



TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN – ĐHQG TP HCM

KHOA KHOA HỌC MÁY TÍNH

# Nhận diện biển báo giao thông với thuật toán Convolutional Neural Network (CNN)

Sinh viên: Trương Văn Nhất  
MSSV: 16521759

GVHD: TS. Lê Minh Hưng

## I. LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành nhất đến **TS Mai Tiên Dũng** đã quan tâm giúp đỡ và hướng dẫn rất tận tình cho nhóm em trong suốt quá trình học tập và thực hiện đồ án.

Có lẽ kiến thức là vô hạn và sự tiếp nhận kiến thức của em có những hạn chế nhất định. Do đó, trong quá trình làm bài tập và đồ án, chắc chắn không tránh khỏi những thiếu sót. Vì vậy, bản thân em rất mong nhận được những đóng góp ý kiến đến từ thầy để đồ án của em được hoàn thiện hơn.

Cuối cùng, em xin kính chúc thầy sức khỏe, hạnh phúc và thành công trên con đường sự nghiệp giảng dạy của mình.

Em xin chân thành cảm ơn.

## II. GIỚI THIỆU

Việc ứng dụng công nghệ thông tin để giải quyết các vấn đề trong lĩnh vực giao thông đang là một chủ đề nóng ở nước ta hiện nay. Vấn đề phát triển giao thông thông minh để giảm thiểu ùn tắc, tai nạn và tiết kiệm nguồn lực cũng được thảo luận trong nhiều hội thảo, diễn đàn như diễn đàn cấp cao Công nghệ Thông tin – Truyền thông Việt Nam (Vietnam ICT Summit), 2015. [1]

Trong đó, việc nhận dạng biển báo giao thông là một trong những vấn đề mang tính thách thức trong các hệ thống giám sát giao thông, hệ thống hỗ trợ tài xế, đặc biệt là xe tự hành.

Bài báo cáo trình bày phương pháp phát hiện và nhận dạng các biển báo giao thông đường bộ sử dụng kết hợp các kỹ thuật phân đoạn ảnh, phát hiện biên và phân tích hình dáng đối tượng để phát hiện vùng ứng viên có thể là biển báo giao thông. Sau đó, rút trích đặc trưng với thuật toán Convolutional

Neutral Network (CNN) và huấn luyện mạng Nơron nhân tạo để nhận dạng biển báo.

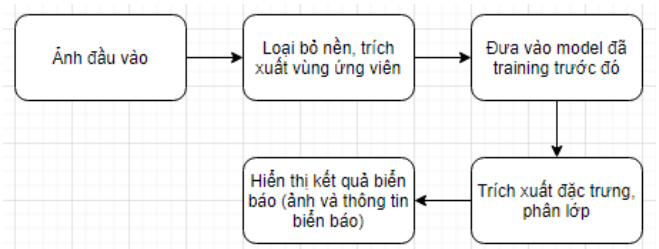
## III. TẬP DỮ LIỆU

Nhóm sử dụng bộ dữ liệu GTSRB - German Traffic Sign Recognition Benchmark gồm 43 loại biển báo của Đức [2].

Trong bài báo cáo này nhóm chỉ giới hạn nhận dạng 16 loại biển báo nhưng được gộp thành 4 nhóm chính đó là: Biển báo đường cấm, biển báo cấm đi ngược chiều, biển báo dừng, nhóm biển báo tốc độ tối đa cho phép.

Như vậy tập dữ liệu của chúng ta bao gồm khoảng 20.000 bức ảnh của 16 loại biển báo. Mỗi loại biển báo có khoảng hơn 1.000 tấm với đầy đủ kích thước khác nhau. Và tập dataset này sẽ được chia thành hai phần đó là training và validation theo tỷ lệ training/validation là 0.8/0.2.

## IV. QUY TRÌNH XỬ LÝ



Hình 1: Quy trình tổng quát của hệ thống phát hiện và nhận dạng biển báo giao thông đường bộ

**Bước 1:** Ảnh đầu vào sẽ được tiếp nhận thông qua input người dùng nhập vào.

**Bước 2:** Ảnh đầu vào sẽ được phân đoạn, loại bỏ nền nhằm làm nổi bật các đối tượng có thể là biển báo giao thông, biên

của các đối tượng này sẽ được xử lý lọc kích thước và phân tích hình dáng để chọn ra các vùng ứng viên là biển báo giao thông.

**Bước 3:** Vùng ứng viên được chọn sẽ đưa vào model đã training trước đó để trích xuất đặc trưng.

**Bước 4:** Sau khi trích xuất đặc trưng, hệ thống sẽ phân lớp các đặc trưng của ứng viên để tiến hành nhận dạng loại biển báo.

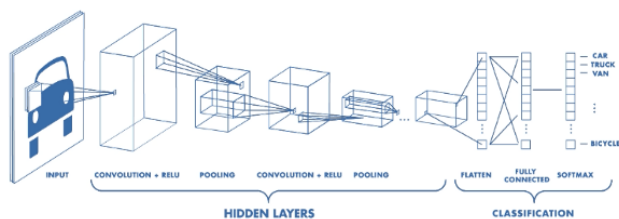
**Bước 5:** Xác nhận được loại biển báo phù hợp, hệ thống sẽ hiển thị lên giao diện cho người dùng thông tin và hình ảnh của biển báo đó. Kết thúc chương trình.

## V. PHƯƠNG PHÁP NGHIÊN CỨU

Trong bài báo cáo lần này, nhóm sử dụng thuật toán Convolutional Neural Network (CNN) đã được thiết kế sẵn để training dữ liệu giúp giải quyết bài toán nhận diện biển báo giao thông.

### A. Thuật toán CNN [3]

Convolutional Neural Networks (CNN) là một trong những mô hình deep learning phổ biến nhất và có ảnh hưởng nhiều nhất trong cộng đồng Computer Vision. CNN được dùng trong nhiều bài toán như nhận dạng ảnh, phân tích video, ảnh MRI, hoặc cho bài các bài của lĩnh vực xử lý ngôn ngữ tự nhiên, và hầu hết đều giải quyết tốt các bài toán này.



**Hình 2: Cấu trúc mạng Convolutional Neural Network**

Mạng CNN tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như **ReLU** và **tanh** để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

Trong mô hình CNN các layer liên kết được với nhau thông qua cơ chế convolution. Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.

Trong quá trình huấn luyện mạng (training) CNN tự động học các giá trị qua các lớp filter dựa vào cách thức mà bạn thực hiện.

Mô hình CNN có 2 khía cạnh cần quan tâm là tính bất biến (Location Invariance) và tính kết hợp (Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các góc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể.

Mạng CNN sử dụng 3 ý tưởng cơ bản: các trường tiếp nhận cục bộ (local receptive field), trọng số chia sẻ (shared weights), tổng hợp (pooling).

So với các thuật toán thông thường, CNN hiệu quả hơn hẳn bởi vì:

- **Weight sharing:** Trong các convolutional layers, khi thực hiện nhân tích chập sẽ dùng cùng các trọng số như nhau. Một kernel có thể dùng được nhiều lần, trong một bức ảnh.
- Các convolutional layer giúp trích xuất đặc trưng của một tấm ảnh/video, nhờ vậy mạng nơ-ron tích chập có khả năng học được các đặc trưng này (Feature learning).
- **Pre-trained model:** Có thể dùng lại được một model đã được train trước đó cho một bài toán hoàn toàn mới, nhờ ứng dụng Feature learning. Các model đã được train sẽ dùng để trích xuất các đặc trưng hữu dụng.

### Cách chọn tham số cho CNN

- Số các convolution layer: càng nhiều các convolution layer thì performance càng được cải thiện. Sau khoảng 3 hoặc 4 layer, các tác động được giảm một cách đáng kể.
- Filter size: thường filter theo size  $5 \times 5$  hoặc  $3 \times 3$ .
- Pooling size: thường là  $2 \times 2$  hoặc  $4 \times 4$  cho ảnh đầu vào lớn.
- Cách cuối cùng là thực hiện nhiều lần việc train test để chọn ra được param tốt nhất.

### B. Phân đoạn ảnh

Phân đoạn ảnh là một bước quan trọng trong các ứng dụng xử lý ảnh. Trong báo cáo này, nhóm sẽ tiến hành phân đoạn ảnh dựa vào màu đỏ (Red) trên các biển báo cấm (giới hạn của hệ thống này).

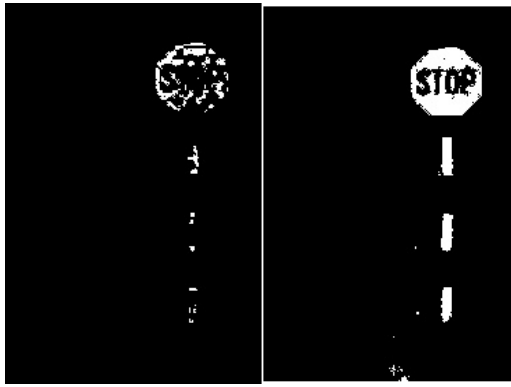
Đầu tiên, ảnh đầu vào (Hình 3) trong không gian màu RGB được chuyển sang không gian màu HSV để xử lý.

Từ ảnh HSV tạo ra mark1 (Hình 4) và mark2 (Hình 5) để tìm ra ngưỡng màu có của biển báo. Sau khi tìm được mark1 và mark2, gộp chúng lại bằng hàm bitwise\_or(), ta thu được mark (Hình 6) của vùng có thể là biển báo.

Khi có được phân vùng ta tiến hành làm mờ và làm mịn phân vùng (Hình 7) thông qua hàm GaussianBlur() trong opencv.



**Hình 3: Ảnh đầu vào, Hình 4: Mark 1**



Hình 5: Mark 2, Hình 6: Mark phân vùng biển báo



Hình 7: Mark được làm mờ và mịn

### C. Phát hiện vùng ứng viên và lọc nhiễu

Trong ảnh mark của vùng ứng viên như Hình 7. Tiếp theo, nhóm tìm và phát hiện cạnh của vùng ứng viên (Hình 8) bằng hàm Canny() của opencv.

Từ các vùng cạnh của vùng ứng viên, nhóm sử dụng hàm findContours() trong thư viện OpenCV để dò biên của các đối tượng trong ảnh sau đó khoanh vùng vẽ lại vùng ứng viên bằng hàm drawContours() kết quả như Hình 9.



Hình 8: Kết quả tìm biên của các đối tượng

Hình 9: Khoanh vùng ứng viên

Sau khi có kết quả khoanh vùng của các ứng viên có thể là biển báo, vì đây là bài toán nhận dạng biển báo đơn (giới hạn biển báo cấm) nên bây giờ chúng ta sẽ tiến hành lọc các phân vùng không cần thiết. Trong bài toán này, vùng ứng viên có khả năng là biển báo cao nhất là vùng ứng viên được khoanh vùng nối liền, không đứt mạch, width và height xấp xỉ bằng nhau và lớn hơn 100. Những vùng ứng viên nào không đủ điều kiện trên, ta sẽ loại bỏ. Khi xác định được vùng ứng viên ta sẽ vẽ hình chữ nhật bao quanh ứng viên (Hình 10) và crop ảnh gốc theo hình chữ nhật đó. Ta thu được ảnh biển báo chính xác như Hình 11.



Hình 10: Khoanh vùng biển báo sau khi lọc nhiễu

Hình 11: Kết quả Crop biển báo từ ảnh đầu vào

## VI. ƯU, NHƯỢC ĐIỂM VÀ HƯỚNG PHÁT TRIỂN

### A. Ưu điểm và nhược điểm

#### 1. Ưu điểm

- Áp dụng thành công thuật toán CNN với độ chính xác cao (khoảng 95%).
- Xây dựng được hệ thống nhận diện biển báo giao thông bằng CNN kết hợp Django để chạy trên nền tảng website.
- Tốc độ xử lý hệ thống nhanh.

#### 2. Nhược điểm

- Số lượng biển báo nhận diện biển báo còn quá ít.
- Chỉ xử lý được đầu vào là loại biển báo đơn.
- Thuật toán phát hiện phân vùng biển báo chưa được tối ưu.

### B. Hướng phát triển

Tăng số lượng mẫu huấn luyện, mở rộng hệ thống nhận diện biển báo nhiều hơn, đa dạng hơn.

Cải thiện thuật toán phát hiện phân vùng.

Xử lý được các ảnh đầu vào đa dạng hơn (2, 3 và nhiều biển báo hơn trong ảnh đầu vào).

Phát triển thêm tính năng nhận diện trích xuất biển báo qua video hoặc camera realtime.

## VII. KẾT LUẬN

Bài báo cáo đã trình bày thuật toán Convolutional Neural Network để nhận dạng biển báo giao thông Việt Nam với thời gian thực hiện nhanh hơn so với cùng hiệu năng nhận dạng. Xây dựng thành công website nhận dạng một số biển báo giao thông đường bộ tại Việt Nam trên ngôn ngữ Python với giao diện trực quan, sinh động, dễ sử dụng và hiệu suất cao.

Trong tương lai nhóm sẽ thực hiện thao tác tiền xử lý ảnh với các thuật toán cao cấp hơn cho quá trình xử lý. Kết hợp nhiều thuật toán như ICA, Neural Network, SVM..... để giúp cho thời gian và hiệu suất nhận dạng được tốt hơn.

## VIII. TÀI LIỆU THAM KHẢO

- [1] Diễn đàn Cấp cao Công nghệ Thông tin – Truyền thông Việt Nam (Vietnam ICT Summit), 2015. CNTT và Quản trị thông minh.
- [2] <https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign>
- [3] <https://topdev.vn/blog/thuật-toán-cnn-convolutional-neural-network/>

## IX. HƯỚNG DẪN CÀI ĐẶT

### A. Yêu cầu chung

Python 3.6.x trở lên.

Khuyến khích sử dụng Anaconda 3 tạo một environment tên opencv để cài các packages.

Packages cần có: pandas, numpy, matplotlib, tensorflow, opencv, scikit-image, scikit-learn, pillow, Django, tensorflow-mkl, keras.

### B. Training model (có thể bỏ qua bước này vì model đã được training sẵn trong project)

**Bước 1:** Tải source code [tại đây](#).

**Bước 2:** Chuẩn bị Input.

- Tải bộ input [tại đây](#)
- Giải nén file **Train.zip** và copy vào trong thư mục **\input\** của source

**Bước 3:** Mở Anaconda 3, vào mục Environment và chọn môi trường opencv chứa các packages đã cài đặt từ trước.

**Bước 4:** Dùng terminal của Anaconda 3, di chuyển đến thư mục **Train-Model-Traffic-Sign-Recognition\**

**Bước 5:** Chạy file **main.py** trong thư mục **main**

**python main/main.py**

**Bước 6:** Sau khi chạy xong, ta thu được file **model.h5** trong thư mục **\model\**. Đây chính là model cần thiết trong project chính của chúng ta.

### C. Cài đặt project

**Bước 1:** Tải source code [tại đây](#).

**Bước 2:** Kiểm tra trong thư mục **\static\model\** đã có file **model.h5** chưa. Nếu chưa có thì chúng ta lấy file **model.h5** đã có được ở trên bỏ vào thư mục này.

**Bước 3:** Mở Anaconda 3, vào mục Environment và chọn môi trường opencv chứa các packages đã cài đặt từ trước.

**Bước 4:** Dùng terminal của Anaconda 3, di chuyển đến thư mục gốc của source code.

**Bước 5:** Chạy sever localhost để vào chương trình

**python manage.py runserver**

**Bước 6:** Sau khi chạy lệnh trên ta vào trình duyệt truy cập **localhost:8000**, ta được giao diện như Hình 12.



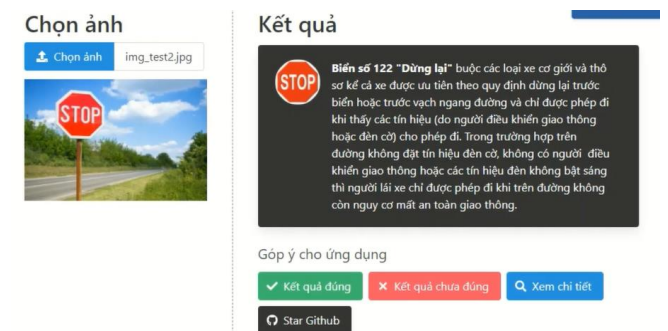
Hình 12: Giao diện chính của chương trình

**Bước 7:** Click vào button “Chọn Ảnh”, sau đó chọn ảnh mà ta cần nhận diện loại biển báo.



Hình 13: Chọn ảnh cần nhận diện

**Bước 8:** Sau khi chọn ảnh xong, đợi hệ thống xử lý và hiển thị kết quả lên màn hình.



Hình 14: Kết quả nhận dạng

Tải file **Demo.mp4** [tại đây](#) để xem chi tiết.