ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Môn học: PHÂN TÍCH XỬ LÝ ẢNH

NHẬN DIỆN CHỮ VIẾT TAY TIẾNG VIỆT

Sinh viên thực hiện:

1. Lê Nho Hãn – 22110054
2. Tạ Quang Duy – 22110047
3. Mai Nguyễn Ngọc Duy – 21110275
4. Phan Thái Hòa – 20110190

Giảng viên môn học:

Huỳnh Thanh Sơn

Mục lục

I. Giới thiệu đề tài:

1. Giới thiệu về OCR:

Optical Character Recognition (OCR), hay nhận dạng ký tự quang học, là công nghệ được sử dụng để chuyển đổi các loại tài liệu khác nhau, chẳng hạn như tài liệu giấy được quét, tệp PDF hoặc hình ảnh chụp bằng máy ảnh kỹ thuật số, thành dữ liệu có thể chỉnh sửa và tìm kiếm được. OCR đã và đang đóng vai trò quan trọng trong các ứng dụng thực tế như nhận diện chữ viết tay, và trích xuất thông tin từ hình ảnh.

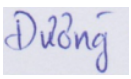
Cho đến nay, đã có nhiều mô hình xử lý ảnh phổ biến với bài toán như trên, như là CRNN (Convolutional Recurrent Neural Network) kết hợp các mạng tích chập (CNN) để trích xuất đặc trưng và mạng hồi tiếp (RNN) để xử lý chuỗi ký tự; sử dụng kiến trúc Encoder-Decoder với mã hóa (Encoder) để trích xuất đặc trưng và bộ giải mã (Decoder) để sinh văn bản đầu ra; mô hình ngôn ngữ thị giác (Vision-Language Models), ...

2. Những thách thức và lựa chọn bài toán:

Việc phân tích xử lý ảnh gặp nhiều thách thức do có nhiều biến đổi trong văn bản, chẳng hạn như những biến dạng trong chữ viết tay, phông chữ, nền lộn xộn, thiếu dữ liệu được chú thích đầy đủ cho chữ viết tay hoặc ngôn ngữ cổ xưa, hiếm được sử dụng. Cụ thể hơn, RNN xử lý dữ liệu theo tuần tự, dẫn đến tốc độ chậm, đặc biệt khi xử lý chuỗi dài; kiến trúc Encoder-Decoder suy giảm khả năng nắm bắt mối quan hệ ngữ cảnh do độ dài chuỗi tăng hay yêu cầu tài nguyên lớn, đặc biệt khi sử dụng cơ chế Attention để học mối quan hệ giữa các ký tự; các mô hình Vision-Language thường yêu cầu lượng lớn dữ liệu có chú thích để huấn luyện,...

Các phương pháp cho mô hình Text Recogntion phổ biến nhất bao gồm 2 phần. Phần một là kiến trúc CNN đóng vai trò như một bộ trích xuất đặc trưng. Phần hai là kiến trúc chuyển bản đồ đặc trưng thành dạng chuỗi. Phần hai này có thể dùng các lớp mạng RNN hoặc Transformer. Nhằm cải thiện tốc độ cũng như đơn giản hóa cấu trúc xử lý ảnh, nhóm sẽ giới thiệu một mô hình là Single Visual Model trong xử lý nhận diện ảnh với bài toán cụ thể là **nhận diện chữ viết tay tiếng Việt**, dựa trên bài báo Scene Text Recognition with a Single Visual Model, với:

- **Input:** một ảnh chứa văn bản được chụp hoặc scan và chưa được xử lý. Ảnh phải bao hàm và bo sát văn bản cần nhận diện. Ảnh này chứa 1 kí tự/văn bản.



- **Output:** một ký tự/văn bản đã được nhận dạng từ ảnh.

Đường

II. Kiến trúc STVR:

Scene Text Recognition with Single Visual Model (SVTR) được thiết kế như một mạng ba giai đoạn (three-stage network), chuyên biệt cho bài toán nhận diện văn bản trong ảnh. Quá trình xử lý tổng quan như sau:

Ảnh đầu vào $H \times W \times 3$ được chuyển đổi thành các đoạn nhỏ (*patch*) kích thước $\frac{H}{4} \times \frac{W}{4} \times D_0$ thông qua tích chập (3×3, stride 2), tạo ra các vector đặc trưng D_0 làm đầu vào CC_0. Trong giai đoạn trích xuất đặc trưng, CC_0 được xử lý qua ba giai đoạn: *Stage 1* sử dụng *Mixing Blocks* và *Merging* để tạo CC_1 kích thước $\frac{H}{8} \times \frac{W}{4} \times D_1$; *Stage 2* tiếp tục xử lý CC_1, tạo CC_2 kích thước $\frac{H}{16} \times \frac{W}{4} \times D_2$; và *Stage 3* thay *Merging* bằng *Combining*, cho đầu ra C kích thước $1 \times \frac{W}{4} \times D_3$. Cuối cùng, đặc trưng đầu ra được đi qua một Fully-Connected Layer để dự đoán văn bản nhận diện từ ảnh.

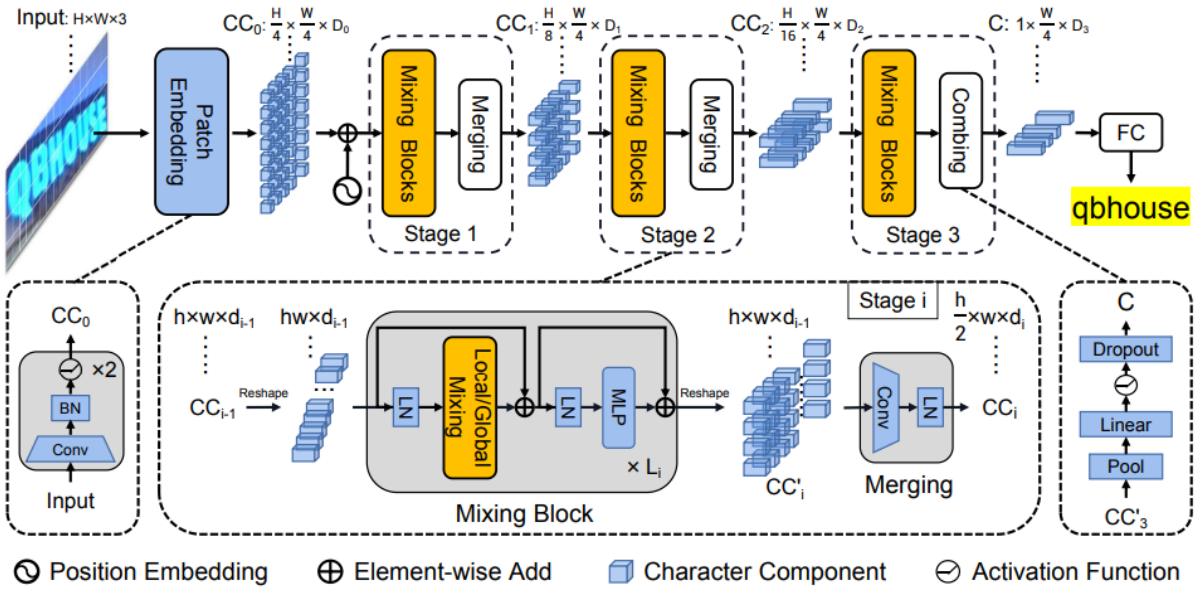


Figure 1: Sơ đồ mô hình

1. Progressive Overlapping Patch Embedding:

Progressive Overlapping Patch Embedding có thể hiểu là một thuật ngữ dùng để mô tả quá trình chuyển đổi ảnh đầu vào thành các đoạn nhỏ với cách tiếp cận đặc biệt nhằm giữ lại nhiều thông tin quan trọng.

Ở bước này, mục tiêu là chuyển đổi ảnh đầu vào $X \in R^{H \times W \times 3}$ thành các patch $CC_0 \in R^{\frac{H}{4} \times \frac{W}{4} \times D_0}$, có thể hiểu rằng CC_0 như là bước đầu tiên trong quy trình trích xuất đặc trưng ảnh văn bản, cung cấp dữ liệu để các khối Mixing Block và Merging tiếp tục xử lý. Phương pháp thực hiện là sử dụng hai phép chập liên tiếp 3×3 với stride 2. Điều này chia ảnh thành các "patch" nhỏ, đồng thời giữ lại mối quan hệ không gian nhờ cơ chế overlapping (stride nhỏ hơn kích thước kernel). Overlapping giúp đảm bảo rằng các patch liền kề không bị mất thông tin ở biên giữa chúng, cải thiện khả năng trích xuất và tổng hợp thông tin từ ảnh. Sau mỗi lớp tích chập, áp dụng Batch Normalization để cải thiện hiệu quả huấn luyện, giúp ổn định việc lan truyền tín hiệu qua các lớp. Trong quá trình huấn luyện, còn sử dụng hàm kích hoạt GeLU để tăng khả năng học phi tuyến tính, đồng thời kết hợp Batch Normalization để giảm hiện tượng overfitting và tăng tốc độ hội tụ của mô hình.

Đầu ra CC_0 mang thông tin về nội dung, do đó Position Embedding chứa thông tin về vị trí không gian sẽ được sử dụng bằng phép cộng Element-wise Add kết hợp hai nguồn thông tin này lại, giúp mỗi đặc trưng trong CC_0 được gắn với một vị trí cụ thể trong không gian ảnh, nhằm chuẩn bị cho cơ chế Self-Attention trong Mixing Blocks.

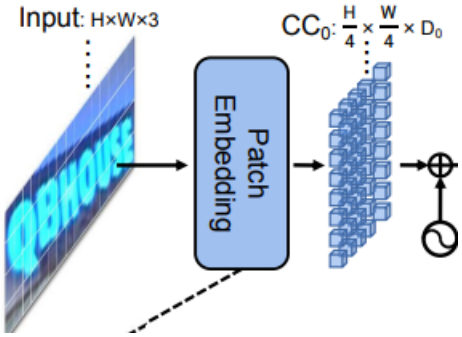


Figure 2: Bước đầu tiên

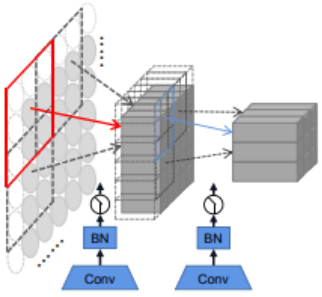


Figure 3: Quá trình Patch Embedding

2. Mixing Block:

Vì hai ký tự có thể khác nhau một chút nên việc nhận diện văn bản chủ yếu dựa vào các đặc điểm ở cấp thành phần kí tự. Có 2 đặc điểm quan trọng:

+ Đặc trưng cục bộ: Các nét chữ, đường cong, cạnh, hoặc hình dạng cơ bản của một ký tự. Đặc trưng này giúp mô hình nắm bắt hình thái (morphology) và mối liên hệ bên trong một ký tự (correlation within a character).

+ Sự phụ thuộc giữa các ký tự: Có thể hiểu rằng đây là mối quan hệ giữa các ký tự khác nhau trong văn bản hoặc giữa các ký tự với các thành phần không phải văn bản

(non-text components). Loại đặc trưng này quan trọng khi xử lý văn bản liền mạch, ví dụ: chữ viết tay không có khoảng cách rõ ràng giữa các ký tự) hay các văn bản có nhiễu từ nền hoặc chứa các thành phần không liên quan (non-text).

Khối trộn - Mixing Block là thành phần cốt lõi của mô hình, thực hiện việc trộn lẫn thông tin giữa các patch từ đầu vào CC_0, trong đó:

- Local Mixing: tập trung vào việc trộn lẫn thông tin giữa các pixel lân cận trong cùng 1 patch.
- Global Mixing: tập trung vào việc trộn lẫn thông tin giữa các patch khác nhau trong toàn bộ hình ảnh.

Mô hình được đề xuất trong SVTR sử dụng hai loại mixing blocks với cơ chế self-attention của mô hình Transformer để học và trích xuất các đặc trưng cần thiết, từ đó cải thiện độ chính xác trong nhận diện văn bản.

Trước hết, ảnh đầu vào CC_{i-1} với kích thước $h \times w \times d_{i-1}$ sẽ điều chỉnh lại (reshape) thành $hw \times d_{i-1}$ và sau đó tiến vào các khối trộn toàn cục và trộn cục bộ. Reshape đơn giản là "trải phẳng" không gian $h \times w$ thành một chiều hw trong khi vẫn giữ nguyên số kênh d_{i-1}.

Global Mixing: Trộn toàn cục là một quá trình đánh giá sự phụ thuộc lẫn nhau giữa tất cả các thành phần ký tự trong một hình ảnh. Nó giúp thiết lập mối quan hệ lâu dài giữa các ký tự khác nhau và giảm ảnh hưởng của các thành phần không phải văn bản trong khi nhấn mạnh tầm quan trọng của các thành phần văn bản. Vì văn bản và phi văn bản là hai thành phần chính trong một hình ảnh, nên việc trộn mục đích chung như vậy có thể thiết lập sự phụ thuộc lâu dài giữa các thành phần từ các ký tự khác nhau. Bên cạnh đó, nó cũng có khả năng làm suy yếu ảnh hưởng của các thành phần phi văn bản, đồng thời tăng cường tầm quan trọng của các thành phần văn bản.

Local Mixing: Trộn cục bộ - local mixing được sử dụng để đánh giá mối quan hệ giữa các thành phần trong một ký tự bằng cách xem xét một cửa sổ lân cận. Nó nhằm mục đích nắm bắt các đặc điểm giống nét vẽ quan trọng để nhận dạng ký tự và sử dụng $self - attention$ để nắm bắt các mẫu cục bộ. Trộn cục bộ cũng sử dụng cơ chế cửa sổ trượt (sliding window) trượt trên các vùng kích thước 7×11 và tính toán mối liên hệ giữa các phần tử trong vùng cửa sổ đó, các phần tử ở đây chính là các phần tử được chia qua lớp Patch Embedding được trình bày bên trên.

Sau hoạt động của Local Mixing và Global Mixing, đầu ra được cộng theo từng phần tử với đầu vào ban đầu của bước Mixing Block. Ngay sau đó, Layer Normalization tiếp tục được áp dụng một lần nữa để tái chuẩn hóa các phần tử, tiếp tục đưa các phần tử qua một Multi-Layer Perceptron (MLP) để tăng cường khả năng học phi tuyến tính và mở rộng không gian đặc trưng. Kết quả đầu ra này lại một lần nữa được cộng theo từng phần tử với đầu ra sau 2 khối trộn Local/Global Mixing và được reshape lại thành đầu ra CC'_i có kích thước $h \times w \times d_{i-1}$ để qua bước xử lý tiếp theo.

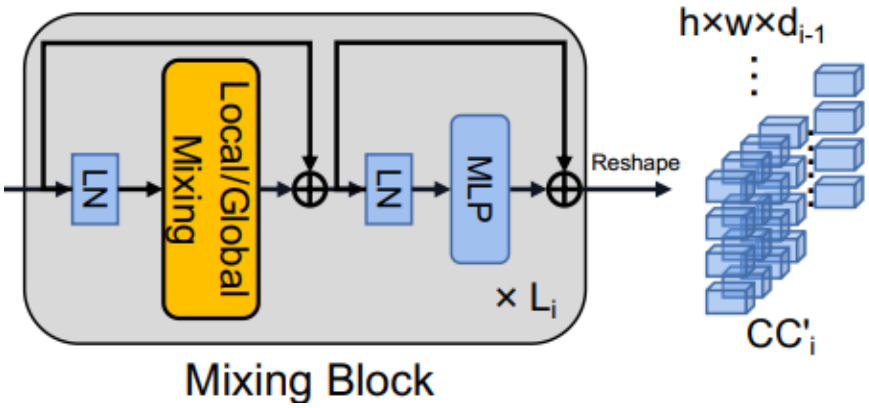


Figure 4: Quá trình Mixing Block

3. Merging:

Sau các khối trộn một hoạt động hợp nhất - merging được đưa vào sau các khối trộn trong mỗi giai đoạn (trừ giai đoạn cuối cùng).

Để thực hiện điều này, đầu ra CC'_i có kích thước $h \times w \times d_{i-1}$ sau mỗi lớp Mixing Blocks được áp dụng một chập 3×3, tuy nhiên áp dụng stride $= 2$ cho chiều cao h để giảm một nửa kích thước chiều cao và áp dụng stride $= 1$ cho chiều rộng w để giữ nguyên kích thước theo chiều rộng. Ngay sau đó, Layer Normalization được áp dụng để đảm bảo giá trị đặc trưng được chuẩn hóa, giúp cải thiện hiệu quả hội tụ trong quá trình huấn luyện. Kết quả tạo thành một đầu ra CC_i có kích thước $\frac{h}{2} \times w \times D_i$.

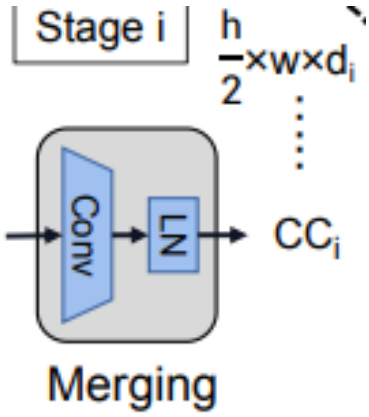


Figure 5: Quá trình Merging

Hoạt động gộp giảm một nửa chiều cao trong khi vẫn chiều rộng được giữ không đổi. Nó không chỉ giảm chi phí tính toán mà còn xây dựng cấu trúc phân cấp tùy chỉnh theo văn bản. Thông thường, hầu hết văn bản hình ảnh xuất hiện theo chiều ngang hoặc tương đương với chiều ngang. Việc nén kích thước chiều cao có thể thiết lập và biểu diễn nhiều tỷ lệ cho từng ký tự, đồng thời không ảnh hưởng đến bố cục bản vá theo chiều rộng.

4. Combining and Prediction:

Đến trước giai đoạn cuối cùng này, ta đang có CC_2 có kích thước là $\frac{H}{16} \times \frac{W}{4} \times D_2$. Sau khi đi qua khối trộn, Merging sẽ không được áp dụng nữa, thay vào đó là Combining hay kết hợp. Lúc này Pooling Layer sẽ được sử dụng để đưa chiều cao về 1. Kĩ thuật Dropout được áp dụng tiếp tục để đưa ra đầu ra $C : 1 \times \frac{W}{4} \times D3$. Đầu ra từ Combining chứa đặc trưng ngữ nghĩa nhưng chưa thể trực tiếp ánh xạ thành ký tự. Fully-Connected Layer là bước cuối cùng kết nối và chuyển đổi các đặc trưng đó thành thông tin rõ ràng để dự đoán ký tự. Sau khi kết thúc Encoding với Fully-Connected Layer, quá trình Decode sẽ được thực hiện với CTC Loss, và đầu ra dạng text sẽ kết thúc toàn bộ quá trình.

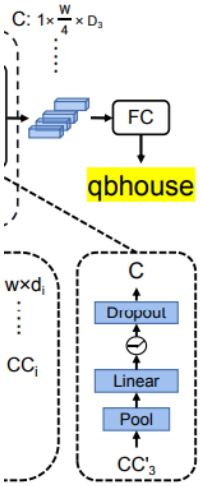


Figure 6: Quá trình Combining

So với hoạt động hợp nhất, việc sử dụng lớp Combining ở cuối thay vì Merging giúp tránh việc sử dụng lớp tích chập đối với các ma trận đặc trưng quá nhỏ gây mất đặc trưng ban đầu, đặc biệt khi chiều cao chỉ là 2.

5. Sự biến đổi mô hình:

Có một số siêu tham số trong SVTR, bao gồm độ sâu của kênh và số lượng đầu ở mỗi giai đoạn, số lượng khối trộn và sự hoán vị của chúng. Bằng cách thay đổi chúng, có thể thu được các kiến trúc SVTR với các dung lượng khác nhau là SVTR-T (Nhỏ), SVTR-S (Nhỏ), SVTR-B (Cơ sở) và SVTR-L (Lớn).

Models	$[D_0, D_1, D_2]$	$[L_1, L_2, L_3]$	Heads	D_3	Permutation	Params (M)	FLOPs (G)
SVTR-T	[64,128,256]	[3,6,3]	[2,4,8]	192	$[L]_6[G]_6$	4.15	0.29
SVTR-S	[96,192,256]	[3,6,6]	[3,6,8]	192	$[L]_8[G]_7$	8.45	0.63
SVTR-B	[128,256,384]	[3,6,9]	[4,8,12]	256	$[L]_8[G]_{10}$	22.66	3.55
SVTR-L	[192,256,512]	[3,9,9]	[6,8,16]	384	$[L]_{10}[G]_{11}$	38.81	6.07

Figure 7: Cấu hình chi tiết.

III. Thực nghiệm của tác giả:

Đối với việc nhận diện tiếng Anh, nhóm tác giả đã sử dụng hai bộ dữ liệu tổng hợp phổ biến là MJSynth (MJ) và SynthText (ST) để huấn luyện mô hình nhận dạng văn bản; sử dụng 6 bộ dữ liệu đánh giá công khai: ICDAR 2013 (IC13); Street View Text (SVT); IIIT5K-Words (IIIT); ICDAR 2015 (IC15); Street View Text-Perspective (SVTP); CUTE80 (CUTE) để kiểm tra.

Trong khi đó, huấn luyện mô hình tiếng Trung sẽ sử dụng bộ dữ liệu cảnh Trung Quốc (Chinese Scene Dataset) với 509164 ảnh huấn luyện, 63645 ảnh xác thực và 63,646 ảnh kiểm tra.

method		English regular			English unregular			Chinese	Params	Speed
		IC13	SVT	IIIT5k	IC15	SVTP	CUTE	Scene	(M)	(ms)
Lan-free	CRNN[Shi *et al.*, 2017]	91.1	81.6	82.9	69.4	70.0	65.5	53.4	8.3	6.3
	Rosetta[Borisyuk *et al.*, 2018]	90.9	84.7	84.3	71.2	73.8	69.2	-	44.3	10.5
	SRN*[Yu *et al.*, 2020]	93.2	88.1	92.3	77.5	79.4	84.7	-	-	-
	PREN*[Yan *et al.*, 2021]	94.7	92.0	92.1	79.2	83.9	81.3	-	29.1	40.0
	ViTSTR[Atienza, 2021]	93.2	87.7	88.4	78.5	81.8	81.3	-	85.5	11.2
	ABINet*[Fang *et al.*, 2021]	94.9	90.4	94.6	81.7	84.2	86.5	-	23.5	50.6
	VST*[Tang *et al.*, 2022]	95.6	91.9	95.6	82.3	87.0	91.8	-	-	-
Lan-aware	ASTER[Shi *et al.*, 2019]	-	89.5	93.4	76.1	78.5	79.5	54.5	27.2	-
	MORAN[Luo *et al.*, 2019]	-	88.3	91.2	-	76.1	77.4	51.8	28.5	-
	NRTR[Sheng *et al.*, 2019]	94.7	88.3	86.5	-	-	-	-	31.7	160
	SAR[Li *et al.*, 2019]	91.0	84.5	91.5	69.2	76.4	83.5	62.5	57.5	120
	AutoSTR[Zhang *et al.*, 2020]	-	90.9	94.7	81.8	81.7	84.0	-	10.4	207
	SRN[Yu *et al.*, 2020]	95.5	91.5	94.8	82.7	85.1	87.8	60.1	54.7	25.4
	PREN2D[Yan *et al.*, 2021]	96.4	94.0	95.6	83.0	87.6	91.7	-	-	-
	VisionLAN[Wang *et al.*, 2021]	95.7	91.7	95.8	83.7	86.0	88.5	-	32.8	28.0
	ABINet[Fang *et al.*, 2021]	**97.4**	93.5	96.2	86	89.3	89.2	-	36.7	51.3
	VST[Tang *et al.*, 2022]	96.4	**93.8**	**96.3**	85.4	88.7	**95.1**	-	64.0	-
Ours	SVTR-T (Tiny)	96.3	91.6	94.4	84.1	85.4	88.2	67.9	6.03	4.5
	SVTR-S (Small)	95.7	93.0	95.0	84.7	87.9	92.0	69.0	10.3	8.0
	SVTR-B (Base)	97.1	91.5	96.0	85.2	**89.9**	91.7	71.4	24.6	8.5
	SVTR-L (Large)	97.2	91.7	**96.3**	**86.6**	88.4	**95.1**	**72.1**	40.8	18.0

Figure 8: Bảng so sánh kết quả. *Lan free:* Là mô hình không dùng kiến thức từ Language Model. *Lan aware:* Là mô hình có sử dụng Language Model.

STVR luôn cho được ra được tham số và tốc độ tốt hơn so với các mô hình khác. Đối với chỉ số Accuracy cũng cho được các con số lớn hơn.

IV. Thực nghiệm của nhóm:

1. Mục tiêu

Mục tiêu của thực nghiệm là sử dụng mô hình STVR được giới thiệu ở trên để thực hiện nhận dạng văn bản tiếng Việt từ hình ảnh. Thực nghiệm bao gồm quá trình tải dữ liệu, thiết lập môi trường, và chạy inference để kiểm tra hiệu quả của mô hình trên tập dữ liệu.

2. Dữ liệu và môi trường:

2.1 Dữ liệu

Dữ liệu được sử dụng bao gồm:

- **charset_official_ver3:** là một file chứa các kí tự bao gồm: 52 kí tự trong bảng chữ cái tiếng Anh bao gồm viết in hoa và viết thường; các chữ cái có thể mang dấu trong tiếng Việt ở dạng in hoa: À, Á, Â, Ã, È,..... và tương ứng ở dạng chữ thường à, á, â, ...; 1 dấu gạch nối cho các tên riêng nước ngoài phiên âm sang tiếng Việt (dấu -), tổng cộng là 373 kí tự.

- **fold0**: là một folder chứa các folder nhỏ hơn và các file ảnh, bao gồm: folder **train** chứa 92700 file ảnh huấn luyện định dạng jpg và folder **valid** chứa 10300 ảnh kiểm tra định dạng jpg, tổng cộng là 10; 2 file text (.txt) là **train_gt_fold0** và **valid_gt_fold0** lần lượt là tệp chứa các nhãn đúng cho dữ liệu huấn luyện và kiểm tra. Dữ liệu này được lấy từ cuộc thi BKAI SoICT Hackathon 2023 thuộc track nội dung **Vietnamese Handwritten Text Recognition**, dữ liệu là tập Training của cuộc thi gồm 103000 ảnh.

- **rec_svtr_large_10local_11global_stn_en.yml:** Tệp cấu hình này được sử dụng để thiết lập quá trình huấn luyện và kiểm thử mô hình nhận dạng ký tự dựa trên mô hình SVTR.

- **weight_svtr.pdparams:** chứa trọng số đã được huấn luyện của mô hình. Đây là các tham số đã được tối ưu trong quá trình huấn luyện, bao gồm trọng số và bias của các lớp trong mạng nơ-ron.

2.2. Môi trường:

Công cụ và Thư viện Công cụ: Google Colab với GPU hỗ trợ.
PaddlePaddle: Nền tảng học sâu mã nguồn mở của Trung Quốc.
Thư viện: Các thư viện chính bao gồm:
PaddleOCR: Dùng để xử lý nhận dạng ký tự trong ảnh.
SciPy: Dùng cho các phép toán khoa học và số học trong quá trình xử lý dữ liệu
Streamlit: Tạo ứng dụng web đơn giản hiển thị kết quả

3. Quy trình thực hiện:

3.1. Tải và giải nén dữ liệu:

Sử dụng thư viện gdown để tải các tệp cấu hình và trọng số từ Google Drive.
Giải nén tệp fold0.zip vào thư mục /content/fold0 bằng thư viện zipfile.

3.2. Thiết lập môi trường:

Quá trình huấn luyện và kiểm thử được thực hiện trong môi trường Google Colab, với hỗ trợ GPU giúp tăng tốc quá trình huấn luyện.

3.3. Chạy inference:

Sau khi đã chuẩn bị dữ liệu và thiết lập môi trường, bước tiếp theo là thiết lập mô hình nhận diện chữ viết tay. Mô hình STVR được sử dụng để xử lý ảnh và nhận diện các ký tự viết tay.

Tệp cấu hình
Mô hình nhận diện chữ viết tay được cấu hình qua tệp rec_svtr_large_10local_11global_stn_en.yml, trong đó các tham số quan trọng như số lượng lớp, kích thước batch, số epoch và các yếu tố khác được thiết lập. Trọng số mô hình ban đầu được nạp từ tệp weight_svtr.pdparams.

Quá trình huấn luyện sử dụng dữ liệu huấn luyện từ thư mục train, với mục tiêu tối ưu hóa mô hình sao cho có thể nhận diện đúng các ký tự từ ảnh chữ viết tay. Các tham số huấn luyện, bao gồm tỷ lệ học (learning rate) và số epoch, sẽ được điều chỉnh để đạt hiệu suất tốt nhất.

3.4. Kết quả:

Sử dụng thư viện Streamlit tạo một web app đơn giản nhận diện chữ viết tay

V. Hướng cải thiện:

Mô hình SVTR hiện tại đã đạt được hiệu quả tốt trong việc nhận dạng văn bản, tuy nhiên vẫn còn một số thách thức cần giải quyết, bao gồm:

- **Tốc độ suy luận (inference speed):** Cần được tối ưu hơn để phù hợp với các ứng dụng thời gian thực.
- **Khả năng mô hình hóa ngữ cảnh ngôn ngữ (linguistic context modeling):** Đặc biệt quan trọng để nâng cao độ chính xác trong nhận dạng văn bản phức tạp.

Nhóm đề xuất xây dựng thêm 1 Module áp dụng cho mô hình:

- **Module thống kê tần suất ký tự (Statistical Module):** Thu thập và học các thông tin thống kê về tần suất xuất hiện của ký tự trong các bối cảnh khác nhau. Hỗ trợ việc dự đoán ký tự dựa trên dữ liệu ngữ cảnh. Ví dụ: Trong một số bối cảnh, ký tự "e" có khả năng xuất hiện cao hơn "z", thông tin này sẽ được sử dụng để định hướng dự đoán.

Tất nhiên việc áp dụng thêm module sẽ ảnh hưởng đến chi phí tính toán. Chính vì vậy, nhóm cũng có một số đề xuất về hướng triển khai:

- Module thống kê tuần suất kí tự: thay vì tính toán tần suất trong thời gian thực, có thể tính toán thông tin tần suất từ trước trong quá trình huấn luyện và lưu trữ kết quả tần suất cho các ký tự. Điều này giúp giảm chi phí tính toán trong giai đoạn dự đoán.

Việc chấp nhận tăng thêm một ít về chi phí để tăng độ chính xác tùy thuộc vào hoàn cảnh thực tế, nếu chi phí không tăng quá nhiều thì việc hi sinh thêm một chút bộ nhớ để tăng độ chính xác là hoàn toàn chấp nhận được.

VI. Tổng kết:

Nhóm đã trình bày việc nhận diện chữ viết tay tiếng Việt thông qua mô hình SVTR. Thông qua quá trình thực nghiệm, mô hình đã cho thấy độ chính xác cao. nhiệm vụ nhận dạng có thể được thực hiện bằng cách sử dụng một mô hình trực quan duy nhất, tận hưởng các ưu điểm về độ chính xác, hiệu quả và tính linh hoạt giữa các ngôn ngữ. SVTR với các khả năng khác nhau cũng được thiết kế để đáp ứng các nhu cầu ứng dụng đa dạng, từ việc nhận diện ảnh viết tay chữ Trung Quốc và chữ tiếng Anh cho đến nhận diện chữ viết tay Tiếng Việt do nhóm đã thực hiện. Điều này chứng tỏ tiềm năng lớn của SVTR trong việc xây dựng các hệ thống nhận dạng chữ viết tay đa ngôn ngữ, góp phần vào việc số hóa tài liệu và tự động hóa các quy trình làm việc.

Việc nhận dạng chữ viết tay tiếng Việt không chỉ đơn thuần là một thành tựu công nghệ mà còn mang ý nghĩa to lớn đối với sự phát triển của xã hội. Công nghệ này đóng vai trò quan trọng trong việc bảo tồn di sản văn hóa, tự động hóa quy trình làm việc, hỗ trợ người khuyết tật và mở ra nhiều ứng dụng thông minh mới. Nhờ có khả năng chuyển đổi chữ viết tay thành văn bản số, chúng ta có thể dễ dàng lưu trữ, tìm kiếm và chia sẻ thông tin, đồng thời giảm thiểu lỗi sai trong quá trình nhập liệu. Việc nhận dạng chữ viết tay tiếng Việt không chỉ giúp nâng cao hiệu quả làm việc mà còn góp phần thúc đẩy quá trình số hóa dữ liệu, tạo tiền đề cho sự phát triển của các ứng dụng trí tuệ nhân tạo trong tương lai.

Tài liệu tham khảo:

- Scene Text Recognition with Single Visual Model
- Attention is all you need
- SVTR - Lời giải hoàn hảo cho bài toàn OCR?