

# Tạo sinh chú thích ảnh với cơ chế chú ý

Trường Đại Học Khoa Học Tự Nhiên, thành phố Hồ Chí Minh, Việt Nam

21120127 Lê Hoàng Sơn

VNUHCM

HCMUS

tp Hồ Chí Minh, Việt Nam

21120127@student.hcmus.edu.vn

21120085 Võ Gia Khang

VNUHCM

HCMUS

tp Hồ Chí Minh, Việt Nam

21120085@student.hcmus.edu.vn

**Tóm tắt nội dung**—Tài liệu này mô tả xây dựng và huấn luyện mô hình phát sinh chú thích ảnh có chú ý. Mô hình này được xây dựng dựa trên kiến trúc Encoder, Decoder và Attention Mechanism trên tập dữ liệu Flickr8k. Bên cạnh đó, chúng ta sẽ thử nghiệm cải tiến mô hình.

**Index Terms**—Show, Tell, and Attend: Image Captioning with Attention Mechanism

## I. GIỚI THIỆU

Chú thích hình ảnh nhằm mục đích tự động tạo ra các câu mô tả cho hình ảnh, chụp các đối tượng và mối quan hệ của chúng. Nhiệm vụ đầy thách thức này kết nối Thị giác máy tính (CV) và Xử lý ngôn ngữ tự nhiên (NLP) trong Trí tuệ nhân tạo (AI). Những tiến bộ gần đây, chẳng hạn như các mô hình dựa trên sự chú ý, đã tạo ra những kết quả mạnh mẽ, cho phép các ứng dụng như công cụ trợ năng và gắn thẻ nội dung. Trong dự án này, chúng tôi triển khai một mô hình chú thích hình ảnh bằng cách sử dụng khuôn khổ CNN-RNN với sự chú ý trực quan, lấy cảm hứng từ 'Show, Attend and Tell' và tinh chỉnh nó trên tập dữ liệu Flickr8k để cải thiện chất lượng chú thích trong hơn 6 tuần.

Bài toán có thể được mô tả như sau: Cho ảnh đầu vào với 03 kênh màu RGB, mô hình sẽ tạo ra một câu mô tả cho ảnh đó bằng ngôn ngữ Tiếng Anh và ở hiện tại đơn. Ví dụ, ảnh đầu vào "chú chó với nền cỏ xanh" thì mô hình sẽ có thể tạo sinh ra chú thích mô tả ngắn gọn "A dog sits on green grass".

## II. CÔNG TRÌNH LIÊN QUAN

### A. Show, Attend and Tell

Show, Attend and Tell [1] giới thiệu cơ chế chú ý cho phép mô hình tập trung vào các vùng quan trọng của hình ảnh. Mô hình sử dụng kiến trúc CNN (encoder) và LSTM (decoder) với cơ chế chú ý trực quan, cho phép nó tự học cách tập trung vào các phần liên quan của hình ảnh khi tạo ra từng từ trong chú thích.

### B. A PyTorch Tutorial to Image Captioning

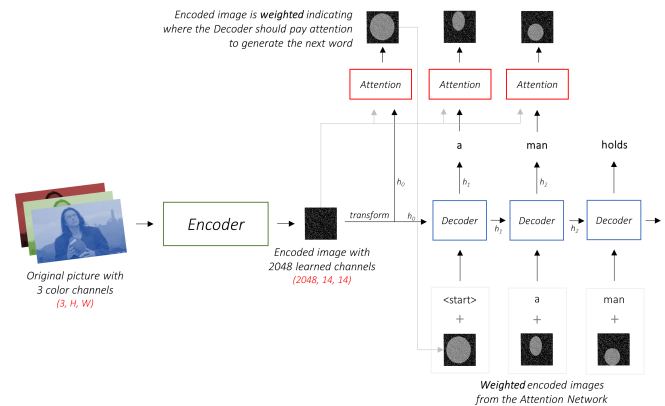
Một triển khai thân thiện với người mới bắt đầu cho mô hình chú thích hình ảnh bằng PyTorch [2] cung cấp cách thực hiện CNN + LSTM với cơ chế chú ý. Dự án này sử dụng tập dữ liệu Flickr8k/COCO và cung cấp mã nguồn mở, hướng dẫn chi tiết giúp hiểu rõ quy trình xây dựng mô hình chú thích hình ảnh.

### C. Show and Tell

Show and Tell [3] là mô hình sớm hơn sử dụng kiến trúc CNN + LSTM không có cơ chế chú ý. Đây được coi là mô hình cơ sở để so sánh hiệu quả của các mô hình sau này có tích hợp cơ chế chú ý. Mô hình này đã đặt nền tảng cho các phương pháp tiên tiến hơn trong lĩnh vực chú thích hình ảnh tự động.

## III. PHƯƠNG PHÁP

### A. Mô hình



Hình 1. Kiến trúc tổng thể của mô hình Image Captioning với cơ chế chú ý, bao gồm Encoder CNN (ResNet-101), cơ chế Attention và Decoder LSTM.

1) *Encoder*: Chúng tôi sử dụng mô hình ResNet-101 đã được tiền huấn luyện làm bộ mã hóa để trích xuất đặc trưng từ ảnh đầu vào. Cụ thể:

- Loại bỏ hai lớp cuối cùng của ResNet-101 (lớp fully-connected và lớp pooling) vì chúng tôi chỉ cần các đặc trưng chứ không cần phân loại ảnh
- Sử dụng Adaptive Average Pooling để chuẩn hóa kích thước đầu ra thành  $14 \times 14$
- Đầu ra có kích thước ( $batch\_size, 2048, 14, 14$ ) sau đó được chuyển thành ( $batch\_size, 14, 14, 2048$ ) để phù hợp với bộ giải mã
- Cho phép tinh chỉnh (fine-tune) các khối tích chập từ 5 trở đi khi cần thiết

2) *Attention*: Cơ chế chú ý được triển khai gồm các thành phần:

- Ba lớp tuyến tính để biến đổi:
  - Đặc trưng ảnh (*encoder\_att*)
  - Trạng thái ẩn của bộ giải mã (*decoder\_att*)
  - Kết hợp thông tin để tính trọng số chú ý (*full\_att*)
- Hàm kích hoạt ReLU và Softmax để tính trọng số chú ý
- Đầu vào: Đặc trưng ảnh (*encoder\_out*) và trạng thái ẩn hiện tại của bộ giải mã (*decoder\_hidden*)
- Đầu ra: Mã hóa có trọng số chú ý (*attention\_weighted\_encoding*) và ma trận trọng số chú ý ( $\alpha$ )

3) *Decoder*: Bộ giải mã sử dụng LSTM với cơ chế chú ý gồm các thành phần chính:

- Khởi tạo trạng thái ẩn ( $h, c$ ) từ đặc trưng ảnh trung bình
- Lớp embedding để chuyển từ ngữ cảnh thành vector
- Tế bào LSTM (*LSTMCell*) xử lý kết hợp:
  - Embedding của từ hiện tại
  - Mã hóa có trọng số chú ý
- Cổng điều chỉnh ( $f\_beta$  với *sigmoid*) để kiểm soát ảnh hưởng của cơ chế chú ý
- Lớp fully-connected cuối cùng để dự đoán từ tiếp theo
- Quy trình hoạt động:
  - Sắp xếp dữ liệu theo độ dài chú thích giảm dần
  - Với mỗi bước thời gian:
    - \* Tính trọng số chú ý và mã hóa có trọng số
    - \* Cập nhật trạng thái LSTM
    - \* Dự đoán từ tiếp theo
  - Chỉ xử lý các mẫu còn hoạt động (chưa kết thúc)

Các kỹ thuật hỗ trợ:

- Dropout để tránh overfitting
- Khởi tạo trọng số đồng đều trong khoảng  $[-0.1, 0.1]$
- Xử lý độ dài thay đổi của chú thích

## B. Hàm mất mát

Hàm mất mát được sử dụng trong mô hình này là hàm mất mát chéo entropy, giúp đo lường sự khác biệt giữa phân phối xác suất dự đoán và phân phối xác suất thực tế. Hàm mất mát này sẽ được tối ưu hóa trong quá trình huấn luyện để cải thiện độ chính xác của mô hình.

## C. Đánh giá BLEU

Đánh giá mô hình sẽ được thực hiện bằng cách sử dụng chỉ số BLEU, một phương pháp phổ biến để đo lường chất lượng của các mô tả được tạo ra so với các mô tả tham chiếu. Chỉ số BLEU sẽ giúp chúng tôi đánh giá độ chính xác và tính tự nhiên của các câu mô tả.

# IV. KẾ HOẠCH

## A. Tổng quan

Dự án này được thực hiện trong khoảng thời gian 6 tuần, trong đó chúng tôi lập kế hoạch triển khai và cải tiến mô hình chú thích hình ảnh dựa trên kiến trúc Encoder-Attention-Decoder.

## B. Kiến trúc đề xuất

Mô hình của chúng tôi sẽ sử dụng kiến trúc tương tự như "Show, Attend and Tell":

- **Encoder**: Sử dụng Convolutional Neural Network (ResNet-101) để trích xuất đặc trưng từ ảnh, tạo ra L vector đại diện cho các phần khác nhau của ảnh, mỗi vector có chiều D.
- **Attention**: Cài đặt cơ chế chú ý bằng các lớp tuyến tính đơn giản để mô hình học cách tập trung vào các phần quan trọng của ảnh khi tạo từng từ.
- **Decoder**: Sử dụng Long Short-Term Memory (LSTM) để tạo ra câu mô tả từ các đặc trưng đã được trích xuất và thông tin chú ý.

## C. Tiến độ thực hiện

- **Tuần 1-2**: Thu thập và tiền xử lý dữ liệu Flickr8k, xây dựng pipeline dữ liệu
- **Tuần 3-4**: Cài đặt mô hình cơ bản (Encoder-Decoder) không có cơ chế chú ý
- **Tuần 4-5**: Tích hợp cơ chế chú ý và tối ưu hóa mô hình
- **Tuần 5-6**: Đánh giá mô hình sử dụng BLEU score và thực hiện cải tiến

## D. Cải tiến dự kiến

- Thử nghiệm với các mạng CNN khác nhau cho Encoder (ResNet, Inception-v3)
- Tích hợp beam search để cải thiện chất lượng mô tả được tạo ra
- Fine-tuning encoder để nâng cao hiệu suất của mô hình
- So sánh hiệu quả giữa mô hình có cơ chế chú ý và không có cơ chế chú ý

# V. HUẤN LUYỆN VÀ PHÂN TÍCH MÔ HÌNH

## A. Mô tả bộ dữ liệu

Dữ liệu được sử dụng trong mô hình này là tập dữ liệu Flickr8k, bao gồm 8,000 hình ảnh và các mô tả tương ứng. Tập dữ liệu này được chia thành các phần huấn luyện, kiểm tra và xác thực để đảm bảo mô hình có thể học và đánh giá hiệu quả. Tuy nhiên, ở đây chúng tôi sẽ sử dụng bộ mô tả ảnh của Andrej Karpathy [2] với 8,000 hình ảnh và 40,000 mô tả. Mỗi hình ảnh có 5 mô tả khác nhau, giúp mô hình học được nhiều cách diễn đạt khác nhau cho cùng một nội dung.

Để chuẩn bị dữ liệu cho mô hình, chúng tôi đã thực hiện các bước sau:

- Xây dựng Worldmap, ánh xạ các từ trong mô tả thành các chỉ số số nguyên.
- Tổ chức dữ liệu thành tập huấn luyện và kiểm tra dưới định dạng hdf5 để dễ dàng truy cập và xử lý.
- Thêm từ đặc biệt như <start> và <end> để đánh dấu bắt đầu và kết thúc của một mô tả.

## B. Các siêu tham số

- Learning rate: 0.001
- Batch size: 32
- Number of epochs: Do bị giới hạn về thời gian và không thể chạy trên GPU, chúng tôi sẽ dừng lại ở epoch 5.
- Optimizer: Adam

## VI. KẾT LUẬN

### TÀI LIỆU

- [1] K. Xu et al., "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention,"arXiv:1502.03044, 2015.
- [2] S. G. Vinod, "A PyTorch Tutorial to Image Captioning,"GitHub repository, 2018.
- [3] O. Vinyals et al., "Show and Tell: A Neural Image Caption Generator,"arXiv:1411.4555, 2014.