DEEP LEARNING TRONG PHÂN LOẠI FONTS CHỮ

Trần Thị Mỹ Linh*, Dương Thị Hồng Hạnh*, Nguyễn Trọng Ân*, Hà Như Chiến*,
Đỗ Trọng Hợp†, Lưu Thanh Sơn†
Email: *{18520999,18520711, 18520434, 18520527}@gm.uit.edu.vn
†{hopdt, sonlt}@uit.edu.vn

I. Giới thiệu

Trong lĩnh vực thiết kế, có thể nói font chữ đóng một vai trò vô cùng quan trọng. Mỗi font khi ra đời đều mang trong mình những nét đặc trưng riêng, thể hiện các tính chất riêng mà người thiết kế muốn gửi gắm. Nó không đơn giản chỉ là một công cụ để truyền tải nội dung đến với độc giả mà còn là một phần giá trị của bản thiết kế. Việc lựa chọn đúng font chữ, kết hợp hài hòa tính chất của các ấn phẩm thiết kế với tính chất của font sẽ góp phần to lớn làm tăng hiệu ứng truyền tải thông điệp của người thiết kế đến với người xem. Tuy nhiên, thực tế có thể thấy rằng để lựa chọn font chữ phù hợp trong thiết kế tốn không ít thời gian, chính vì vậy chúng tôi mong muốn xây dựng nên một mô hình gợi ý font chữ phù hợp dựa trên các đặc tính của bản thiết kế. Để thực hiện bài toán đặt ra, ta cần giải quyết hai bài toán nhỏ: phân loại font chữ và gợi ý font phù hợp với bản thiết kế. Trước tiên, chúng tôi bước đầu sẽ thực hiện xây dựng mô hình phân loại font một cách tự động dựa trên ảnh của nó. Ở bài báo cáo này, thực hiện phân loại 3 fonts chữ gồm Old School, Handwritten-Fancy và Horror-Fancy với các mô hình Deep Learning, mà cụ thể là áp dụng hai kiến trúc LeNet và AlexNet, kèm theo những điều chỉnh khác. Các bộ dữ liệu được sử dụng trong bài báo cáo này được xây dựng theo trên 3 hướng:

- Dataset 1: ảnh của tất cả các chữ cái La Tinh (gồm cả chữ thường và chữ hoa) theo 3 loại font.
  => Xây dựng mô hình một cách tổng quát trên tất cả ký tự.
- Dataset 2: ảnh của 3 loại font được phân thành 5 loại chữ cái riêng biệt(A,B,C,D,E).Các chữ cái còn lại trong bảng chữ cái sẽ được tiến hành tương tự trong tương lai.
  => Xây dựng mô hình phân loại 3 fonts độc lập cho từng chữ cái riêng.
- Dataset 3: ảnh của 5 chữ cái A,B,C,D,E theo 3 loại font.
  => Xây dựng mô hình tổng quát phân loại 3 fonts nhưng ở quy mô nhỏ hơn trường hợp.

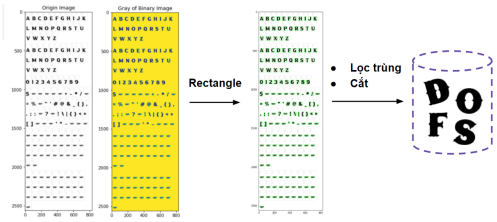
Tất cả kết quả của quá trình thực nghiệm đều được ghi lại và cho thấy rằng, mô hình phân loại font chữ dựa trên bộ dữ liệu được gộp 5 ký tự A,B,C,D,E (dataset 3) với kiến trúc AlexNet mang lại kết quả tốt nhất hiện tại.

II. Bộ dữ liệu

A. Phương pháp thu thập

1) Thu thập dữ liệu không có background:

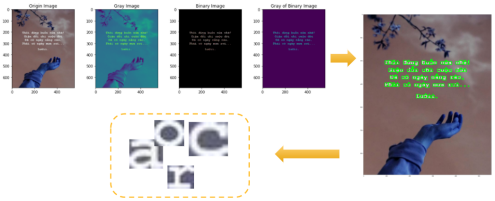
- Nguồn thu thập: Dafont.com.
- Phương pháp: Từ các thể loại font có sẵn, tiến hành thu thập các bảng kí tự của từng font, sử dụng các kỹ thuật có sẵn được tích hợp trong OpenCV để cắt thành từng kí tự riêng biệt (Hình 1).



Hình 1: Quy trình cắt ảnh từ bảng kí tự thu được từ Dafont

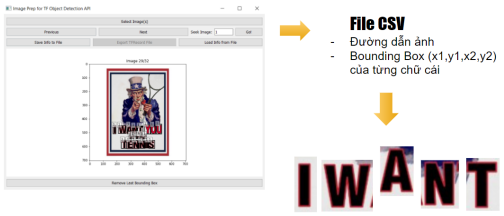
2) Thu thập dữ liệu có background:

- Nguồn thu thập: Pinterest, Google.
- Phương pháp: Thu thập các ảnh, ấn phẩm từ các trang web, có 2 loại ảnh cần xử lý:
  - Background đơn giản: sử dụng các kỹ thuật có sẵn được tích hợp trong OpenCV để cắt thành từng kí tự riêng biệt (Hình 2).



Hình 2: Ví dụ xử lý ảnh có Background đơn giản

  - Background phức tạp: sử dụng công cụ hỗ trợ vẽ boundingbox, lưu vị trí chữ về file csv, dùng code để cắt thành kí tự riêng biệt (Hình 3).



Hình 3: Ví dụ xử lý ảnh có Background phức tạp

3) Tạo dữ liệu thủ công:

- Công cụ hỗ trợ: Adobe Illustator.
- Phương pháp: Từ các loại font có sẵn, tiến hành tạo dữ liệu với màu chữ, background khác nhau, Hình 4 là một số dữ liệu mẫu.

Hình 4: Ví dụ dữ liệu tạo thủ công

B. Thông tin các bộ dữ liệu

Dữ liệu là những ảnh màu của một kí tự chữ La Tinh tương ứng với font chữ.

- Dataset 1: Gồm 3634 ảnh có kích thước khác, được chia thành 3 file tương ứng với 3 phân lớp (3 loại font chữ: Handwritten-Script, Horror-Fancy, Oldschool-Fancy). Với sự hỗ trợ của thư viện splitfolders, thực hiện chia dữ liệu thành 2 file train và val theo tỷ lệ 8:2. Thông tin chi tiết được thể hiện tại bảng bên dưới.

Bảng I: Thống kê trên bộ dữ liệu 1

	Train (Images)	Val (Images)	Total (Images)
Handwritten-Script	566	142	718
Horror-Fancy	1609	403	2012
Old School-Fancy	723	181	904
Total (by train,val)	2898	726	3634

Xem thêm dataset tại: Dataset 1.
**Nhận xét: Dữ liệu xuất hiện tình trạng mất cân bằng, lớp Horror-Fancy có số lượng dữ liệu gấp hơn 2 lần so với Handwritten-Script và Old-School
=> Có thể gây ra sai lệch trong quá trình huấn luyện và dự đoán.

- Dataset 2: Gồm 21246 ảnh có cùng kích thước 224x224 pixel, được chia thành 5 file tương ứng với 5 loại chữ cái (A,B,C,D,E), trong mỗi file sẽ gồm ảnh chữ cái tương ứng của 3 loại font (Handwritten-Script, Horror-Fancy, Oldschool-Fancy). Thông tin chi tiết được thể hiện tại bảng bên dưới.

Bảng II: Thống kê trên bộ dữ liệu 2

		Train Val (Images)	Test (Images)	Total by Character (Images)
A	A-Handwritten-Script	1137	300	4078
	A-Horror-Fancy	1091	300	
	A-Oldschool-Fancy	950	300	
	Total A	**3178**	**900**	
B	B-Handwritten-Script	1200	300	4298
	B-Horror-Fancy	1200	300	
	B-Oldschool-Fancy	998	300	
	Total B	**3398**	**900**	
C	C-Handwritten-Script	1200	300	4290
	C-Horror-Fancy	1200	300	
	C-Oldschool-Fancy	990	300	
	Total C	**3390**	**900**	
D	D-Handwritten-Script	1200	300	4290
	D-Horror-Fancy	1200	300	
	D-Oldschool-Fancy	990	300	
	Total D	**3390**	**900**	
E	E-Handwritten-Script	1200	300	4290
	E-Horror-Fancy	1200	300	
	E-Oldschool-Fancy	990	300	
	Total E	**3390**	**900**	

Xem thêm dataset tại: Dataset 2.

- Dataset 3: Dựa trên dataset 2, ta thực hiện thao tác gộp dữ liệu các loại chữ cái (A,B,C,D,E) theo loại font tương ứng trên train và val. Dataset thu được gồm 21246 ảnh và có thông tin chi tiết được thể hiện ở bảng bên dưới:

Bảng III: Thống kê trên bộ dữ liệu 3

	Train_Val-Merge-ABCDE	Test-Merge-ABCDE
Horror-Fancy	5937	1500
Oldschool-Fancy	4918	1500
Handwritten-Script	5892	1500
Total	16746	4500

Xem thêm dataset tại: Dataset 3.

III. Thách thức

A. Thách thức trên Dataset 1

- Dữ liệu thuộc cùng 1 class có nhiều hình dạng, kích thước khác nhau => khó khăn trong việc trích xuất đặc trưng để học (hình 5a,b).
- Cùng là hình ảnh của cùng 1 loại chữ nhưng đặc trưng lại khác nhau (hình 5c).

(a) Ký tự có hình dạng khác nhau (b) Kích cỡ ảnh đa dạng

(c) Cùng thuộc class 'Horror-Fancy" nhưng chữ V khác biệt nhau rõ ràng

Hình 5: Đặc điểm dữ liệu.

- Dữ liệu là background trắng (không background), chữ đen => có thể gây khó khăn khi dự đoán ảnh kí tự có background trong thực tế.
- Dữ liệu còn khá ít mà có đến 24 chữ và 10 chữ số => mỗi kí tự được học ít => dễ nhầm lẫn.

B. Thách thức trên Dataset 2 và Dataset 3

Đã khắc phục được hầu hết các lỗi trên, tuy nhiên vẫn tồn tại những sự khác biệt nhất định về hình dạng (hình 6a) cũng như những hình ảnh có background quá phức tạp (hình 6b).

(a) Chữ cùng loại bị biến dạng (b) Background quá phức tạp

Hình 6: Đặc điểm dữ liệu.

IV. Phương pháp tiếp cận

A. Bộ dữ liệu đầu vào

Tập Train và Val được chia theo tỷ lệ 8:2 từ bộ dữ liệu Train-Val đã được mô tả ở phần Dữ liệu.
Tập Test được chuẩn bị riêng.

B. Tiền xử lý dữ liệu

Sử dụng ImageDataGenerator từ thư viện keras.preprocessing.image. Mục đích là chuyển ảnh về kích cỡ mong muốn (28x28 hoặc 64x64) và chuẩn hóa các giá trị pixels của ảnh về khoảng 0 đến 1 bằng cách chia từng pixel cho 255.

Áp dụng kiến trúc LeNet và AlexNet kèm theo những điều chỉnh cho phù hợp với bộ dữ liệu.

Kiến trúc dựa trên mô hình AlexNet:

1) Input: 28x28x3
2) Conv2D: 32 filters with 3x3, S=1
3) BatchNomorlization()
4) MaxPool2D: 3x3 filters with S=1, padding='valid'
5) Conv2D: 32 filters with 3x3, S=1
6) BatchNomorlization()
7) MaxPool2D: 3x3 filters with S=1, padding='valid'
8) Conv2D: 64 filters with 3x3, S=1
9) Conv2D: 64 filters with 1x1, S=1
10) Conv2D: 256 filters with 1x1, S=1, padding='same'
11) MaxPool2D: 3x3 filters with S=2
12) Flatten()
13) Dense: 16
14) Dropout: 0.3
15) Dense: 3, activation='softmax'

Kiến trúc dựa trên LeNet:

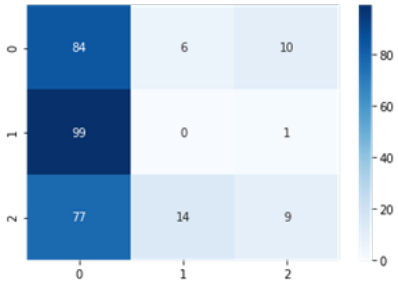
1) Input: 224x224x3
2) Conv2D: 32 filters with 2x2
3) MaxPool2D: 2x2 filters
4) Conv2D: 32 filters with 2x2
5) MaxPool2D: 2x2 filters
6) Conv2D: 64 filters with 2x2
7) MaxPool2D: 2x2 filters
8) Flatten()
9) Dense: 64
10) Dropout: 0.5
11) Dense: 3, activation='softmax'

Các lớp Conv2D và Dense đều dùng hàm kích hoạt 'relu', riêng lớp Dense cuối cùng dùng để phân loại 3 lớp nên dùng hàm kích hoạt là 'softmax'. Loss sử dụng '`categorical_crossentropy`'. Hàm tối ưu: Stochastic Gradient Descent(SGD) với learning rate = 0.001. Độ chính xác 'accuracy'. Batch size = 16, epochs = 10 cho mô hình dựa trên AlexNet và 20 cho mô hình dựa trên LeNet.

V. Kết Quả

A. Thử nghiệm lần 1

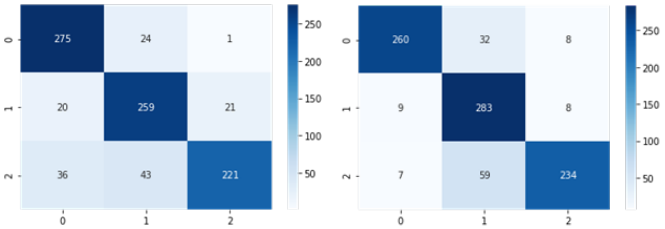
- Sử dụng toàn bộ dữ liệu của tất cả các chữ cái và số (Dataset 1) để huấn luyện.
- Bộ dữ liệu huấn luyện lúc này chưa có backgrounds phức tạp mà chỉ là nền trắng.
- Kết quả dự đoán có độ chính xác thấp, bị lệch về nhãn Handwritten-Fancy



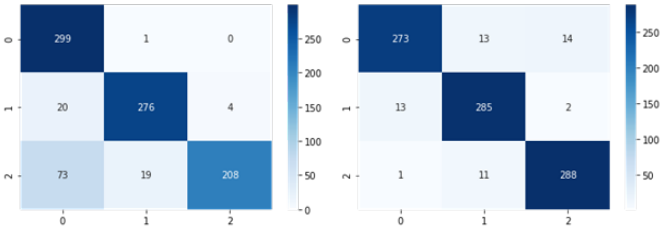
Hình 7: Confusion matrix của tập test

B. Thử nghiệm lần 2

- Sử dụng bộ dữ liệu có chứa backgrounds phức tạp (Dataset 2)
- Chia việc huấn luyện mô hình theo từng ký tự A, B, C, D, và E.
- Kết quả khả quan hơn nhiều nên hướng này sẽ được cân nhắc tiếp tục phát triển.



(a) Confusion matrix trên tập test của mô hình LeNet huấn luyện cho chữ A (b) Confusion matrix trên tập test của mô hình AlexNet huấn luyện cho chữ A

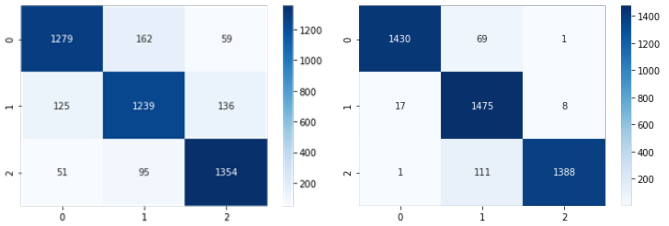


(a) Confusion matrix trên tập test của mô hình LeNet huấn luyện cho chữ B (b) Confusion matrix trên tập test của mô hình AlexNet huấn luyện cho chữ B

C. Thử nghiệm lần 3

Sử dụng bộ dữ liệu có backgrounds phức tạp của lần Thử nghiệm 2 để huấn luyện mô hình. Tuy nhiên, bộ dữ liệu trên sẽ được gộp tất

cả các chữ cái thay vì huấn luyện mô hình cho từng chữ cái như ở lần Thử nghiệm 2 (Dataset 3)



(a) Confusion matrix trên tập test của mô hình LeNet

(b) Confusion matrix trên tập test của mô hình AlexNet

D. So sánh

Bảng IV: Bảng so sánh

	LeNet		AlexNet	
Characters	F1-score	Accuracy	F1-score	Accuracy
A	84%	84%	86%	86%
B	87%	87%	93%	93%
C	92%	92%	95%	95%
D	87%	87%	89%	89%
E	92%	92%	93%	93%
ABCDE	86%	86%	95%	95%

*Nhận xét:

- Nhìn chung kết quả thu được sau khi thực nghiệm ở tất cả các trường hợp đều cho kết quả khá tốt trên cả hai mô hình kiến trúc LeNet và AlexNet, tuy nhiên AlexNet có phần nhỉnh hơn so với LeNet.
- Mô hình tốt nhất hiện tại ta thu được là mô hình phân loại chữ cái C trên Dataset 2 với kiến trúc AlexNet (95%) và mô hình phân loại trên dữ liệu gộp cả 5 loại chữ cái (Dataset 3) cũng với kiến trúc AlexNet và accuracy=95%.
- Việc độc lập hóa từng mô hình dự đoán riêng theo từng chữ cái cho các mô hình dự đoán có hiệu suất khác nhau. Trong đó, mô hình B, C, E cho kết quả dự đoán tốt hơn so vớ B, A.
  => Do số lượng chữ cái thực nghiệm khá ít (5/48-chữ hoa và chữ thường) nên dựa vào kết quả trên ta vẫn chưa xác định được chính xác phương pháp tốt nhất giữa: độc lập hóa mô hình dự đoán cho từng chữ cái và xây dựng mô hình tổng quát trên toàn bộ chữ cái.
  => Tiếp tục xây dựng dữ liệu và xây dựng mô hình trên toàn bộ bảng chữ cái để có kết quả chính xác nhất.

VI. Kết luận

Việc phân loại fonts chữ có là một vấn đề phức tạp bởi có quá nhiều chữ cái cũng như hình dạng thiết kế của các fonts chữ là rất đa dạng.

Backgrounds là một trong những yếu tố quan trọng ảnh hưởng đến kết quả phân loại. Việc tăng cường khối lượng ảnh huấn luyện cho các mô hình Deep Learning cùng với sự đa dạng backgrounds sẽ góp phần cải thiện chất lượng phân loại fonts chữ.

Để giảm bớt độ phức tạp cho bài toán thì việc chia làm nhiều mô hình cho từng chữ cái mang lại những kết quả khả quan ban đầu.Tuy nghiên, kết quả trên mô hình tổng quát 5 chữ cái cũng rất khả quan. Chính vì vậy, trong tương lai nên tiếp tục bổ sung dữ liệu của các loại chữ cái còn lại mới co thể xác định chính xác hiệu suất của hai phương pháp trên.

VII. Deploy

Video Deploy : Deploy model FONTS.

Hình 11: Quy trình deploy mô hình

Hình 12: Minh họa về deploy mô hình