

Báo cáo đề tài BẢN TỐC ĐỘ

Chủ đề tài: Ngô Văn Thiện

GVHD: TS. Đào Việt Hùng

Người thực hiện báo cáo: Nguyễn Hoàng Việt

Ngày: 19/6/2023

1. Tóm tắt mục tiêu đề tài

- > Mục tiêu: từ ảnh phương tiện có chèn tốc độ, nhận diện được giá trị tốc độ trên ảnh, biển số xe (BKS) của phương tiện vi phạm
- > Phần cứng sử dụng: Jetson Nano

2. Phân tích dữ liệu

Bộ dữ liệu ảnh bao gồm 373 ảnh, có đặc điểm như sau:

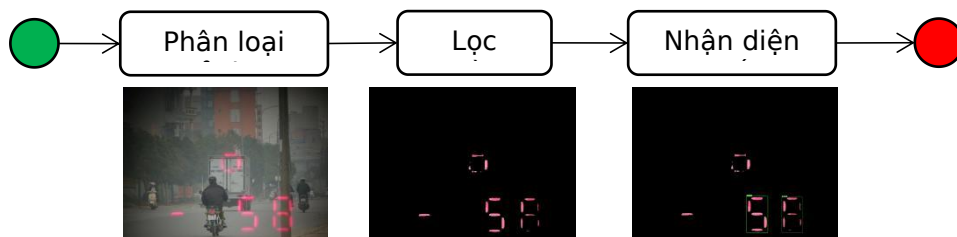
- > Kích thước: 4000×3000
- > Chỉ có số màu đỏ trên ảnh
- > Ví dụ:



Hình 1. Một số hình ảnh ví dụ

3. Giải pháp

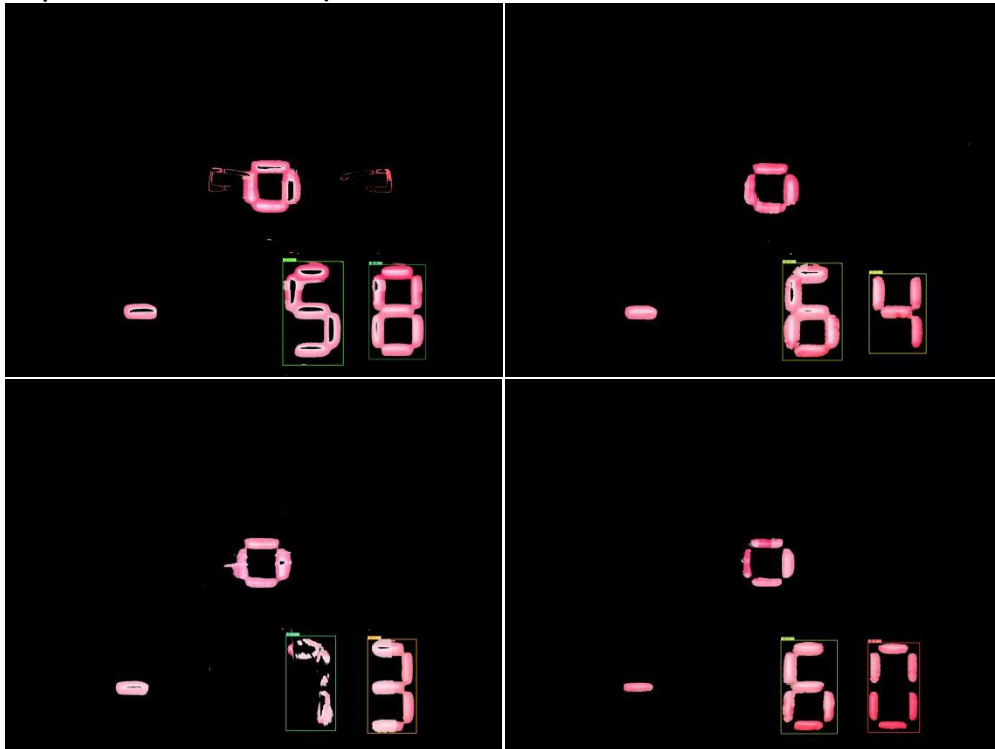
3.1. Nhận diện tốc độ (đã báo cáo)



Hình 2. Luồng xử lý nhận diện tốc độ

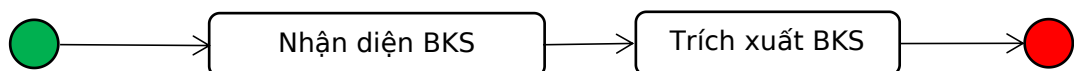
- > Phân loại ảnh: dựa vào đầu vào là kiểu ảnh loại nào (phần 1) để áp dụng phương pháp lọc màu phù hợp
- > Lọc màu: lọc thành phần có màu đỏ trong ảnh
- > Nhận diện số: sử dụng mô hình yolov5 để phát hiện số

Một số hình ảnh dự đoán:



Hình 3. Kết quả dự đoán

3.2. Nhận diện biển số xe



Hình 4. Luồng xử lý nhận diện BKS

- > Nhận diện BKS: sử dụng mô hình Yolov5, tự gán nhãn 356 ảnh
- > Trích xuất BKS: sử dụng mô hình PaddleOCR đã được pretrained sẵn trước đó
- > Hậu xử lý phần logic:
 - ✓ Mô hình Yolov5 detect được nhiều hơn 1 BKS: lấy biển nằm gần trung tâm nhất

- ✓ Mô hình PaddleOCR nhận diện thừa chữ cái (như trong ảnh, mô hình trích xuất được chữ “AA” trong logo “KIA”): chỉ lấy những ký tự đạt độ chính xác > 95%

```
(speed) vietnh@vietnh:~/Documents/project/speed_detect$ cd /home/vietnh/Documents/project/speed_detect/
vietnh/.vscode/extensions/ms-python.python-2023.10.1/pythonFiles/lib/python/debugpy/adapters/././cr/inference.py
[[[86.0, 0.0], [148.0, 0.0], [147.0, 14.0], [85.0, 10.0]], ('AA', 0.7061518430709839)]
[[[60.0, 29.0], [155.0, 35.0], [152.0, 92.0], [58.0, 86.0]], ('99A', 0.9955800175666809)]
[[[18.0, 85.0], [188.0, 93.0], [186.0, 150.0], [15.0, 141.0]], ('055.17', 0.9964777827262878)]
```

Hình 5. Đầu ra của mô hình PaddleOCR

- ✓ Ghép nối BKS thành 1 dòng
- ✓ Trả kết quả là một dictionary chứa 2 trường thông tin: “BKS” và “speed”

3.3. Deploy trên Jetson Nano

Jetson Nano:

- > Kiến trúc aarch64
- > JetPack L4T R32.5.1
- > Phiên bản Python mặc định: 3.6.9

Khó khăn:

- > Hầu hết các thư viện hiện tại đều hỗ trợ Python >= 3.7
- > 2 thư viện sử dụng cho AI: Torch, Torchvision có ít tài liệu hơn trên kiến trúc aarch64
- > Các thư viện khác sử dụng (opencv, numpy,...) có các phụ thuộc thư viện thuộc các phiên bản khác nhau
- > Paddle không hỗ trợ trên kiến trúc aarch64

=> Sử dụng docker để triển khai với ưu điểm không phụ thuộc vào môi trường của thiết bị triển khai mà chỉ dựa trên tài nguyên phần cứng, dễ triển khai với 1 câu lệnh duy nhất

Khó khăn gặp phải khi dựng docker:

- > Tài liệu cho Jetson Nano ít, không sẵn có phù hợp
- > Thời gian dựng lâu
- > Tài nguyên hạn chế

4. Công việc tương lai

- > Dựng Yolov5 trên Jetson Nano
- > Thử nghiệm Paddle Lite trên Jetson Nano hoặc tìm kiếm mô hình/thuật toán khác phù hợp
- > Tối ưu tài nguyên về dung lượng lưu trữ, tốc độ xử lý

5. Tài liệu

Link github:

https://github.com/VietNguyenHoang812/speed_detect.git
(nhánh develop, chỉ có demo, không bao gồm datasets)

