* **Lý do chọn đề tài:** Nhìn hình ta có thể thấy, mật độ xe ô tô rất cao. Trong khi đó, cơ sở hạ tầng giao thông lại không đáp ứng được, dẫn đến tình trạng đậu xe trái phép diễn ra thường xuyên. Xây dựng các bãi đậu xe là điều cần thiết. Và ở các các bãi đậu xe, còn sử dụng việc ghi giấy thủ công để lưu lại thông tin xe, điều này có thể xảy ra các sai sót không đáng có. Vì vậy, em làm để tải này để nhận dạng và lưu thông tin xe một cách tự động và dễ dàng, tiết kiệm thời gian và chi phí.
* **Mục tiêu:** nghiên cứu, xây dựng hệ thống nhận dạng biển số xe và ứng dụng quản lý bãi giữ xe ô tô.
* **Phạm vi nghiên cứu:** nghiên cứu tổng quan một số phương pháp xử lý ảnh và nhận dạng ký tự.
* **Có nhiều phương pháp** phát hiện cạnh khác nhau như Sobel - Robet, nhưng Canny cho phép hiện đầy đủ hơn các biên có thể có trên các bức ảnh, cho ra kết quả tốt hơn.

**Canny** là một phương pháp phát hiện cạnh, được phát triển bởi John F. Canny vào năm 1986, được thực hiện qua bốn bước:

**Bước 1: Giảm nhiễu** bằng cách áp dụng một bộ lọc Gauss.

**Bước 2: Tìm Gradient** là để phát hiện các cạnh ngang, dọc và chéo trong ảnh bị mờ. Từ đó, độ dốc và hướng của cạnh có thể được xác định. Khi tính toán, giá trị hướng gradient sẽ nằm trong 4 hướng: hướng ngang (0 độ), hướng chéo phải (45 độ), hướng dọc (90 độ) và hướng chéo trái (135 độ).

**Bước 3: Tiến hành “Non-maximum suppression”**

Tức là loại bỏ các pixel ở vị trí không phải cực đại toàn cục. Kết thúc bước này ta được một mặt nạ nhị phân (ảnh nhị phân).

**Bước 4: Lọc ngưỡng**: ta sẽ xét các pixel dương trên mặt nạ nhị phân. Nếu giá trị gradient vượt ngưỡng **maxVal** thì pixel đó chắc chắn là cạnh. Các pixel có độ lớn gradient nhỏ hơn ngưỡng **minVal** sẽ bị loại bỏ. Còn các pixel nằm trong khoảng 2 ngưỡng trên sẽ được xem xét rằng nó có nằm liên kề với những pixel được cho là "chắc chắn là cạnh" hay không. Nếu liền kề thì ta giữ, còn không liền kề bất cứ pixel cạnh nào thì ta loại.

**Công cụ nhận dạng ký tự Tesseract**

Để nhận dạng được các ký tự sau khi đã tách được biển số ra khỏi ảnh gốc, ta sử dụng công cụ nhận dạng **Tesseract**. Bao gồm 4 bước:

**Phân tích layout**

Trước tiên, hình ảnh sẽ được phân tích để tìm ra các vùng kết nối (connected component).

**Tìm kiếm dòng và ký tự**

Để nhận dạng, ta phải tìm dòng và ký tự. Mỗi dòng văn bản luôn có tọa độ chặn dưới vàchặn trên, trong khi đó mỗi ký tự có tọa độ chặn dưới, chặn trên, giới hạn trái và giớihạn phải.

**Nhận dạng từ**

Đầu vào sẽ được đánh giá, phân tích hai lần. Ở lần đầu tiên, sẽ nhận diện kết quả phân tích ở bước trước đó. Các kết quả nhận diện thoả mãn yêu cầu sẽ được đưa vào tập tin huấn luyện để hỗ trợ cho quá trình nhận diện lần thứ hai với các kết quả chưa đạt yêu cầu. Trải qua một quá trình lặp đi lặp lại gồm các bước nhận diện ký tự, ghép ký tự và so khớp với từ điển.Việc xác nhận kết quả có thoả mãn yêu cầu hay không cần phải dựa trên nhiều tiêu chí bao gồm khoảng cách của các ký tự, độ phù hợp với từ điển và khoảng cách đến các dấu câu. Cuối cùng, sẽ xử lý những dấu cách không rõ ràng để đi đến kết quả cuối cùng.

* **Trong quá trình nhận dạng**, có trường hợp nhầm lẫn là giữa các số: 9 và 8, 6 và 5, 0 và 6, 0 và 9, B và D, D và P. Các trường hợp nhầm lẫn này là do các cặp số có đặc trưng gần giống nhau. Có trường hợp bị ảnh hưởng bởi tác động của môi trường như bị chói do ánh sáng, quá tối và mờ để nhận dạng.

**Đánh giá**

Kết quả này dựa vào việc thử nghiệm 30 xe tô khác nhau, 24 xe ô tô nhận dạng đúng hết cả dãy số,6 ô tô còn lại nhận dạng sai từ 1 đến 2, nhìn chung việc nhận dạng khá chính xác. Nhưng bên cạnh đó cũng có một số kết quả sai và cần cải thiện để có kết quả tốt hơn. Điều này sẽ phát triển thêm trong tương lai.