VINBIGDATA

BÁO CÁO ĐỀ TÀI MÔN XỬ LÝ NGÔN NGỮ TỰ NHIÊN

IMAGE CAPTIONING IN VIETNAMESE LANGUAGE

Học viên
Nguyễn Minh Dũng
Lê Công Pha
Nguyễn Duy Nhất
Võ Minh Tâm

Giảng viên hướng dẫn
TS. Phan Việt Anh

Ngày 1 tháng 3 năm 2021

Mục lục

Danh sách hình vẽ

Danh sách bảng

Danh mục từ viết tắt

CNN	**C**onvolutional **N**eural **N**etwork
SVM	**S**upport **V**ector **M**achine
UIT-ViIC	A Dataset for the First Evaluation on **Vi**etnamese **I**mage **C**aptioning
BLEU	**Bi**L**ingual **E**valuation **U**nderstudy

Tóm tắt nội dung

Phát triển một hệ thống máy tính có thể hiểu được thế giới thị giác và giao tiếp với chúng ta bằng ngôn ngữ là một trong các đích đến lớn của trí tuệ nhân tạo. Để hiện thực hóa giấc mơ này, vô số bài toán đã được đặt ra, trong đó có Image Captioning. Bài toán này nhận đầu vào là một hình ảnh và cố gắng sinh ra một câu mô tả bằng ngôn ngữ tự nhiên cho ảnh đó.

Hiện nay, bằng cách áp dụng khai thác đặc trưng ảnh qua CNN và sử dụng RNN để sinh câu mô tả, hướng tiếp cận CNN+LSTM đã mang lại đột phá mới trong bài toán Image Captioning. Tuy nhiên, phần lớn các nghiên cứu hiện tại chủ yếu tạo chú thích bằng tiếng Anh hoặc tiếng Trung cho ảnh. Trong đồ án này, chúng tôi tập trung giải quyết bài toán Image Captioning cho tiếng Việt – ngôn ngữ đang có gần 100 triệu người sử dụng. Chúng tôi sẽ kế thừa bộ dữ liệu UIT-ViIC - bộ dữ liệu đầu tiên cho bài toán Image Captioning cho tiếng Việt. Chúng tôi hy vọng kết quả đạt được sẽ tạo động lực cho các nghiên cứu sâu hơn về lĩnh vực Image Captioning trên tiếng Việt cũng như đa ngôn ngữ.

Chương 1

Tổng quan

1.1 Đặt vấn đề

Tự động nhận biết và tạo tiêu đề cho ảnh là bài toán đầy tính thách thức. Phần lớn các bộ dữ liệu và nghiên cứu hiện tại được thực hiện cho tiếng Anh trong khi các ngôn ngữ như tiếng Việt thì chưa được phổ biến. Cách tiếp cận đơn giản nhất là sử dụng các công cụ dịch máy như Google translation để dịch các câu tiêu đề từ tập dữ liệu nguồn sang ngôn ngữ đích. Tuy nhiên, các công cụ dịch máy này đôi khi không mang lại hiệu quả do ngôn ngữ có tính nhập nhằng rất cao dẫn đến kết quả dịch không được tự nhiên như trong ngôn ngữ của người bản xứ.

Qua khảo sát, nhóm chúng tôi tìm được bộ dữ liệu UIT-ViIC của nhóm tác giả [1] thực hiện tạo tiêu đề cho 3850 ảnh (phiên bản 2017) thuộc lĩnh vực thể thao với bóng từ tập dữ liệu nổi tiếng MS COCO (hơn 150,000 ảnh, 5 captions mỗi ảnh, challenge 2015). Trong đồ án này, nhóm chúng tôi sẽ cố gắng cải thiện độ chính xác từ baseline ban đầu của nhóm tác giả bằng các mô hình mới nhất.

1.2 Giới thiệu bài toán

Input: Một bức ảnh (chủ đề giới hạn trong đồ án này là ảnh thuộc lĩnh vực thể thao với bóng, chẳng hạn như bóng bầu dục, golf, bóng đá...).

Output: Một câu mô tả bằng tiếng Việt cho bức ảnh.

HÌNH 1.1: Minh họa đầu vào và đầu ra của bài toán.

Ứng dụng:

- Mô tả những hoạt động diễn ra trước mắt để hỗ trợ người khiếm thị (sau khi sinh văn bản thì chuyển thành tiếng nói).
- Giải thích những gì diễn ra trong một video để có thể ứng dụng trong bài toán tìm kiếm sự vật, sự việc trong video.
- Trong thương mại điện tử: tự động sinh tiêu đề, nội dung cho các trang bán hàng online.
- Tự động sinh tiêu đề, nội dung cho hình ảnh, video trên báo...

1.3 Mục tiêu đề tài

- Đọc hiểu những kỹ thuật được sử dụng trong bài toán Image Captioning, bộ dữ liệu UIT-VilC.
- Xây dựng mô hình sinh câu mô tả cho ảnh bằng một số kỹ thuật học sâu.
- Xây dựng được chương trình demo cho bài toán Image Captioning.

1.4 Phương pháp thực hiện

- Tìm hiểu bộ dữ liệu UIT-VilC được giới thiệu trong bài báo "A Dataset for the First Evaluation on Vietnamese Image Captioning".
- Tìm hiểu một số kiến trúc mạng encoder như EfficientNet, Resnet, Inception và decoder như LSTM có thể sử dụng để giải quyết bài toán Image Captioning.
- Huấn luyện mô hình trên tập dữ liệu UIT-VilC và đánh giá mô hình.
- Xây dựng chương trình demo Image Captioning cho dữ liệu tiếng Việt.

1.5 Nội dung thực hiện

- Xác định đầu vào, đầu ra và ứng dụng của bài toán.
- Tìm hiểu một số thông tin cơ bản về bộ dữ liệu UIT-VilC (kích thước ảnh, số lượng mẫu, số lượng nhãn ...).
- Tìm hiểu ý tưởng, cách hoạt động của một số mô hình Show and Tell, NIC.
- Xây dựng chương trình từ pipeline: Rút trích đặc trưng $\rightarrow$ Huấn luyện $\rightarrow$ Kiểm tra $\rightarrow$ So sánh, đánh giá.
- Xây dựng ứng dụng demo để trực quan hóa kết quả.
- Viết báo cáo đề tài.

Chương 2

Cơ sở lý thuyết

2.1 Embedding

2.1.1 Image Embedding

Ảnh đầu vào sau khi được rút trích đặc trưng sẽ được biểu diễn bằng một vector 2048 chiều. Tuy nhiên đầu vào của LSTM thường là một vector 512 chiều nên cần một layer image embedding với nhiệm vụ ánh xạ vector 2048 chiều này về 512 chiều thích hợp để đưa vào input LSTM.

2.1.2 Word Embedding

Ngoài truyền vector đặc trưng của ảnh ta cũng cần phải truyền các words trong caption vào LSTM. Như vậy, cần một phương pháp lượng tử hóa các words ở dạng ký tự thành các vector số học đại diện cho word đó, các phương pháp này được gọi là word embedding. Ý tưởng đơn giản nhất để làm việc này là sử dụng one-hot vector. One-hot vector là một vector có số chiều bằng số lượng words có trong vocabulary, và vector đại diện cho word nào sẽ có giá trị bằng 1 tại vị trí của word đó trong vocabulary, tất cả các vị trí còn lại đều bằng 0. Tuy nhiên, do số chiều của one-hot vector sẽ rất lớn khi vocabulary có nhiều words nên ta không thể truyền trực tiếp vector này vào LSTM. Hơn nữa, các vector này không thể hiện được tính chất liên kết ngữ nghĩa giữa các words. Sẽ là hay hơn nếu các word có nghĩa gần nhau sẽ có vector đại diện cũng tương đồng. Có thể hình dung như sau: các words được chiếu lên một không gian vector với đặc tính nếu 2 word càng gần nghĩa với nhau thì khoảng cách giữa chúng càng nhỏ.

Dựa trên ý tưởng ở trên, các phương pháp cũ như Latent Semantic Analysis (LSA) sẽ tạo ra ma trận gồm dòng là đại diện cho word và cột là các documents với giá trị là trọng số TF-IDF. Vấn đề với các phương pháp cũ là có thể tốn chi phí tính toán khổng lồ khi dữ liệu đầu vào lớn. Với sự xuất hiện của Neural Network, ta có thể thực hiện thiết kế một layer Word Embedding có thể học cách ánh xạ các onehot vector sang một không gian vector phù hợp hơn.

2.1.3 Word Segmentation

Tiếng Việt có những hạn chế riêng của nó có thể ảnh hưởng đến kết quả của việc xử lý ngôn ngữ tự nhiên. Một trong các hạn chế lớn nhất đó là sự nhập nhằng khi sử dụng khoảng trống (space character). Như chúng ta đã biết, khoảng trống trong tiếng Việt không được dùng để chia tách các từ khác nhau. Nghĩa là, một từ trong tiếng Việt có thể có hai hoặc nhiều âm tiết bị chia cách bởi các dấu cách. Ví dụ, “máy tính xách tay”

có 4 âm tiết cách nhau bởi các khoảng trống nhưng chỉ được tính là một từ trong tiếng Việt. Do đó, việc sử dụng Word Segmentation (còn gọi là Tokenization) để tiền xử lý các câu caption trước khi training sẽ giúp hô hình được huấn luyện hiệu quả hơn.

2.1.4 Tạo caption sử dụng LSTM

Trong model Show and Tell, Vinyal đã sử dụng một mạng RNN để sinh câu caption cho ảnh bằng cách truyền đặc trưng của ảnh vào ở step đầu tiên của RNN và lần lượt từ đó sinh ra từng word một miêu tả ảnh. Một trong các vấn đề với các mạng RNN thông thường là Long-Term Dependency, nghĩa là khi ta thực hiện predict tại step càng xa so với thông tin cần thiết thì trạng thái của RNN sẽ không còn nhớ quá rõ về thông tin cần thiết đó nữa. Như vậy khi thực hiện lan truyền ngược (back-propagation) để tối ưu trọng số cho mạng RNN sẽ gặp hiện tượng suy giảm độ dốc (vanishing gradient) không mong muốn. Vì trong NIC, thông tin về ảnh sẽ được truyền vào đầu tiên, nên ở các step RNN càng xa thì càng khó "nhớ" thông tin về ảnh đó là gì. Để giải quyết vấn đề này, Vinyal đề xuất sử dụng một mạng RNN có thể xử lý tốt Long-Term Dependency đó là Long Short-Term Memory (LSTM).

Một tế bào LSTM [9] gồm 3 cổng (gate) là:

• Input gate: quản lý thông tin nào sẽ được đưa vào cell.

• Output gate: quản lý thông tin nào trong cell sẽ được dùng để tính output của LSTM unit.

• Forget gate: quản lý thông tin nào trong cell sẽ bị quên đi (forget) do không liên quan đến vấn đề cần predict.

Tại mỗi gate là một activation function (dùng hàm sigmoid)

Tại mỗi step việc dự đoán được dựa trên thông tin lưu trong hidden state của LSTM, đó là đặc trưng của ảnh và các word đã trước đó đã dự đoán. Word dự đoán được ở step t là input cho step t+1. Cụ thể, quá trình lan truyền tiến (forward) Show and Tell có thể tóm gọn lại thành công thức như sau: Quá trình học của NIC được thực hiện bằng các tối ưu toàn bộ trọng số (trọng số của LSTM, word embedding layer, image embedding layer, fine-tune Inception) thông qua tối thiểu hóa hàm tính loss.

2.2 Độ đo

- **BLEU [4] (Bilingual Evaluation Understudy - precision-based)**

  Đây là độ đo thường được dùng trong dịch máy.

  Ý tưởng chính của BLEU là đếm số matching n-grams của candidate (câu được mô hình sinh ra) và reference (là câu ground truth) hoặc match trên bất kỳ reference nào nếu như có nhiều references. Kết quả sẽ là số match chia cho số từ của candidate. Các match này không phụ thuộc vào vị trí, do vậy BLEU không sử dụng word order. Càng match nhiều tức là càng tốt. Do đó, khi đếm matching n-grams cần chú ý cả số lần xuất hiện của từ trong reference, một từ trong reference khi được match rồi thì không nên match nữa để tránh hiện tượng một từ match với reference nhưng được lặp lại nhiều lần trong candidate.

  BLEU còn được dùng để đánh giá một corpus (tập hợp của các sentence, hay một đoạn văn). Đầu tiên là tính số match với từng câu. Cộng các số này rồi chia cho tổng số n-gram từ các câu là ra modified precision score cho test corpus.

$$\frac{\sum_{C \epsilon\{\text{ Candidates }\}} \sum_{n-\text{ gram } \in C} \text{ Count }_{\text{clip}}\left(n-\text{ gram }\right)}{\sum_{C' \in\{\text{ Candidates }\}} \sum_{n-\text{ gram }' \epsilon C'} \text{ Count } \left(n \text{ -gram }'\right)} \tag{2.1}$$

Ngoài ra, còn có thể sử dụng trọng số (weights) khác nhau cho các n-grams khác nhau để tính điểm BLEU cuối cùng. Dựa và trọng số này ta chia ra cumulative và individual BLEU với weights là 1 tuple thể hiện trọng số tương ứng với từng i-gram score ở vị trí thứ i. Thông thường, trong các bài báo nghiên cứu, để so sánh các kiến trúc khác nhau trên một benchmark dataset, cumulative BLEU-1, BLUE-2, BLEU-3, BLEU-4 được sử dụng.

Hạn chế: Có nhiều cách để dịch (tốt) một câu. Một bản dịch tốt có thể có điểm BLEU thấp vì nó có ít n-gram trùng với ground truth. Vì vậy, điểm BLEU phụ thuộc nhiều vào chất lượng của ground truth.

- **ROUGE Recall-Oriented Understudy for Gisting Evaluation (recall-based) [2]** Độ đo này được dùng trong tóm tắt văn bản, dựa trên n-grams. Công thức tính như sau:

$$ROUGE - n = \frac{p}{q}$$

Trong đó:

- p: số n-gram giống nhau giữa candidate và reference. - q: số lượng n-gram của reference.

ROUGE-L: Longest common subsequence (LCS) dựa trên việc xác định chuỗi con dài nhất.

- **CIDEr [8]**

Độ đo này được tạo ra dành riêng cho bài toán sinh câu mô tả cho ảnh, đánh giá độ đồng thuận giữa câu mô tả được sinh ra bởi mô hình cho ảnh thứ i (gọi là candidate sentence c_i) với tập mô tả ground truth tương ứng cho ảnh thứ i này $S_i = s_{ij}, ..., s_{im}$ (còn được gọi là các reference sentences).

Các bước thực hiện:

- B1: các word được đưa về dạng từ gốc, gọi là "stem" hay "root form". Ví dụ: "fishs", "fishing", "fished" được đưa về "fish".

- B2: Biểu diễn mỗi câu bằng một tập các n-grams (n thường từ 1 đến 4).

Ý tưởng: sự đo lường độ đồng thuận mã hóa số lượng n-grams trong câu candidate thường xuất hiện bao nhiêu lần trong các câu reference. Tương tự, những n-grams không nằm trong các reference sentences thì cũng không nên hiện diện trong cadidate sentence. Cuối cùng, các n-grams thường xuất hiện trong các câu mô tả cho tất cả ảnh trong bộ dữ liệu thì nên có trọng số thấp hơn vì nó thường cung cấp ít thông tin hơn. Để làm điều này, độ đo sử dụng trọng số TF-IDF cho mỗi n-grams.

Số lần một n-grams w_k xuất hiện trong một reference sentence s_{ij} được biểu thị là $h_k(s_{ij})$ hoặc $h_k(c_i)$ cho candidate sentence c_i. Tính toán trọng số TF-IDF cho mỗi n-grams $g_k\left(s_{ij}\right)$ như sau:

$$g_k\left(s_{ij}\right) = \frac{h_k\left(s_{ij}\right)}{\sum_{w_l \in \Omega} h_l\left(s_{ij}\right)} \log\left(\frac{|I|}{\sum_{I_p \in I} \min\left(1, \sum_q h_k\left(s_{pq}\right)\right)}\right) \tag{2.2}$$

Với Ω là tập từ vựng của tất cả các n-grams và I là tập tất cả ảnh trong bộ dữ liệu. Theo công thức trên, cụm đầu tiên (trước log) tính TF cho mỗi n-gram w_k, cụm thứ hai tính độ hiếm của w_k sử dụng IDF. TF sẽ đánh trọng số cao hơn vào những n-grams thường xuất hiện trong reference sentence cho một ảnh trong khi IDF giảm trọng số của các n-grams xuất hiện phổ biến trong tất cả ảnh trong bộ dữ liệu. Điều này có nghĩa là, IDF cung cấp một thước đo độ quan trọng của từ bằng cách giảm trọng số các từ phổ biến có khả năng mang ít thông tin trực quan hơn. IDF được tính bằng cách sử dụng logarit của số lượng hình ảnh trong tập dữ liệu $|I|$ chia cho số lượng hình ảnh mà w_k xuất hiện trong bất kỳ reference sentence nào của nó.

Điểm $CIDEr_n$ cho n-grams với chiều dài n được tính bằng trung bình cosine giữa candidate sentence và reference sentences, vốn cân bằng giữa precision và recall:

$$CIDEr_n\left(c_i, S_i\right) = \frac{1}{m} \sum_j \frac{g^n\left(c_i\right) \cdot g^n\left(s_{ij}\right)}{\left\|g^n\left(c_i\right)\right\| \left\|g^n\left(s_{ij}\right)\right\|}$$

Với $g^n\left(c_i\right)$ là một vector tạo thành bởi $g_k\left(c_i\right)$ ứng với tất cả n-grams có chiều dài n và $\left\|g^n\left(c_i\right)\right\|$ là độ lớn của vector $g^n\left(c_i\right)$. Tương tự cho $g^n\left(s_{ij}\right)$.

Thông thường, người ta dùng n -grams có thứ tự cao hơn (dài hơn) để nắm bắt các đặc tính ngữ pháp cũng như ngữ nghĩa giàu hơn. Có thể kết hợp các điểm số n -grams của các độ dài khác như như sau:

$$CIDEr\left(c_i, S_i\right) = \sum_{n=1}^{N} w_n CIDEr_n\left(c_i, S_i\right)$$

Trong thực tế, các nhà nghiên cứu (nhóm tác giả bài báo) nhận thấy rằng các trọng số đồng đều (uniform weights) $w_n = 1/N$ là tốt nhất. Trong bài báo các tác giả lấy $N = 4$.

Chương 3

Phương pháp thực hiện

Hướng tiếp cận phổ biến cho bài toán Image captioning là sử dụng một mô hình encoder để biểu diễn đặc trưng của ảnh. Đặc trưng này sau đó qua một mô hình decoder để tạo câu caption cho ảnh đầu vào. Hình 3.1 minh họa pipeline chúng thôi sử dụng trong đề tài này.

3.1 Tiền xử lí dữ liệu

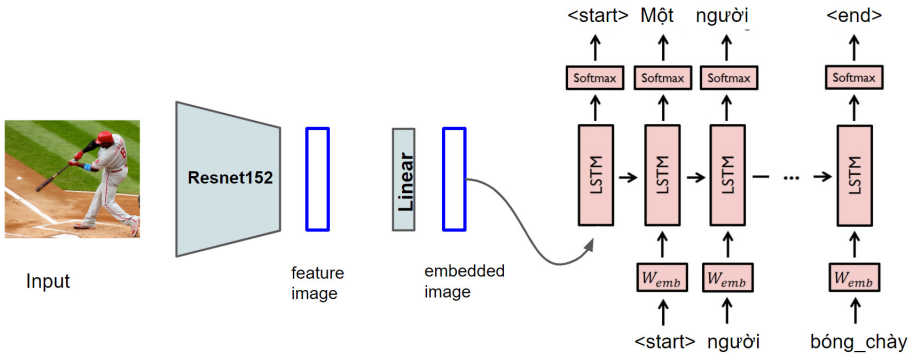
Những câu caption trong bộ dữ liệu UIT-ViIC được gán nhãn thủ công nên có những lỗi về ngữ pháp, chính tả hoặc khoảng cách, sai dấu câu, ... Vì vậy, trước khi huấn luyện mô hình, chúng tôi đã tiến hành loại sửa những lỗi này.Hình 3.2 minh họa một ảnh có câu caption bị sai.

3.2 Phương pháp thực hiện

3.2.1 Rút trích đặc trưng ảnh

Ảnh đầu vào được rút trích đặc trưng từ những bộ rút trích đặc trưng đã được huấn luyện trên bộ dữ liệu ImageNet[5] như Inception-V[6], EfficientNetB4[7] và Resnet152[**resnet**]. Đặc trưng được rút trích là đầu racủa mô hình sau khi loại bỏ lớp Fully connected và Softmax.

Sau công đoạn này, một ảnh input sẽ được biểu diễn bằng một vector 2048 chiều. Tuy nhiên đầu vào của LSTM được quy định ở đây là một vector 512 chiều nên cần một



HÌNH 3.1: Pipeline bài toán Image Captioning.

1. Các cầu thủ bóng đá dang di chuyển ở trên sân để tranh bóng.
2. Một nam thủ môn đang truwowc trên sân để bắt bóng.
3. Các cầu thủ bóng đá đang thi đấu ở trên sân. /
4. Một cầu thủ áo xanh đang khuỵu gối ở cạnh quả bóng.
5. Những cậu bé đang chơi bóng đá ở trên sân.

HÌNH 3.2: Ví dụ về lỗi ngữ pháp, dấu câu của caption trong bộ dữ liệu

lớp image embedding với nhiệm vụ ánh xạ vector 2048 chiều này về 512 chiều thích hợp để đưa vào input mạng RNN.

3.2.2 Word segmentation cho tiếng Việt

Do sự nhập nhằng giữa các âm tiết của Tiếng Việt. Trước đi đưa vào huấn luyện, chúng tôi đã tiến hành Word Segmentation sử dụng package Underthesea cho những câp caption của mỗi ảnh.

Bên cạnh đó, mỗi câu caption sẽ được thêm tiền tố <start> và hậu tố <end> để mô hình RNN học được từ bắt đầu câu mà kết thúc câu.

- Caption gốc: Một người đàn ông đang xoay người để đánh quả bóng chày.
- Caption sau khi xử lí : <start> Một người đàn_ông đang xoay người để đánh_quả bóng_chày <end>

3.2.3 Cơ chế attention

Với bộ não con người, khi nhìn một bức ảnh, chúng ta sẽ chỉ chú trọng đến các vùng nổi bật nhất của nó để từ đó đưa ra câu miêu tả nội dung phù hợp. Cơ chế Attention hoạt động bằng cách đảo ngược nguyên lý này. Nghĩa là với một word trong câu caption thì vùng nào trên ảnh sẽ có tác động nhất đến word này?

Tại ICML 2015, Kelvin Xu và các cộng sự đã lần đầu áp dụng cơ chế này để giải quyết bài toán Image Captioning trong bài báo "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention" [9].

Có 2 cơ chế attention được tác giả giới thiệu:

- "Hard" Attention: theo cơ chế này thì một vùng trong ảnh hoặc là liên quan đến word đang dự đoán hoặc không liên quan gì cả. Nói cách khác, mỗi word sẽ chỉ do một vùng ảnh nhất định tác động và các vùng khác không có cống hiến gì đến word này. Model theo cơ chế này có thể được huấn luyện bằng cách các vùng sẽ được khởi tạo ngẫu nhiên và áp dụng các quy tắc của REINFORCE learning – trao thưởng cho các vùng chọn phù hợp với word – để tối ưu hóa trọng số.

- "Soft" Attention: khác với ở trên, "soft" attention quan niệm là tất cả các pixels trong ảnh đều có đóng góp gì đó với word đang dự đoán, chỉ khác nhau là nhiều hay ít mà thôi. Như vậy, ta cần xác định trọng số ảnh hưởng của từng pixel đến một word, pixel có ảnh hưởng càng lớn thì trọng số càng cao. Việc này có thể thực hiện bằng cách áp dụng một hàm Softmax. Quá trình train model theo cơ chế này được thực hiện đơn giản bằng lan truyền ngược thông thường.

3.2.4 Tạo câu caption sử dụng RNN

Sau khi có vector biểu diễn đặc trưng của ảnh đầu vào và trọng số attention. Chúng tôi huấn luyện mô hình decoder sử dụng GRU và LSTM.

Chương 4

Kết quả thực nghiệm

4.1 Bộ dữ liệu

4.1.1 UIT-ViIC

Bộ dữ liệu UIT-ViIC [**fer**] bao gồm 19,250 tiêu đề (captions) tiếng Việt cho 3,850 ảnh lấy từ bộ dữ liệu Microsoft COCO[3].

Tập train có 2695 ảnh, tập val có 924 ảnh, tập test có 231 ảnh.

- Kích thước mỗi ảnh:
- Kích thước tập ảnh train: 2695
- Kích thước tập ảnh validation: 924
- Kích thước tập ảnh test: 231

4.2 Các thực nghiệm và kết quả

4.2.1 Các thực nghiệm

Chúng tôi tiến hành thuấn luyện và tinh chỉnh mô hình encoder và decoder cho bài toán Image captioning.

Những thực nghiệm dưới đây chúng tôi tiến hành huấn luyện trên bộ dữ liệu UIT-ViIC theo tham số :
- Learning rate: 1e-3, 1e-4, 1e-5
- Batch size: 512
- Epoch: 2000
- Framework: Tensorflow, Keras

- **Rút trích đặc trưng trên nhiều mô hình:** Chúng tôi đã tiến hành thực nghiệm rút trích đặc trưng của 3 mô hình Inception-V3, Resnet152 và EfficientNetB4. Sử dụng cơ chế attention và GRU. Kết quả thu được mô hình rút trích đặc trưng bằng Resnet152 cho kết quả CIDEr-D cao nhất (0.762).
- **Tạo câu caption bằng GRU và LSTM:** Chúng tôi sử dụng mô hình Resnet152 để rút trích đặc trưng kết hợp với cơ chế attention làm đầu vào cho mô hình GRU và LSTM. Kết quả thu được cho thấy LSTM cho kết quả thấp hơn so với GRU.

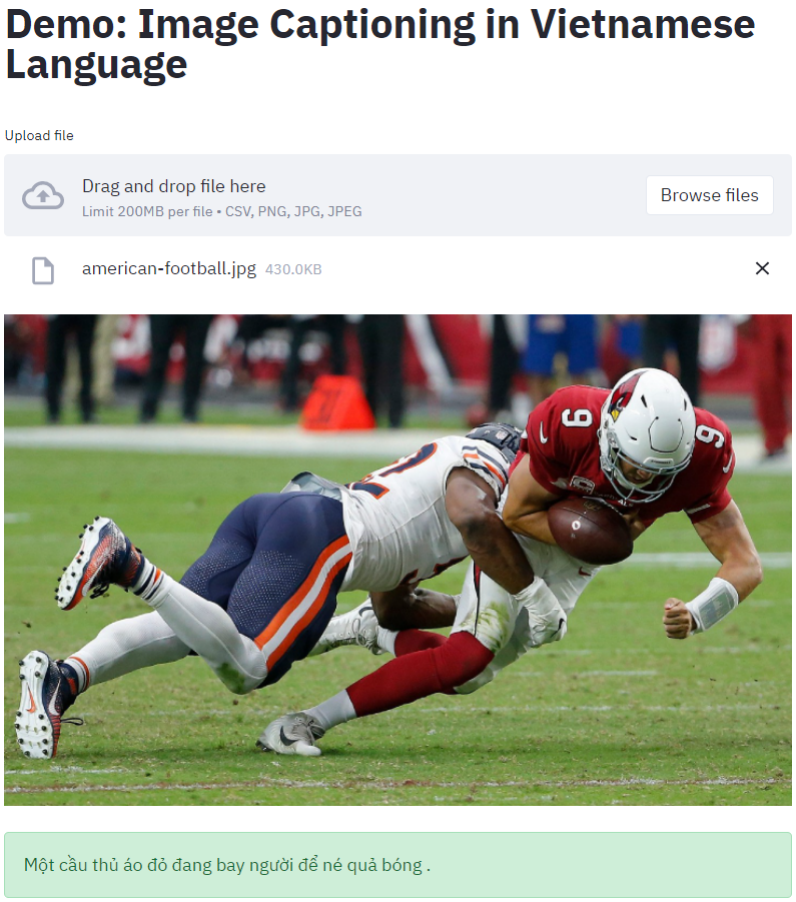
TN	Encoder	Decoder	Att?	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE-L	CIDEr-D
1	InceptionV3	GRU	x	0.563	0.438	0.344	0.270	0.483	0.645
2	EfficientNetB4	GRU	x	0.571	0.455	0.366	0.296	0.504	0.662
3	Resnet152	GRU	x	0.630	0.517	0.426	0.353	0.541	0.762
4	Resnet152	LSTM	x	0.551	0.433	0.340	0.266	0.486	0.586
5	Resnet152	LSTM	-	**0.669**	**0.544**	**0.452**	**0.380**	**0.575**	**1.031**

BẢNG 4.1: Một số kết quả thực nghiệm

- **Không dùng attention:** Chúng tôi tiến hành sử dụng Resnet152 kết hợp với LSTM để huấn luyện. Không dùng cơ chế attention. Kết quả đánh giá độ đo CIDEr-D cao nhất. Nguyên nhân do quá trình huấn luyện chưa tới hoặc thiếu dữ liệu dẫn đến trong số attention chưa học được một cách chính xác.

4.3 Demo

Chúng tôi thực hiện một bản demo cho bài toán Image Captioning cho tiếng Việt.



HÌNH 4.1: Chú thích một ảnh môn bóng bầu dục, bộ môn không nằm trong dataset

Chương 5

Kết luận và hướng phát triển

5.1 Kết luận

Từ những mục tiêu đã đề ra, trong đề tài này chúng tôi đã thực hiện được những việc và kết luận như sau:

- Tiền xử lí lỗi câu caption của dữ liệu.
- Xây dựng được một hệ thống tạo chú thích cho ảnh một cách tự động bằng Tiếng Việt.
- Xây dựng được một chương trình demo.

5.2 Hướng phát triển

Thông qua một số thử nghiệm kết hợp các mô hình đã được huấn luyện từ trước cùng với các phương pháp giúp cân bằng giữ liệu, chúng tôi đưa ra một số hướng phát triển như sau:

- Tạo thêm dữ liệu và huấn luyện tiếp mô hình để mô hình học tốt hơn.
- Thực nghiệm các phương pháp mới nhất hiện nay cho bài toán này.

Bibliography

[1] Quan Hoang Lam et al. *UIT-ViIC: A Dataset for the First Evaluation on Vietnamese Image Captioning*. 2020. arXiv: 2002.00175 [cs.CL].

[2] Chin-Yew Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*. 2004, pp. 74–81.

[3] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[4] Kishore Papineni et al. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[5] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[6] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[7] Mingxing Tan and Quoc V Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *arXiv preprint arXiv:1905.11946* (2019).

[8] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. *CIDEr: Consensus-based Image Description Evaluation*. 2015. arXiv: 1411.5726 [cs.CV].

[9] Kelvin Xu et al. "Show, attend and tell: Neural image caption generation with visual attention". In: *International conference on machine learning*. PMLR. 2015, pp. 2048–2057.