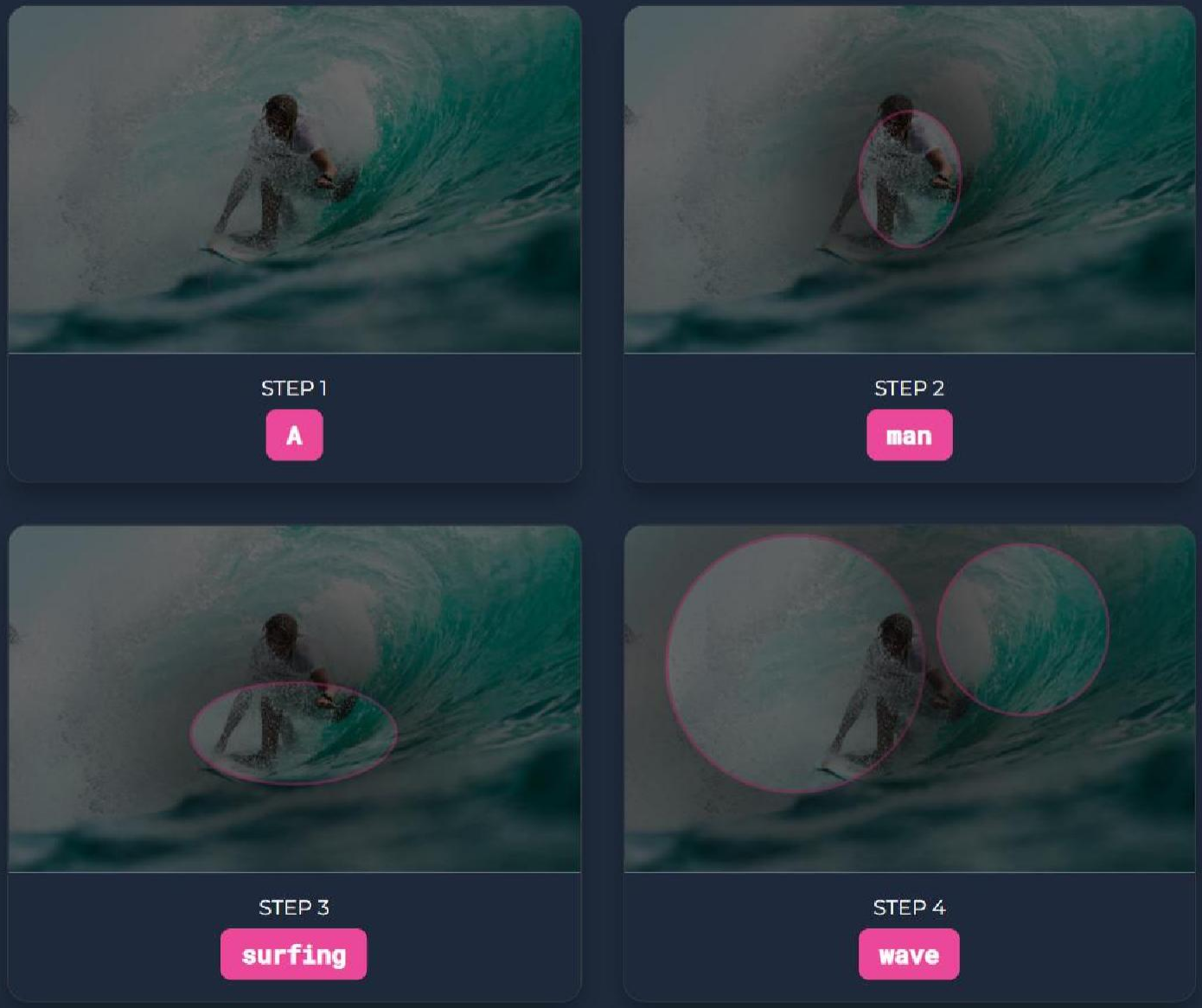
Computes a weighted sum of image regions (pixels) relevant to the current word being generated.

Interpretability

Visualizes exactly what the model is looking at for each word, making the "black box" transparent.

```
context_vector = sum(weights * features)
weights = softmax(score(decoder_state, features))
```

Visualization of Attention Weights



Brighter areas indicate higher attention weight (Soft Attention)

MODEL TRAINING

Configuration & Strategy

Step 4/5: Optimization

Hyperparameters

Optimizer	Adam
Loss Function	Categorical Cross-Entropy
Batch Size	64
Epochs	20
Learning Rate	0.001

model_train.py UTF-8

```python
# Compile the model
optimizer = Adam(learning_rate=0.001)
model.compile(
  optimizer=optimizer,
  loss='categorical_crossentropy',
  metrics=['accuracy']
)

# Fit with Teacher Forcing generator
history = model.fit(
  train_generator, epochs=20,
  steps_per_epoch=steps,
  verbose=1
)
```

PERFORMANCE ANALYSIS

Evaluation Metrics

Assessing quality through quantitative scores and qualitative human review.

Automatic Metrics

CALCULATED VS REFERENCE

BLEU Score (Bilingual Evaluation Understudy)

B-4

Measures n-gram precision overlap between generated caption and reference. BLEU-4 is the standard benchmark for fluency.

METEOR

M

Considers synonym matching and stemming. Often correlates better with human judgment than BLEU for semantic correctness.

CIDEr (Consensus-based Image Description)

C

Designed specifically for image captioning. Weights n-grams based on TF-IDF across the dataset to capture distinctiveness.

Human Evaluation

QUALITATIVE REVIEW

Semantic Correctness

Does the caption accurately describe the main objects, actions, and scenes present in the image?

Grammatical Fluency

Is the generated sentence grammatically correct and natural-sounding in the target language?

Why Manual Checking?

Automatic metrics can be misleading. A caption like "A dog on grass" vs "A puppy in the field" might have low n-gram overlap (low BLEU) but perfect semantic match. Human review ensures the model actually "understands" the image.

SYSTEM FEATURES

Implementation Overview

Key functionalities and real-time execution pipeline of the application.

Multi-Modal Input

Supports real-time webcam capture for live captioning and static image upload for existing files.

Smart Inference

Generates context-aware captions using the trained Encoder-Decoder model with Attention.

Accessibility Output

Converts generated text to speech instantly, aiding visually impaired users.

Runtime Execution Flow

1. Data Acquisition
Capture frame or load file
`OpenCV / PIL`

2. Preprocessing
Resize (299x299), Normalize
`tf.image`

3. Model Inference
CNN Features → LSTM Generation
`TensorFlow`

4. Output Delivery
Display Caption + Speak Audio
`pyttsx3`

MODEL ARCHITECTURE BREAKDOWN

Pretrained vs. Trained Components

Pretrained Components

Frozen Parameters • Transfer Learning

InceptionV3 Backbone

FROZEN

The CNN model pretrained on ImageNet (1000 classes). Used solely as a static feature extractor to encode visual content into vectors.

Image Processor

STATIC

Standard preprocessing pipeline: Resizing images to 299x299, pixel normalization (-1 to 1), and channel ordering.

Tokenizer (Vocab)

FIXED

The mapping from words to integers is built once from the training corpus and remains fixed during model training.

VS

Trained Components

Active Learning • Custom Parameters

LSTM Decoder

TRAINABLE

The recurrent neural network that learns to generate sequence tokens based on the current state and visual context.

Attention Layers

TRAINABLE

Dense layers W1, W2, V that learn to calculate the relevance score (alignment) between image features and hidden states.

Embedding Layer

TRAINABLE

Learns dense vector representations for the vocabulary words specific to the COCO dataset context.

DEVELOPMENT ECOSYSTEM

Technology Stack

Core · Vision · NLP · App

Frameworks

Python

v3.8+

Primary programming language used for model development and backend logic.

TensorFlow / Keras

v2.x

Deep learning library for building, training, and deploying the CNN-LSTM model.

Computer Vision

OpenCV

cv2

Used for real-time webcam video capture and image frame processing.

InceptionV3

Pretrained

State-of-the-art CNN architecture used for extracting high-level visual features.

NLP & Data

MS COCO Dataset

2014/17

Large-scale dataset for object detection, segmentation, and captioning training.

LSTM + Attention

RNN

Recurrent neural network with Bahdanau attention for sequence generation.

Application

Speech Libs

I/O

`SpeechRecognition` for input and `pyttsx3` for text-to-speech output.

Dev Tools

Dev

Jupyter Notebooks for training experiments; Git for version control.

PROJECT ROADMAP

Team Execution Plan

PHASE 1

Data Preparation

Cleaning MS COCO dataset, resizing images, and tokenizing captions.

PHASE 2

Feature Extraction

Implementing InceptionV3 to extract feature vectors and storing as pickle files.

PHASE 3

Model Building

Coding LSTM Decoder and Attention layers. Integration with CNN features.

PHASE 4

Training & Evaluation

Hyperparameter tuning, training loop execution, and BLEU score validation.

PHASE 5

App Integration

Building the UI, connecting webcam input, and final system testing.

Module-wise Development

Independent development of CNN and RNN modules before integration.

Agile Sprints

Weekly code reviews and incremental updates to the codebase.

Continuous Testing

Unit testing for data loaders and model layers at each stage.

ORGANIZATION STRUCTURE

Team Distribution

Functional Roles & Resource Allocation

Team Lead / Integrator

Overall coordination, architecture design, and system integration.

Data Prep

Cleaning, resizing, and tokenization.

2 Members

Feature Extraction

InceptionV3 implementation.

1-2 Members

Captioning

LSTM decoder and language modeling.

2 Members

Integration

Attention mechanism & model fusion.

1-2 Members

Training

Model training, tuning, and evaluation.

2 Members

MODEL INFERENCE

Results & Output Examples

Test Set Evaluation

> a man riding a wave on top of a surfboard

CONFIDENCE: 98.4%

145MS

> two dogs running through a grassy field

BLEU: 0.65

> a pizza with cheese and toppings on a table

BLEU: 0.72

0.58	0.32
BLEU-1 SCORE	BLEU-4 SCORE

Key Observations

Context Awareness

Model successfully identifies relationships (e.g., "riding", "on top of") rather than just listing objects.

Attention Impact

Visual attention mechanism significantly improves descriptive accuracy in cluttered scenes.

Real-Time Performance

Inference speed is sufficient for live video captioning applications on standard GPUs.

PROJECT SUMMARY

Conclusion & Learning Outcomes

We successfully bridged the gap between Computer Vision and Natural Language Processing to create an intelligent, real-time image captioning system.

End-to-End Integration

Successfully combined a powerful CNN (InceptionV3) for visual encoding with an LSTM decoder for language modeling, creating a seamless pipeline from pixels to semantics.

Attention Mechanism

Implemented a visual attention mechanism that allows the model to "focus" on relevant image regions dynamically, significantly improving caption accuracy and interpretability.

Real-Time Application

Moved beyond theory to build a functional prototype with webcam support and Text-to-Speech integration, demonstrating real-world accessibility potential.

- Successful MS COCO Training
- Real-time Inference < 200ms
- Deployed Interactive GUI

STRATEGIC OUTLOOK

Future Enhancements

Roadmap to improving accuracy, performance, and accessibility.

OPTIMIZATION

Immediate Steps

Q3 2024

Beam Search Decoding

DESIGN

Implement Beam Search (k=3, 5) to replace greedy search, exploring multiple caption paths for better fluency.

Model Quantization

PLAN

Convert weights to FP16 or INT8 using TensorFlow Lite to reduce model size for mobile deployment.

ARCHITECTURE

Core Upgrades

Q4 2024

Transformer Model

R&D

Replace LSTM decoder with a Transformer-based decoder (e.g., GPT-2 style) to better capture long-range dependencies.

Dense Captioning

PLAN

Generate multiple captions for different regions of the image instead of a single global description.

EXPANSION

Broad Impact

2025+

Multilingual Support

IDEA

Train on multi-language datasets (e.g., Multi30k) to generate captions directly in Spanish, French, and Hindi.

Web/Mobile App

IDEA

Deploy full-stack application with React Native frontend and Flask/FastAPI backend for public access.

Thank You!

Bridging Vision and Language with Deep Learning

Q&A SESSION OPEN

EVOASTRA VENTURES PVT. LTD.

Team D

TEAM LEAD
Rakshada Renapurkar

CO-LEAD
Subham Maharana