BÁO CÁO ĐỒ ÁN CS114.L22.KHCL

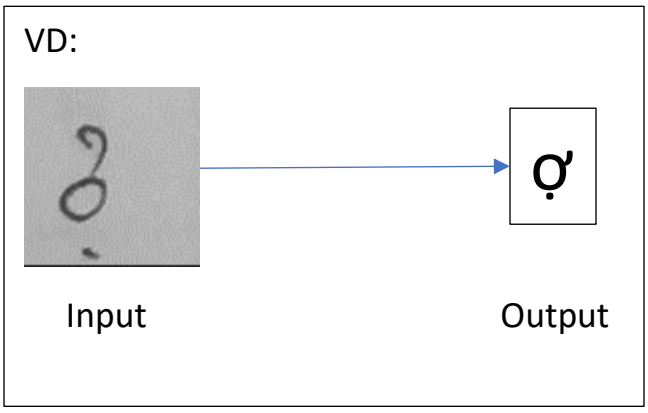
Đề tài: Phân loại chữ cái tiếng việt viết tay

Nội dung báo cáo:

- Nhận diện bài toán
- Thu thập, xử lí dữ liệu
- Tìm kiếm, xây dựng các model
- Thử cải tiến với mô hình CNN

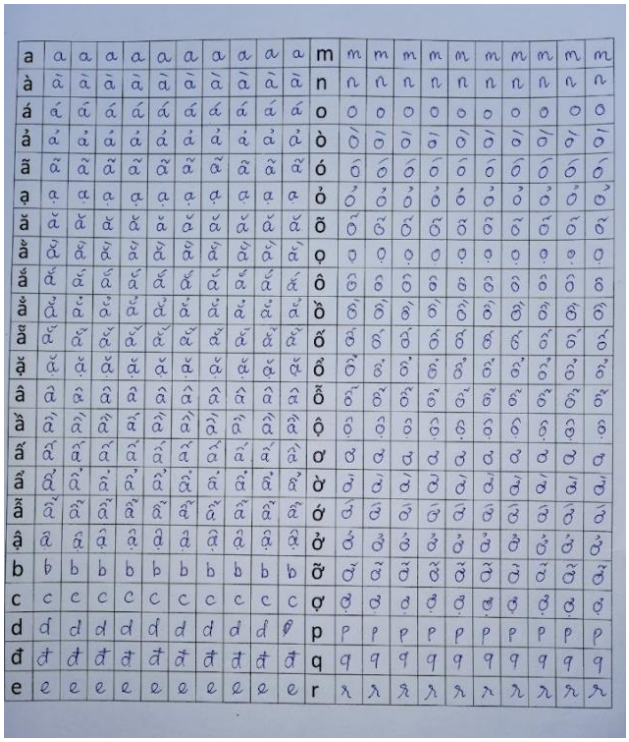
I. Nhận diện bài toán

- Bài toán phân biệt chữ cái tiếng việt viết tay thuộc loại bài toán classification
- Có tổng cộng 89 class bao gồm các kí tự kèm với các dấu thanh (ngang, sắc, huyền, hỏi, ngã, nặng)
- Input: Hình ảnh chữ cái tiếng việt được viết bằng tay
- Output: string chữ xuất hiện trong tấm ảnh

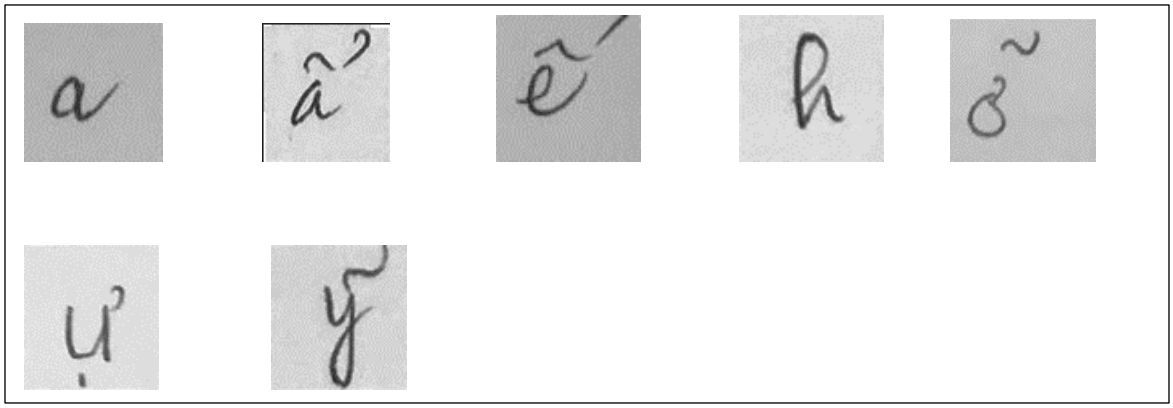


II. Thu thập và xử lí dữ liệu

- Thu thập dữ liệu:
  - Nhóm đã thu thập được khoảng gần 25000 ảnh cho 89 class, dữ liệu được lấy từ người than, bạn bè của các thành viên trong nhóm ở các tỉnh Gia Lai, Quảng Ngãi
  - Nhóm có kết hợp với nhóm của bạn Trần Vĩ Hào để làm chung tập train
  - Cách thu thập dữ liệu :
    Mỗi chữ cái sẽ được viết vào một tờ giấy A4 có kẻ sẵn từng ô như hình dưới



Sau đó nhóm có tham khảo cách detect grid ở đường link:
https://stackoverflow.com/questions/59182827/how-to-get-the-cells-of-a-sudoku-grid-with-opencv
Sau khi detect thu được một vài hình như sau:



- Phân chia dữ liệu:
  - Các dữ liệu train và test được phân chia riêng biệt.Khoảng 20000 dữ liệu sử dụng để làm tập train và ~ 5000 dữ liệu sử dụng để làm tập test
  - Nhóm đã tìm hiểu và sử dụng phương pháp "K-Fold cross validation" để có thể đánh giá model hiệu quả hơn khi có ít dữ liệu
  - Với mỗi tấm ảnh sau khi đã xử lí sẽ được lưu thành 1 dòng trong file csv để thuận tiện cho việc train model
- Xử lí dữ liệu:
  - Mỗi tấm ảnh sẽ được đưa về dưới dạng ảnh nhị phân
  - Tất cả các ảnh sẽ được đưa về cùng một kích thước (16x20)
  - Mỗi tấm ảnh sẽ được xử lí theo hai cách:
    + Đưa tấm ảnh về mảng 1 chiều
    + Áp dụng hog (histogram of oriented gradient)

Một số ảnh sau khi đã xử lí qua tất cả các bước đã nêu ở trên

III. Tìm kiếm xây dựng các model

Với bài toán này nhóm đã sử dụng 6 thuật toán để thử nghiệm:

+ SVM (kernel = ['rbf', 'poly', 'linear'] , C = 7)

+ Navie Bayes

+ Logistic regression (C=8)

+ MLP Classification (hidden_layer_sizes = (320,320,320), max_iter = 500)

+ K-Neightbors

+ Random Forest

Kết quả thu được

1. Bộ data thường (flatten vector)

	SVM			LOGISTIC REGRESSION	NAIVE BAYES			(MLP) CLASSIFICATION	K-Neightbors	Random Forest
	SVM (kernel='linear')	SVM (kernel='rbf)	SVM (kernel='poly)	Logistics Regression	Multinomial Naive Bayes	Gaussian Naive Bayes	Bernoulli Naive Bayes			
Val	73.6%	83.5%	78.9%	63.3%	44.0%	41.6%	28.8%	81.3%	60.9%	75.4
Test	42.7%	51.9%	46.7%	39.6%	32.7%	29.0%	19.0%	54.5%	30.5%	44.3%

Nhận xét: Từ bảng số liệu nhóm đã thống kê có thể thấy được 2 mô hình SVM(kernel = 'rbf') và MLP Classification là hai mô hình cho kết quả tốt hơn hẳn so với các mô hình còn lại

2. Bộ data sử dụng trích xuất đặc trưng hog(hog + fatten vector)

Sau khi xem xét kết quả thu được với bộ data thường nhóm quyết định loại bỏ một vài mô hình: SVM(kernel = 'linear') và Navie bayes

	SVM		LOGISTIC REGRESSION	(MLP) CLASSIFICATION	K-Neightbors	Random Forest
	SVM (kernel='rbf)	SVM (kernel='poly)				
Val	84.1%	79.8%	60.1%	83.0%	74.1%	78.9%
Test	55.7%	53.5%	42.7%	57.3%	48.6%	49.7%

Nhận xét: Có thể thấy được các mô hình đều cho kết quả tốt hơn so với bộ data thường. accuracy trên tập validation đều tăng từ khoảng 2-3%, trên tập test là khoảng 3-6%. Riêng với mô hình K-Neightbors cả tập validation và test đều tăng khá nhiều (lần lượt là 14% và 18%)

Nhận xét chung:

- Các mô hình đều xuất hiện tình trạng underfit

- Các mô hình sau khi huấn luyện có độ chính xác chưa thực sự cao, Giá trị accuracy cao nhất là 57.3%
- Các class như ‘a’, ‘b’, ‘c’, ‘m’, ‘p’, ‘q’ thường cho độ chính xác khá tốt

VD: một vài kết quả thu được từ các mô hình khác nhau

	Precission	recall	f1-score	suport
a	0.80	1.00	0.89	39
b	0.86	0.91	0.89	47
m	0.90	0.80	0.85	56
p	0.98	0.75	0.85	65
q	0.88	0.79	0.83	56

- Các class còn lại ở các mô hình khác nhau cho các giá trị khác nhau. Không thấp cùng nhau hoặc cao cùng nhau tại các mô hình
- Nguyên nhân chủ yếu khiến mô hình cho kết quả chưa đủ tốt:
  + Dữ liệu có vẻ còn khá ít với khoảng 170 ảnh cho mỗi lớp
  + Số class quá nhiều. Các mô hình máy học không phù hợp
  + Sự hiểu biết của nhóm về các mô hình còn chưa tốt. Điều chỉnh các parameter trong các mô hình chưa phù hợp dẫn tới hiệu quả mô hình chưa thực sự tối ưu

- Hướng cải thiện:
  + Tìm hiểu và sử dụng bóc tách ảnh nhỏ cho từng lớp theo cách khác để có thể tránh mất mát dữ liệu quá nhiều (Dự kiến ban đầu của nhóm sẽ là khoảng 250-300 ảnh cho mỗi lớp nhưng kết quả chỉ thu về được khoảng 170 ảnh cho mỗi lớp)
  + Tìm kiếm thêm các mô hình khác phức tạp hơn và chuyên dụng cho các bài toán xử lí hình ảnh nhiều lớp
  + Tìm hiểu các phương pháp để cải thiện chất lượng data, trích xuất đặc trưng và tiền xử lí hiệu quả hơn
  + Tìm hiểu kỹ hơn về các mô hình, cách điều chỉnh các parameter để đem lại kết quả tối ưu

IV. Cải tiến: Convolutional neural network (CNN) in keras

V. Ứng dụng và hướng phát triển