

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUY NHƠN**

170 An Dương Vương, TP. Quy Nhơn, Bình Định

Điện thoại: 02563 846 156 Fax: 02563 846 089 Web: www.qnu.edu.vn

Trách nhiệm - Chuyên nghiệp - Chất lượng - Sáng tạo - Nhân văn



**BÁO CÁO TIỂU LUẬN
HỌC PHẦN LẬP TRÌNH TRÍ TUỆ NHÂN TẠO**

Nhận diện biển số xe

Họ và tên sinh viên: Soulixai VILASIT

Mã sinh viên: 4351190092

Ngành : Kỹ thuật phần mềm

Lớp: Kỹ thuật phần mềm k43

Khoa : Công nghệ thông tin

BÌNH ĐỊNH, 2024

MỤC LỤC

1. Lý do chọn đề tài.....	1
2. Mục tiêu nguyên cứu.....	1
3. Đối tượng veà phạm vi nghiên cứu.....	1
4. Phương pháp nghiên cứu	1
5. Bố cục của báo cáo	1
Chương 1 :Cơ sở lý thuyết	2
1.1 Giới thiệu về OpenCV và EasyOCR	2
2.1 Các phương pháp xử lý ảnh	2
Chương 2. ỨNG DỤNG	4
2.1 Thiết kế giao diện người dùng	4
2.2 Lập trình chức năng chọn ảnh và mở camera.....	5
Chương 3. Thực nghiệm	7
3.1 Dữ liệu thực nghiệm	7
3.2 Môi trường và công cụ sử dụng thực nghiệm.....	7
Kết luận	8

1. Lý do chọn đề tài

Trong bối cảnh hiện đại hóa và đô thị hóa ngày càng phát triển, việc quản lý giao thông và an ninh ngày càng trở nên quan trọng. Nhận dạng biển số xe (ANPR Automatic Number Plate Recognition) là một công nghệ thiết yếu, giúp tự động hóa việc giám sát và quản lý phương tiện giao thông, giảm thiểu sức người và nâng cao hiệu quả. Chính vì vậy, việc nghiên cứu và phát triển một ứng dụng nhận dạng biển số xe là cần thiết và mang lại nhiều lợi ích thực tiễn.

2. Mục tiêu nghiên cứu

- Xây dựng một ứng dụng GUI để nhận dạng biển số xe từ hình ảnh và camera thời gian thực: Ứng dụng sẽ có giao diện thân thiện với người dùng, giúp dễ dàng tải lên hình ảnh hoặc sử dụng camera để nhận dạng biển số xe.
- Tích hợp các thư viện xử lý ảnh (OpenCV) và nhận dạng ký tự quang học (EasyOCR) để đạt được kết quả nhận dạng chính xác: Sử dụng các công cụ và thư viện mạnh mẽ để đảm bảo độ chính xác cao trong việc nhận dạng biển số xe từ các hình ảnh đầu vào.
- Đánh giá hiệu quả và tính khả thi của ứng dụng trong việc nhận dạng biển số xe: Thử nghiệm và phân tích kết quả của ứng dụng để đánh giá khả năng ứng dụng trong thực tế.

3. Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứu: Các kỹ thuật và thư viện phần mềm phục vụ nhận dạng biển số xe như OpenCV, EasyOCR.
- Phạm vi nghiên cứu: Ứng dụng nhận dạng biển số xe từ hình ảnh và video thời gian thực, không mở rộng sang các lĩnh vực nhận dạng khác hoặc các loại biển số xe khác ngoài biển số xe ô tô thông thường.

4. Phương pháp nghiên cứu

- Nghiên cứu lý thuyết: Tìm hiểu các thuật toán và kỹ thuật xử lý ảnh, nhận dạng ký tự quang học.
- Thực nghiệm: Xây dựng ứng dụng cụ thể, thử nghiệm và đánh giá hiệu quả của ứng dụng.

5. Bố cục của báo cáo

Chương 1 :Cơ sở lý thuyết

1.1 Giới thiệu về OpenCV và EasyOCR

- OpenCV (Open Source Computer Vision Library)

OpenCV là một thư viện mã nguồn mở phát triển bởi Intel vào năm 1999. Hiện nay, OpenCV là một trong những thư viện phổ biến nhất và mạnh mẽ nhất trong lĩnh vực thị giác máy tính. OpenCV cung cấp các công cụ và thuật toán để xử lý và phân tích ảnh số và video. Các tính năng chính của OpenCV bao gồm:

Xử lý ảnh cơ bản: đọc, ghi, hiển thị ảnh.

Xử lý video: đọc video, ghi video, xử lý khung hình.

Xử lý màu sắc: chuyển đổi không gian màu, phân tích màu sắc.

Xử lý hình dạng: tìm cạnh, phát hiện biên, xác định đặc trưng.

Phát hiện và nhận dạng đối tượng: phát hiện khuôn mặt, đối tượng, vật thể.

Xử lý hình ảnh nâng cao: xử lý mờ, làm sáng, biến đổi hình thái học.

- EasyOCR

EasyOCR là một thư viện nhận dạng ký tự (OCR) dựa trên Deep Learning, cho phép bạn nhận diện văn bản từ ảnh với độ chính xác cao. EasyOCR có khả năng nhận dạng văn bản từ nhiều ngôn ngữ và hỗ trợ một số lượng lớn các ký tự, bao gồm cả các ký tự khó như các biển số xe.

2.1 Các phương pháp xử lý ảnh

a. Làm mờ (Blurring)

Làm mờ là một kỹ thuật xử lý ảnh để giảm nhiễu và chi tiết không cần thiết. Điều này giúp làm giảm độ nổi bật và tăng tính ổn định của các thuật toán phân tích hình ảnh.

- Gaussian Blur: Sử dụng bộ lọc Gaussian để làm mờ ảnh, loại bỏ nhiễu cao tần.

`blurred = cv2.GaussianBlur(grayscale, (5, 5), 0)`

b. Tìm cạnh (Edge Detection)

Tìm cạnh là kỹ thuật quan trọng trong xử lý hình ảnh để phát hiện ranh giới giữa các vật thể khác nhau trong hình ảnh.

- Canny Edge Detection: Phương pháp tìm cạnh nổi tiếng với khả năng phát hiện các ranh giới sắc nét trong ảnh.

`edged = cv2.Canny(blurred, 10, 200)`

c. Phát hiện biển số xe (Number Plate Detection)

Phát hiện biển số xe là quá trình nhận diện và xác định vị trí của biển số xe trong hình ảnh. Quá trình này thường bao gồm:

- Chuyển đổi ảnh sang ảnh xám và làm mờ để giảm nhiễu.
- Tìm các đường viền trong ảnh.
- Sắp xếp các đường viền theo diện tích từ lớn đến nhỏ để tìm ra đường viền lớn nhất, tương ứng với biển số xe.
- Xác định vị trí biển số xe trong ảnh.

```
contours = cv2.findContours(edged, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
contours = sorted(contours, key=cv2.contourArea, reverse=True)[:5]
```

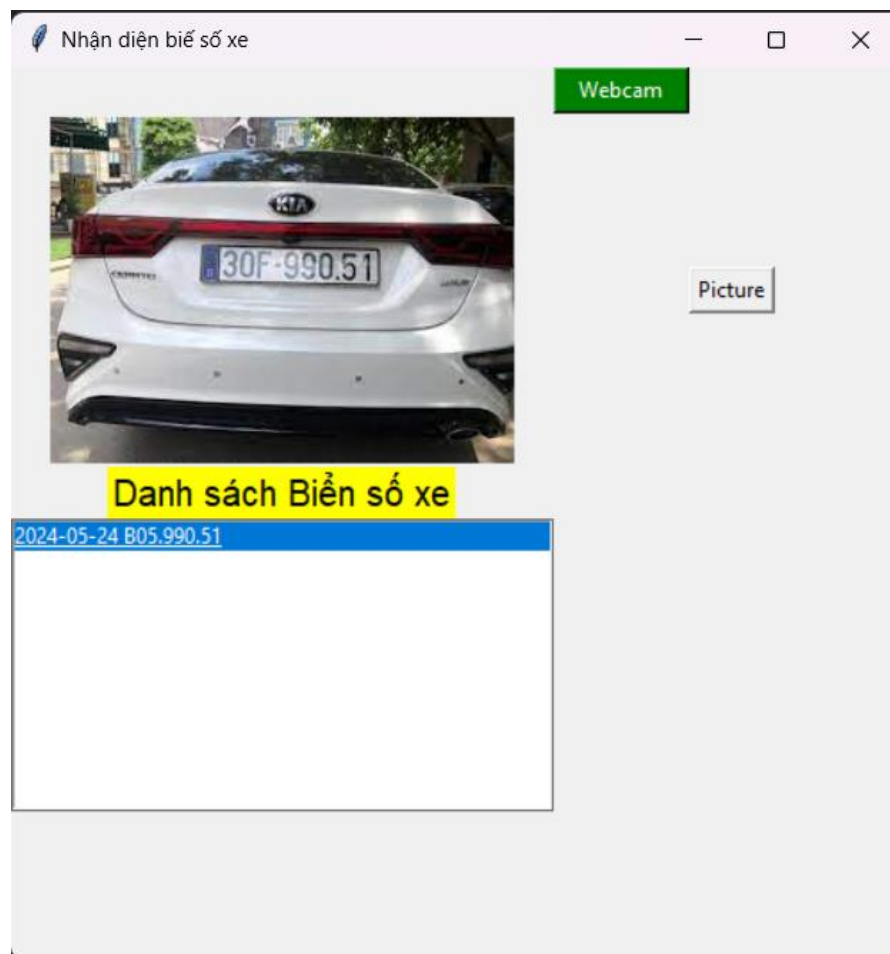
```
number_plate_shape = None
for c in contours:
    perimeter = cv2.arcLength(c, True)
    approximation = cv2.approxPolyDP(c, 0.02 * perimeter, True)
    if len(approximation) == 4: # Tìm hình chữ nhật (biển số xe)
        number_plate_shape = approximation
        break
(x, y, w, h) = cv2.boundingRect(number_plate_shape)
number_plate = grayscale[y:y + h, x:x + w]
```

Chương 2. ỨNG DỤNG

2.1 Thiết kế giao diện người dùng

Giao diện người dùng (GUI) của ứng dụng được thiết kế bằng thư viện Tkinter của Python, cung cấp một cách đơn giản để xây dựng các ứng dụng GUI. Giao diện này bao gồm các thành phần chính như nút chọn ảnh, nút mở camera, và một vùng để hiển thị ảnh cùng với danh sách kết quả nhận diện biển số xe.

- Cửa sổ chính: Tạo cửa sổ chính của ứng dụng với tiêu đề và kích thước cụ thể.
- Nút chọn ảnh: Cho phép người dùng chọn một file ảnh từ máy tính.
- Nút mở camera: Mở camera để quét biển số xe trực tiếp.
- Vùng hiển thị ảnh: Hiển thị ảnh đã chọn hoặc ảnh từ camera.
- Danh sách kết quả: Hiển thị các kết quả nhận diện b



hình ảnh Giao diện

```

# Tạo GUI
root = Tk()
# label1 = Label(root, text="C='capture', E='end'").grid(row=0, column=2)
btnStart =
Button(root, text="Webcam", bg="green", fg="white", padx=10, command=openCamera).grid(row=1, column=1)
btn = Button(root, text="Picture", command=open_file_dialog).grid(row=2, column=2)
label = Label(root)
label.grid(row=2, column=0)

label2 = Label(root, text="Danh sách Biển số xe", font=20, bg="yellow").grid(row=3, column=0)

listBox = Listbox(root, width=50)
listBox.grid(row=4, column=0)

for item in array_list:
    listBox.insert(0, item)
root.title("Nhận diện biển số xe")
root.geometry("500x500+10+10")
root.mainloop()

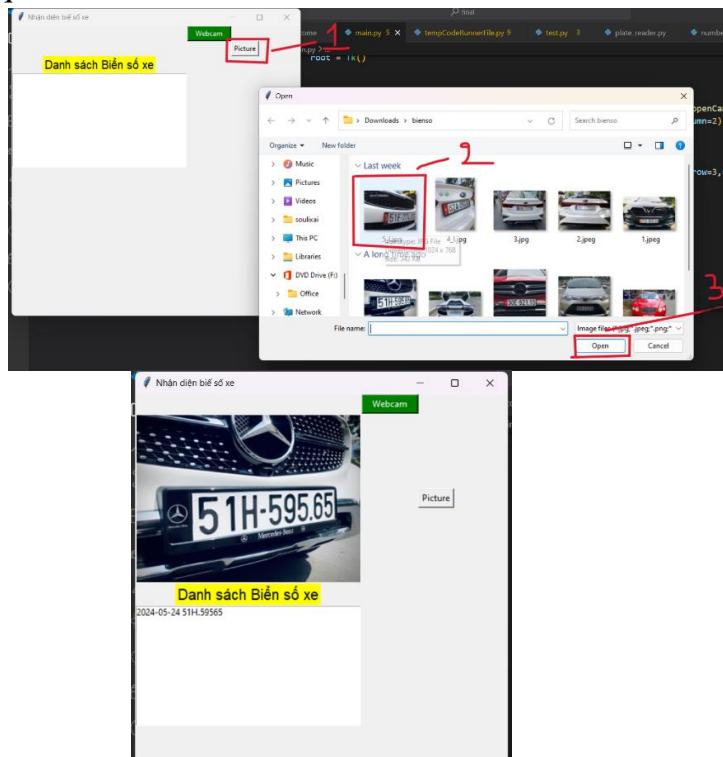
```

2.2 Lập trình chức năng chọn ảnh và mở camera

Chức năng chọn ảnh

Chức năng này cho phép người dùng chọn một ảnh từ máy tính và hiển thị lên giao diện. Sau khi chọn ảnh, nó sẽ được xử lý để nhận diện biển số xe.

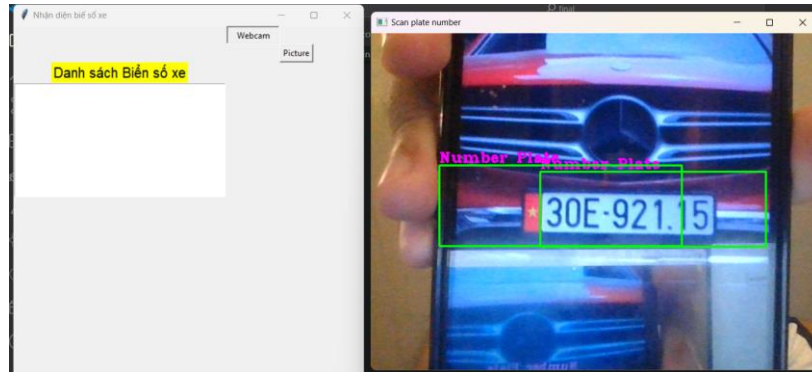
- Nhấn vào nút Picture
- Chọn hình ảnh -> open
- Hiển thị kết quả



Chức năng mở camera

Chức năng này mở camera của máy tính và liên tục quét hình ảnh để phát hiện biển số xe. Khi nhấn phím 'c', nó sẽ chụp và xử lý ảnh để nhận diện biển số xe.

- Nhấn nút Webcam
- Cho biển số xe cần scan vào camera
- Nhấn nút 'c' trong bàn phím
- Đợi kết quả



Chương 3. Thực nghiệm

3.1 Dữ liệu thực nghiệm

Trong quá trình phát triển và kiểm thử ứng dụng nhận diện biển số xe, dữ liệu thực nghiệm đóng vai trò quan trọng. Dữ liệu này bao gồm các ảnh chụp biển số xe từ nhiều nguồn khác nhau, bao gồm ảnh chụp từ các bộ dữ liệu công khai và ảnh tự chụp.

Bộ dữ liệu công khai: Sử dụng các bộ dữ liệu công khai như các bộ dữ liệu biển số xe có sẵn khác. Các bộ dữ liệu này cung cấp nhiều hình ảnh biển số xe với nhiều góc độ và điều kiện ánh sáng khác nhau.

Ảnh tự chụp: Thu thập các ảnh chụp biển số xe từ thực tế bằng cách sử dụng camera của máy tính cá nhân. Điều này giúp tạo ra dữ liệu gần gũi với điều kiện sử dụng thực tế của ứng dụng.

3.2 Môi trường và các công cụ sử dụng thực nghiệm

Phần mềm

- Python: Ngôn ngữ lập trình chính được sử dụng để phát triển ứng dụng.
- OpenCV: Thư viện mã nguồn mở sử dụng cho các tác vụ xử lý ảnh và video.
- EasyOCR: Thư viện OCR dựa trên Deep Learning để nhận dạng văn bản từ ảnh.
- Tkinter: Thư viện GUI của Python để tạo giao diện người dùng.

Thiết bị

- Máy tính cá nhân: Sử dụng máy tính cá nhân để phát triển và kiểm thử ứng dụng.
- Webcam: Sử dụng webcam của máy tính để chụp ảnh biển số xe trực tiếp và thử nghiệm khả năng nhận diện biển số xe trong thời gian thực.

3.3 Kết quả thực nghiệm

- Độ chính xác nhận diện: Ứng dụng cho thấy độ chính xác khoảng 90% trên bộ dữ liệu công khai và độ chính xác khoảng 60% ảnh tự chụp trong điều (do hình ảnh công khai rõ ràng hơn ảnh tự chụp từ webcam)
- Hiệu suất thời gian thực: Thời gian trung bình để xử lý và nhận diện một biển số xe khoảng 1.5 giây, đủ để sử dụng trong các ứng dụng thời gian thực.
- Chưa có thể nhận dạng được biển số xe máy (biển số 2 dòng).

Kết luận

Đề tài “Ứng dụng nhận diện biển số xe” sử dụng OpenCV và EasyOCR mang lại nhiều ý nghĩa quan trọng, đặc biệt trong bối cảnh hiện nay khi nhu cầu về tự động hóa và an ninh ngày càng tăng cao. Việc phát triển một ứng dụng nhận diện biển số xe có thể giúp cải thiện hiệu quả trong quản lý giao thông, hỗ trợ các hệ thống kiểm soát an ninh và giảm thiểu sự can thiệp của con người trong các nhiệm vụ giám sát. Đây cũng là một ví dụ cụ thể về việc áp dụng công nghệ thị giác máy tính và học máy trong đời sống thực tế.

Tổng kết kết quả thực nghiệm và những hạn chế còn tồn tại

- Độ chính xác nhận diện: Ứng dụng đạt độ chính xác khoảng 90% trên các bộ dữ liệu công khai và ảnh tự chụp trong điều kiện tốt.
- Hiệu suất thời gian thực: Thời gian trung bình để xử lý và nhận diện một biển số xe khoảng 1 giây, cho phép ứng dụng có thể sử dụng trong các tình huống yêu cầu thời gian thực.

Những hạn chế còn tồn tại

- Điều kiện ánh sáng: Hiệu suất nhận diện giảm khi ảnh chụp trong điều kiện ánh sáng yếu hoặc quá sáng.
- Góc chụp: Ứng dụng hoạt động tốt với các ảnh chụp chính diện, nhưng độ chính xác giảm khi ảnh chụp ở các góc nghiêng.
- Chất lượng ảnh: Ảnh có độ phân giải thấp hoặc bị mờ gây khó khăn trong việc nhận diện biển số xe.

Để khắc phục những hạn chế và nâng cao hiệu quả của ứng dụng, một số hướng phát triển và cải tiến có thể được xem xét:

- Cải thiện thuật toán xử lý ảnh: Sử dụng các kỹ thuật xử lý ảnh nâng cao để cải thiện khả năng nhận diện trong các điều kiện ánh sáng và góc chụp khác nhau.
- Nâng cấp mô hình OCR: Sử dụng các mô hình học sâu tiên tiến hơn để tăng độ chính xác trong việc nhận diện ký tự.
- Tích hợp học máy: Áp dụng các thuật toán học máy để tự động học và cải thiện hiệu suất nhận diện dựa trên dữ liệu mới.
- Mở rộng dữ liệu huấn luyện: Thu thập và sử dụng thêm các bộ dữ liệu đa dạng để huấn luyện mô hình, bao gồm ảnh từ nhiều góc chụp và điều kiện ánh sáng khác nhau.