Nguyễn Hoàng Long

uit  20520239@gm.uit.edu.vn

**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**ĐỀ TÀI: IMAGE CAPTIONING AND APPLICATION**

**LỜI CẢM ƠN**

Trong suốt quá trình học tập và tìm hiểu **đồ án môn học Các kĩ thuật học sâu và Ứng dụng*,*** em đã nhận được sự quan tâm, hỗ trợ rất tận tình của thầy. Thông qua việc báo cáo đồ án môn học, chúng em xin bày tỏ lòng biết ơn đến thầy **Nguyễn Vĩnh Tiệp** – giảng viên hướng dẫn môn học đã tận tình hướng dẫn, giải đáp thắc mắc, định hướng và phát triển việc xây dựng, phát triển đồ án môn học.

Nhóm em rất mong sẽ nhận được những đóng góp quý báu từ phía thầy nhằm hoàn thiện hơn về đồ án môn học này.

Trân trọng.

*TP. Hồ Chí Minh, ngày 01 tháng 7 năm 2023*

**NHÓM THỰC HIỆN**

**Hoàng Long**

**MỤC LỤC**

[1. Giới thiệu bài toán 3](#_Toc139236198)

[1.1. Image Captioning là gì? 3](#_Toc139236199)

[1.2. Các ứng dụng của Image Captioning? 3](#_Toc139236200)

[2. BỘ DỮ LIỆU 4](#_Toc139236201)

[2.1. Bộ dữ liệu sử dụng 4](#_Toc139236202)

[3. THUẬT TOÁN 4](#_Toc139236203)

[3.1. Phân tích bài toán 4](#_Toc139236204)

[3.2. Image Captioning hoạt động như thế nào? 5](#_Toc139236205)

[3.2.1. Bộ mã hóa đặc trưng của ảnh (Image Feature Encoder) 5](#_Toc139236206)

[3.2.2. Bộ giải mã trình tự (Sequence Decoder) 7](#_Toc139236207)

[3.2.3. Trình tạo câu (Sentence Generator) 7](#_Toc139236208)

[3.3. Kiến trúc sử dụng 8](#_Toc139236209)

[3.3.1. Kiến trúc: Multi-Modal 8](#_Toc139236210)

[3.3.2. Kiến trúc: Encoder-Decoder với Attention 9](#_Toc139236211)

[3.4. GoogleNet - Inception-V3 (2015) 11](#_Toc139236212)

[3.5. Các bước chi tiết 12](#_Toc139236213)

[3.5.1. Text preprocessing 12](#_Toc139236214)

[3.5.2. Word embedding 12](#_Toc139236215)

[3.5.3. Output 13](#_Toc139236216)

[4. ĐÁNH GIÁ 13](#_Toc139236217)

[4.1. Bleu Score 13](#_Toc139236218)

[4.2. Kết quả 14](#_Toc139236219)

[5. HƯỚNG PHÁT TRIỂN 14](#_Toc139236220)

# Giới thiệu bài toán

## Image Captioning là gì?

**Image Captioning (Chú thích hình ảnh)** là quá trình tạo mô tả văn bản của một hình ảnh. Sử dụng cả Xử lý ngôn ngữ tự nhiên và Thị giác máy tính để tạo phụ đề cho ảnh.

Ảnh có chứa văn bản, trang phục, áp phích, Mặt người

Mô tả được tạo tự động

Tập dữ liệu sẽ ở dạng **[image → captions]**. Bộ dữ liệu bao gồm các hình ảnh đầu vào và chú thích đầu ra tương ứng của chúng.

## Các ứng dụng của Image Captioning?

Ta có thể thấy ngay 2 ứng dụng của **Image captioning**:

* Để giúp những người già mắt kém hoặc người mù có thể biết được cảnh vật xung quanh hay hỗ trợ việc di chuyển. Quy trình sẽ là: **Image -> text -> voice.**
* Ảnh có chứa văn bản, ảnh chụp màn hình, Website, Trang web

  Mô tả được tạo tự độngẢnh có chứa trang phục, người, giày dép, đồ bò

  Mô tả được tạo tự độngGiúp google search có thể **tìm kiếm** được hình ảnh dựa vào caption

# bộ dữ liệu

## Bộ dữ liệu sử dụng

Dữ liệu dùng trong bài toán là [***Flickr8k Dataset***](https://academictorrents.com/details/9dea07ba660a722ae1008c4c8afdd303b6f6e53b)**.** Dữ liệu gồm **8000** ảnh, **6000** ảnh cho traning set, **1000** cho dev set (validation set) và 1000 ảnh cho test set.

Ảnh có chứa ảnh chụp màn hình, ảnh ghép tổng hợp, tranh ghép kính

Mô tả được tạo tự động

Với mỗi ảnh sẽ có 5 captions, như vậy traning set sẽ có 6000 \* 5 = **40000** datasets.

# THUẬT TOÁN

## Phân tích bài toán

**Input** là ảnh và **output** là text, ví dụ ***“man in black shirt is playing guitar”.***

Nhìn chung các mô hình machine learning hay deep learning đều **không xử lý trực tiếp** với text như ‘man’, ‘in’, ‘black’,… mà thường phải **quy đổi (encode)** về dạng số. Từng từ sẽ được **encode** sang dạng **vector** với độ dài cố định, gọi là **word embedding.**

**Input** là ảnh thường được extract feature qua pre-trained model với dataset lớn như ImageNet và model phổ biến như VGG16, ResNet, quá trình được gọi là embedding và output là 1 vector.

Nhìn thấy **output** là text nghĩ ngay đến **RNN** và sử dụng mô hình **LSTM**.

🡺 **Ý tưởng** sẽ là dùng embedding của ảnh và dùng các từ phía trước để dự đoán từ tiếp theo trong caption.

**Ví dụ:**

Embedding vector + A -> girl

Embedding vector + A girl -> going

Embedding vector + A girl going -> into

Embedding vector + A girl going into -> a.

Embedding vector + A girl going into a -> wooden building .

Embedding vector + A girl going into a wooden -> building .

Ảnh có chứa ảnh chụp màn hình, Phông chữ, văn bản, Đồ họa

Mô tả được tạo tự động

Để dự đoán từ tiếp theo ta sẽ xây dựng từ điển các từ xuất hiện trong training set (ví dụ 2000 từ) và bài toán trở thành bài toán phân loại từ, xem từ tiếp theo là từ nào, khá giống như bài phân loại ảnh.

## Image Captioning hoạt động như thế nào?

Image Captioning sử dụng ba thành phần chính, bao gồm:

### Bộ mã hóa đặc trưng của ảnh (Image Feature Encoder)

Thao tác này lấy ảnh nguồn làm đầu vào và biểu diễn dưới dạng mã hóa các đặc điểm thiết yếu của bức ảnh đó.

Ảnh có chứa văn bản, động vật có vú, ảnh chụp màn hình

Mô tả được tạo tự động

Tại đây, kiến trúc CNN và phương pháp transfer learning (học chuyển giao) thường được sử dụng phổ biến. Có thể kể đến một số mô hình được đào tạo từ trước, phục vụ việc phân loại ảnh như VGGNet, ResNet và Inception.

“Xương sống” của mô hình này là các khối CNN, từng bước trích xuất các đặc điểm khác nhau từ ảnh và tạo ra một tóm tắt nhỏ gọn để nắm bắt các yếu tố quan trọng nhất trong bức ảnh.

Ví dụ. Nó bắt đầu bằng cách trích xuất các dạng hình học đơn giản như đường cong và bán nguyệt trong các lớp ban đầu, tiến dần đến các cấu trúc cấp cao hơn như mũi, mắt và bàn tay, và cuối cùng xác định các yếu tố như khuôn mặt và bánh xe.

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Phông chữ

Mô tả được tạo tự động

Trong mô hình Phân loại hình ảnh, bản tóm tắt đặc trưng (còn được gọi là feature map) sau đó được đưa đến Trình phân loại, nhằm dự đoán đầu ra cuối cùng của lớp (ví dụ: mèo hoặc ô tô) của đối tượng chính trong hình ảnh.

Khi áp dụng mô hình này cho chú thích ảnh, ta quan tâm đến việc biểu diễn bản tóm tắt đặc trưng của ảnh và không cần dự đoán phân loại. Vì vậy, các nhà phát triển thường giữ lại “xương sống” (khối CNN) và loại bỏ các lớp phân loại.

### Bộ giải mã trình tự (Sequence Decoder)

Thao tác này lấy biểu diễn được mã hóa của ảnh và xuất ra một chuỗi các mã tokens mô tả bức ảnh đó.

Thông thường, đây là mô hình Recurrent Network bao gồm các lớp LSTM (Long-short term memory) được cung cấp bởi một lớp nhúng (Embedding).

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Phông chữ

Mô tả được tạo tự động

Đầu vào của mô hình là vectơ mã hóa hình ảnh cùng mã token duy nhất: ‘Start’. Từ đây, mô hình ‘giải mã’ vector hình ảnh đầu vào và xuất ra một chuỗi các mã tokens.

Quy trình này xảy ra theo dạng vòng lặp, mã token được xuất ra sau đó được đưa trở lại mạng làm đầu vào cho lần lặp tiếp theo. Cuối cùng, mô hình trả ra mã token ‘End’ hoàn thành chuỗi.

### Trình tạo câu (Sentence Generator)

Dựa vào chuỗi token trước đó, trình tạo câu sẽ xuất ra phụ đề mô tả bức ảnh theo ngôn ngữ yêu cầu.

Nó bao gồm một lớp Linear theo sau là một Softmax. Điều này dẫn đến xác suất cho mọi vựng tồn tại trong ngôn ngữ đích và cho mỗi vị trí trong chuỗi.

Xác suất này là khả năng một từ xuất hiện ở một vị trí trong câu. Sau đó, ta có thể sử dụng Greedy Search để tạo ra câu cuối cùng bằng cách chọn từ có xác suất cao nhất ở mỗi vị trí.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Câu sau đó được xuất ra dưới dạng chú thích dự đoán.

Hầu hết tất cả các kiến trúc Image Captioning đều sử dụng ba thành phần nêu trên. Tuy nhiên, qua nhiều năm, khuôn khổ này cũng trải qua nhiều biến chuyển và bước phát triển nhất định.

## Kiến trúc sử dụng

### Kiến trúc: Multi-Modal

Kiến trúc Inject là kiến trúc ban đầu cho Image Captioning và vẫn rất phổ biến. Tuy nhiên, một giải pháp thay thế được gọi là kiến trúc “Multi-Modal” đã được tìm thấy để tạo ra kết quả tốt hơn.

Thay vì kết nối Bộ mã hóa hình ảnh làm đầu vào của Bộ giải mã trình tự, hai thành phần này hoạt động độc lập với nhau. Nói cách khác, ta không kết hợp hai dạng thức: hình ảnh với văn bản. Mạng CNN chỉ xử lý hình ảnh và mạng LSTM chỉ hoạt động trên chuỗi token được tạo ra.

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ

Mô tả được tạo tự động

Đầu ra của hai mạng này sau đó được kết hợp với nhau bằng một lớp Đa phương thức (có thể là một lớp Linear và Softmax). Nó thực hiện công việc diễn giải cả hai kết quả đầu ra và sau đó, Trình tạo câu sẽ dự đoán chú thích cuối cùng.

Một ưu điểm khác của cách tiếp cận này là nó cho phép ta sử dụng transfer learning không chỉ cho Bộ mã hóa hình ảnh mà còn cho Bộ giải mã trình tự. Chúng ta có thể sử dụng mô hình ngôn ngữ được đào tạo trước cho Bộ giải mã trình tự.

### Kiến trúc: Encoder-Decoder với Attention

Trong vài năm qua, việc sử dụng mô hình Attention với NLP đã thu hút được nhiều sự chú ý. Nó cho thấy khả năng cải thiện đáng kể hiệu suất của các ứng dụng NLP. Khi mô hình tạo ra từng từ của đầu ra, Attention giúp mô hình tập trung vào các từ trong chuỗi đầu vào có liên quan nhất đến từ đầu ra đó.

Do đó, không có gì ngạc nhiên khi thấy rằng Attention cũng đã được áp dụng cho Chú thích ảnh.

Khi Bộ giải mã trình tự tạo ra từng từ trong chú thích, Attention được sử dụng để giúp nó tập trung vào phần hình ảnh có liên quan nhất đến từ mà nó đang tạo.

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ

Mô tả được tạo tự động

Mô-đun Attention lấy vectơ hình ảnh được mã hóa cùng với mã token đầu ra hiện tại từ LSTM. Nó tạo ra Attention Score có trọng số. Khi điểm đó được kết hợp với hình ảnh, nó sẽ tăng trọng số của các pixel mà LSTM nên tập trung vào trong khi dự đoán mã token tiếp theo.

## GoogleNet - Inception-V3 (2015)

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, Kế hoạch

Mô tả được tạo tự động

Inception-V3 là kế thừa của Inception-V1 bao gồm 24 triệu tham số. Toàn bộ các layer tích chập của Inception-V3 được theo sau bởi một layer batch normalization và một ReLU activation. Batch normalization là kỹ thuật chuẩn hóa đầu vào theo từng minibatch tại mỗi layer theo phân phối chuẩn hóa **N(0,1)**, giúp cho quá trình huấn luyện thuật toán nhanh hơn.

Inception-V3 giải quyết được vấn đề thắt cổ chai (representational bottlenecks). Tức là kích thước của các layers không bị giảm một cách đột ngột. Đồng thời Inception-V3 có một cách tính toán hiệu quả hơn nhờ sử dụng phương pháp nhân tố (factorisation methods).

Hiện tại Inception module bao gồm 4 version. Chúng ta hãy cùng xem qua các điểm đặc biệt ở từng version.

* Inception-A: Cải tiến so với Inception module V1. Tại nhãnh thứ nhất thay 1 layer tích chập 5 x 5 bằng 2 layer tích chập 3 x 3 liên tiếp giúp giảm số lượng tham số từ 25 về 18 và tăng độ sâu cho mô hình.
* Inception-B: Cải tiến so với Inception-A. Thay tích chập 3 x 3 bằng tích chập 7 x 7 ở nhánh thứ nhất và nhánh thứ 2. Đồng thời chúng ta phân tích nhân tố tích chập 7 x 7 thành 2 tích chập liên tiếp 7 x 1 và 1 x 7 số lượng tham số sẽ ít hơn so với tích chập 2 tích chập 3 x 3 liên tiếp. Nhờ đó số lượng tham số giảm từ 18 xuống còn 14.
* Inception-C: Cải tiến so với Inception-B. Thay tích chập 7 x 1 bằng tích chập 3 x 1 và 1 x 7 bằng 1 x 3 và đồng thời thay vì đặt layer 3 x 1 và 1 x 3 liên tiếp thì đặt chúng song song. Kiến trúc này giúp giảm số lượng tham số từ 14 về còn 6.

Ngoài ra ở Inception-V3 chúng ta còn sử dụng 2 kiến trúc giảm chiều dữ liệu là Reduction-A và Reduction-B.

## Các bước chi tiết

Ta sẽ sử dụng pre-trained model Inception-V3 với dataset Imagenet. Do là pre-trained model yêu cầu ảnh đầu vào là 229\*229 nên ra sẽ resize ảnh về kích thước này. Sau khi qua pre-trained model ta sẽ lấy được embedding vector của ảnh, kích thước 256\*1

### Text preprocessing

Ta xử lý text qua một số bước cơ bản.

* Chuyển chữ hoa thành chữ thường, “Hello” -> “hello”
* Bỏ các kí tự đặc biệt như “%”, “$”, “#”
* Loại bỏ các chữ có số như hey199

Sau đó ta sẽ thêm 2 từ “startseq” và “endseq” để biểu thị sự bắt đầu và kết thúc của caption. Ví dụ: “startseq a girl going into a wooden building endseq“. “endseq” dùng khi test ảnh thì biết kết thúc của caption.

Ta thấy có 8763 chữ khác nhau trong số 40000 caption. Tuy nhiên ta không quan tâm lắm những từ mà chỉ xuất hiện 1 vài lần, vì nó giống như là nhiễu vậy và không tốt cho việc học và dự đoán từ của model, nên ta chỉ giữ lại những từ mà xuất hiện trên 10 lần trong số tất cả các caption. Sau khi bỏ những từ xuất hiện ít hơn 10 lần ta còn 1651 từ.

Tuy nhiên do độ dài các sequence khác nhau, ví dụ: “A”, ” A girl going”, ” A girl going into a wooden”, nên ta cần padding thêm để các chuỗi có cùng độ dài bằng với độ dài của chuỗi dài nhất là 34. Do đó số tổng số từ (từ điển) ta có là 1651 + 1 (từ dùng để padding).

### Word embedding

Pre-trained [GLOVE Model](https://www.kaggle.com/datasets/anindya2906/glove6b) được sử dụng cho quá trình word embedding.

Từng dòng trong file sẽ lưu text và encoded vector khích thước 200 \* 1.

### Output

Bài toán là dự đoán từ tiếp theo trong chuỗi ở input với ảnh hiện tại, nên output là từ nào trong số 1652 từ trong từ điển mà ta có. Với bài toán phân loại thì softmax activation và categorical\_crossentropy loss function được sử dụng.

# ĐÁNH GIÁ

## Bleu Score

Trong quá trình đào tạo, sau khi chú thích được tạo, làm sao để quyết định xem chú thích đó tốt như thế nào? Một chỉ số phổ biến để đánh giá các mô hình Chú thích ảnh là Bleu score. Bleu score cũng được dùng nhiều cho các ứng dụng NLP khác như dịch thuật và các mô hình ngôn ngữ.

Đây là một chỉ số đơn giản, đo lường số lượng các từ liên tiếp khớp giữa chú thích được dự đoán và chú thích đúng. Nó so sánh n-gam độ dài khác nhau từ 1 đến 4 để thực hiện điều này.

Ví dụ. Chú thích dự đoán: “Một con chó đứng trên bãi cỏ xanh”. Chú thích đúng là: “Con chó đang đứng trên bãi cỏ”

Bleu score cho 1 gam = Các từ được dự đoán chính xác / Tổng số từ được dự đoán

Có ba từ được dự đoán cũng xuất hiện trong chú thích đúng nghĩa là. “Dog”, “on” và “grass”, trong tổng số sáu từ được dự đoán.

Điểm Bleu cho 1 gam (tức là các từ đơn) = 3/6 = 0,5



## Ảnh có chứa con chó, văn bản, thú cưng, Giống chó Mô tả được tạo tự độngKết quả

Ảnh có chứa Phương tiện đường bộ, bánh xe, ngoài trời, xe đạp

Mô tả được tạo tự động

# HƯỚNG PHÁT TRIỂN

* Xây dựng thêm Web và App để hỗ trợ cho người dùng bị khiếm thị
* Huấn luyện trên nhiều bộ dữ liệu hơn
* Fine-tuning các mô hình end-to-end như LAVIS để xây dựng ứng dụng