MÔ HÌNH NHẬN DIỆN CHỮ NGHỆ THUẬT

Mục lục

1 Giới thiệu chung

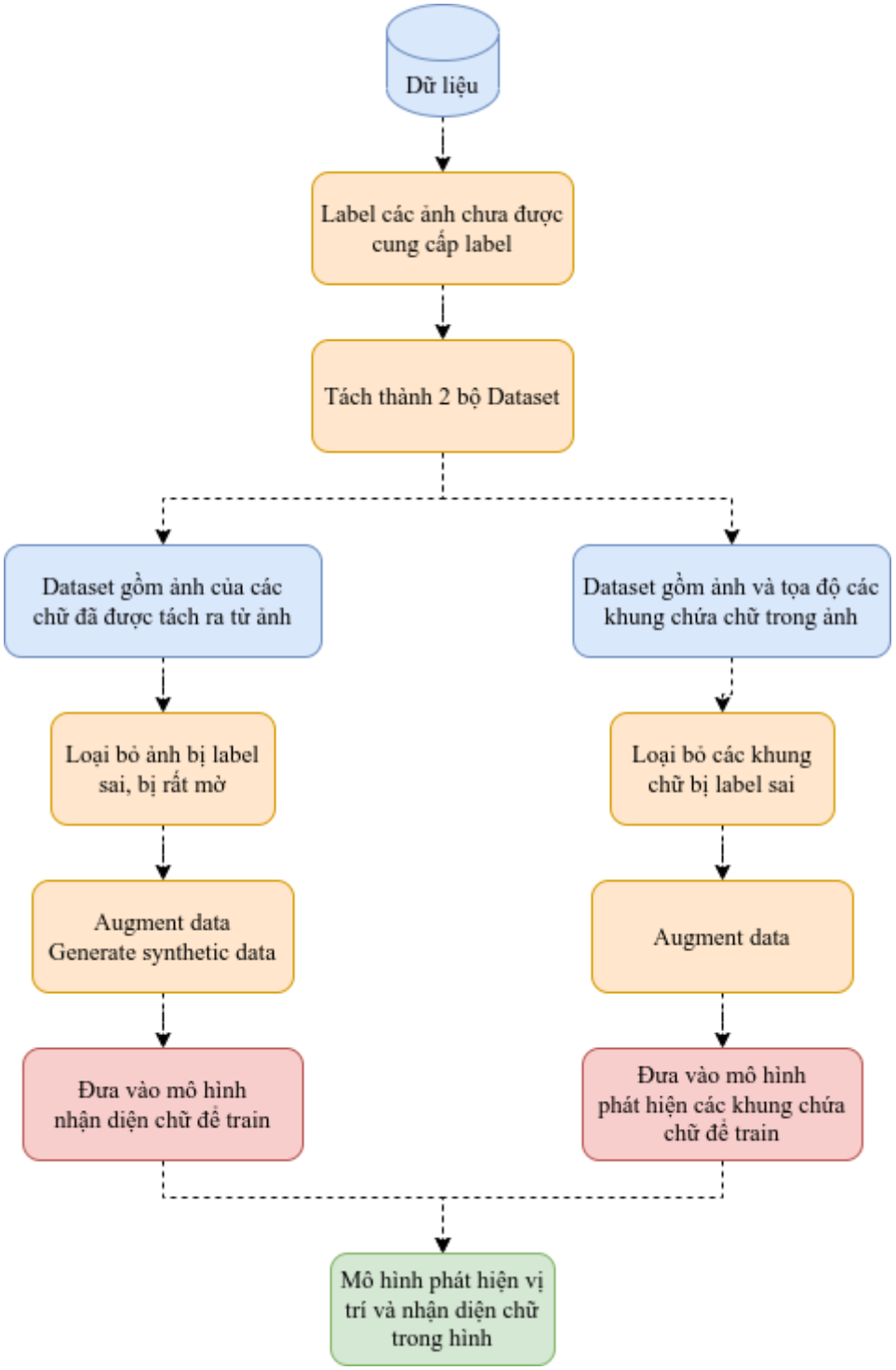
Artificial Intelligence (A.I) nói chung đã xuất hiện từ lâu và có vô số những ứng dụng thực tiễn trong đời sống hằng ngày của chúng ta. Để có được những ứng dụng ấy là cả quá trình nghiên cứu, mà cụ thể là giải quyết

những bài toán từ đơn giản tới phức tạp. Lần này, tham gia cuộc thi UIT-AI Challenge 2022 với chủ đề "ARTISTIC TEXT CHALLENGE", nhóm chúng mình gặp phải những bài toán điển hình đó là phát hiện chữ (Text Detection) và nhận diện chữ (Text Recognition) trong hình, tập trung vào những chữ nghệ thuật. Bài viết này sẽ nói cụ thể về những vấn đề chúng mình gặp phải và những ý tưởng, giải pháp chúng mình đã sử dụng để giải quyết những vấn đề ấy.

2 Quá trình thực hiện

2.1 Pipeline

Với mỗi bài toán Text Detection và Text Recognition, chúng mình sẽ sử dụng tương ứng model YOLOv7 và model SRN để giải quyết, việc đầu tiên cần phải làm ấy là chuẩn bị nguồn dữ liệu đầu vào cho 2 model này.



2.2 Chuẩn bị dữ liệu (Data Preparation)

Sau khi phân tích bộ dữ liệu (Dataset) gồm ảnh và nhãn (label) được BTC cung cấp, nhận ra bộ dữ liệu có khá nhiều lỗi nên chúng mình quyết định sẽ xem xét từng ảnh và từng label trước khi đem dữ liệu vào các mô hình để tiến hành huấn luyện.

Đầu tiên, với những ảnh chưa có nhãn, chúng mình sẽ gán nhãn thủ công cho các ảnh ấy, sau đó chia bộ Dataset do BTC cung cấp thành 2 bộ Dataset khác nhau:

- Bộ Dataset đầu tiên (gọi là Dataset_1) sẽ chứa ảnh do BTC cung cấp, cùng với các file label chứa tọa độ của các khung chữ theo format của model YOLOv7.
- Bộ Dataset còn lại (gọi là Dataset_2) sẽ chứa các ảnh được cắt ra dựa trên tọa độ đã được label, cùng với file label theo format của model SRN (framework PaddleOCR).

Tiếp đến, chúng mình sẽ tiến hành kiểm tra dữ liệu:

- Đối với Dataset_1: kiểm tra xem liệu các tọa độ đã được label của các ảnh có phải là tọa độ của các chữ không, nếu không sẽ label lại hoặc xóa.
- Đối với Dataset_2: kiểm tra xem ảnh đó có chứa chữ hay không và chữ đó có giống với chữ được label không rồi tiến hành xóa các ảnh có kích thước mỗi cạnh nhỏ hơn 5px (vì những ảnh này thường rất mờ hoặc là ảnh bị label lỗi), cuối cùng kiểm tra nhanh các ảnh còn lại, nếu bị quá mờ thì xóa.

Và với mỗi bộ Dataset, chúng mình sẽ chia Dataset đó thành hai phần, một phần sử dụng cho việc huấn luyện mô hình (ảnh train) và phần còn lại dùng để đánh giá mô hình (ảnh val) theo tỉ lệ 80-20. Đồng thời, chúng mình còn tiến hành sinh thêm ảnh để tăng lượng dữ liệu đầu vào cho mô hình và tránh được hiện tượng overfitting:

- Đối với Dataset_1, chúng mình sử dụng thư viện imgaug[1] để tiến hành augment các ảnh train trong Dataset bằng nhiều cách như: Rotate, Perspective Transform, Shear, Blur, Dropout, Enhance,... Thông qua phương pháp này, chúng mình có khoảng 12,000 ảnh train và khoảng 600 ảnh val.
- Đối với Dataset_2, chúng mình vẫn tiếp tục sử dụng thư viện imgaug[1] để tiến hành augment các ảnh train trong Dataset. Sau đó sử dụng thư viện TextRecognitionDataGenerator[2] để tiến hành sinh ra các synthetic text. Đối với phương pháp này chúng mình có khoảng 250,000 ảnh train và khoảng 6,000 ảnh val.

2.3 Huấn luyện mô hình (Model Training)

Như đã đề cập ở trên, chúng mình sẽ tiến hành huấn luyện 2 model phục vụ cho 2 mục đích khác nhau:

- Với bài toán Text Detection, chúng mình sử dụng model YOLOv7 được train từ đầu với dữ liệu đầu vào là bộ Dataset_1.
- Với bài toán Text Recognition, chúng mình sử dụng model SRN đã được pretrained trên tập dữ liệu tiếng anh ICDAR2015 kết hợp với dữ liệu đầu vào là bộ Dataset_2.

2.4 Đánh giá mô hình (Model Validating)

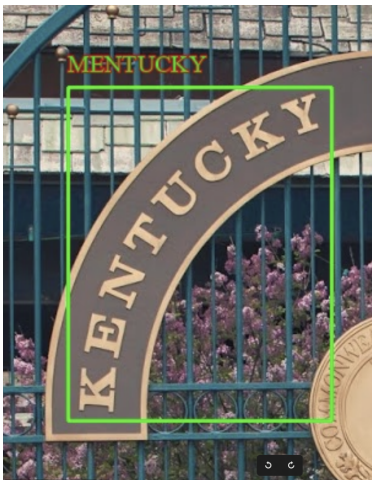
Sau khi huấn luyện tương đối, chúng mình tiến hành đánh giá sơ bộ để rút ra những cản trở làm giảm hiệu quả của các model:

- Text Detection: YOLOv7 hoạt động tốt với phần lớn các ảnh nhưng khá không hiệu quả với những ảnh có các chi tiết trong điều kiện phức tạp như cây cối, ánh sáng,... và cho ra nhiều ảnh nhiễu (ảnh không phải chữ, ảnh chứa chữ cực kì mờ,…). Bên cạnh đó, vì mô hình chỉ phát hiện chữ theo các khung hình chữ nhật, nên kết quả thu được là các bounding box có kích thước lớn đối với các chữ cong.

Mô hình nhận diện cả phần bảng không sáng đèn và hình ô tô bên cạnh là chữ

Mô hình cho bounding box lớn với chữ cong

- Text Recognition: Vì lượng dữ liệu đầu vào là ảnh nghệ thuật khá ít, nên SRN hoạt động không tốt lắm với những kiểu ảnh có chứa chữ nghệ thuật và chữ bị cong.



Mô hình nhận diện sai một số chữ nghệ thuật và chữ bị cong

2.5 Tối ưu mô hình (Model Optimizing)

Vì tiêu chí đánh giá mô hình của cuộc thi dựa trên chỉ số chính là weighted-F1-score, tức điểm số càng cao thì mô hình nhận diện càng chính xác, nên mục tiêu chính của chúng mình là làm sao để có F1-Score cao nhất có thể.

$$FP = incorrect\_pred$$

$$FN = missing\_gt\_normal\_text + weight \times missing\_gt\_art\_text$$

$$TP = correct\_pred\_normal\_text + weight \times correct\_pred\_art\_text$$

$$Precision = \frac{TP}{TP + FP}; Recall = \frac{TP}{TP + FN}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

- Text Detection: nguyên nhân chính là do model YOLOv7 detect luôn những vật thể (object) không phải là text (và cũng không có trong đáp án của BTC). Chính vì thế chúng mình đã nảy ra một ý tưởng là set một giá trị threshold (confidence threshold) cho model YOLOv7 nhằm loại bỏ phần nào những object ấy, từ đó giúp tăng điểm Precision. Mặt khác, chính việc loại bỏ này cũng vô tình loại bỏ luôn một số text trong ảnh, khiến cho điểm Recall cũng giảm theo. Tuy nhiên lượng điểm Recall giảm là không đáng kể so với lượng tăng của điểm Precision (vì mô hình YOLOv7 detect ra phần lớn đều là những ảnh không rõ chữ hoặc ảnh không liên quan). Do đó, nhìn chung phương án này vẫn giúp tăng điểm F1 score.
- Text Recognition: chúng mình có thử sinh thêm các ảnh chứa chữ nghệ thuật nhằm khắc phục sự hạn chế của dữ liệu đầu vào. Tuy nhiên, kết quả thu được là không mấy khả quan. Cộng với sự gấp rút của thời gian các vòng thi, phương án khả thi nhất chúng mình có là tinh chỉnh các thông số như threshold, hyperparameters, …
- Cuối cùng, chúng mình tận dụng tối đa số lượng submission cho phép ở mỗi phase, điển hình là 47/60 lần ở “SystemTestingPhase”. Tuy không lấy gì làm tự hào nhưng như đã nói ở trên, mục tiêu chính của chúng mình là tối đa hoá số điểm F1-Score có thể đạt được, cách làm này là có thể hiểu được.

3 Tổng kết

Tóm lại, qua bài viết, chúng mình đã trình bày chi tiết quy trình cũng như những ý tưởng, hướng giải quyết với những vấn đề gặp phải trong cuộc thi UIT-AI Challenge 2022 lần này. Bởi vì là lần đầu tham gia một cuộc thi về AI, nên đã có nhiều điểm chúng mình làm chưa tốt và không hợp lý. Cuộc thi lần này đã cho chúng mình khá nhiều bài học và kinh nghiệm cũng như tiếp thêm động lực trong việc tìm tòi và nghiên cứu, chúng mình sẽ tiếp tục nỗ lực học hỏi để gặt hái những thành quả tốt hơn trong các cuộc thi sắp tới.

Tham khảo

[1] Aleju. (2020) Image Augmentation for machine learning experiments. *GitHub* [Online]. Truy cập tại: https://github.com/aleju/imgaug

[2] Belval. (2022) TextRecognitionDataGenerator: A synthetic data generator for text recognition. *GitHub* [Online]. Truy cập tại: https://github.com/Belval/TextRecognitionDataGenerator

[3] Atienza, R. *Data Augmentation for Scene Text Recognition* [Online]. Truy cập tại: https://arxiv.org/pdf/2108.06949.pdf

[4] PaddlePaddle. (2022) PaddleOCR. *GitHub* [Online]. Truy cập tại: https://github.com/PaddlePaddle/PaddleOCR/blob/release/2.6/doc/doc_en/algorithm_rec_srn_en.md.

[5] WongKinYiu. (2022) yolov7. *Github* [Online]. Truy cập tại: https://github.com/WongKinYiu/yolov7

[6] Hung, C.V. (2022) *Baseline-UAIC* [Online]. Truy cập tại: https://colab.research.google.com/drive/19NFpiXvRwW21lqrtAJCf0-K3-PNcfdPl