ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN

Tel. (84-511) 736 949, Fax. (84-511) 842 771

Website: [itf.dut.edu.vn](mailto:itf.dut.edu.vn), E-mail: cntt@dut.udn.vn



LUẬN VĂN TỐT NGHIỆP KỸ SƯ

KHOA CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI

ỨNG DỤNG TRA TỪ ĐIỂN ANH – VIỆT SỬ DỤNG OCR ĐỂ NHẬN DẠNG KÝ TỰ QUANG HỌC TRÊN HỆ ĐIỀU HÀNH ANDROID

SINH VIÊN : HUỲNH BÁ HẢI

LỚP : 12T3

MSSV : 102120174

GVHD : TS. LÊ THỊ MỸ HẠNH

**ĐÀ NẴNG, 05/2017**

# LỜI CÁM ƠN

Em chân thành cảm ơn Cô TS. Lê Thị Mỹ Hạnh về những hướng dẫn và chỉ bảo tận tình của Cô trong suốt thời gian em nghiên cứu và hoàn thành đồ án tốt nghiệp.

Em cũng xin gửi lời cảm ơn đến toàn thể các Thầy Cô giáo trong khoa Công Nghệ Thông Tin về những kiến thức quý báu mà các Thầy Cô đã mang đến cho chúng em trong suốt các năm học tại trường.

Mình cũng muốn gửi lời cảm ơn đến những người bạn cùng học trong suốt những năm đại học. Cảm ơn những chia sẻ và góp ý của các bạn.

Con cũng xin gửi lời cảm ơn đến cha mẹ và người thân trong gia đình đã luôn ở bên con, động viên con vượt qua mọi khó khăn trên con đường học vấn. Con luôn ghi nhớ công ơn của cha mẹ đã vất vả vì con để con có được thành công như ngày hôm nay.

# LỜI CAM ĐOAN

*Em xin cam đoan:*

* 1. *Những nội dung trong luận văn này là do em thực hiện dưới sự hướng dẫn trực tiếp của Cô TS. Lê Thị Mỹ Hạnh.*
  2. *Mọi tham khảo dùng trong luận văn đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.*
  3. *Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo, hay gian trá, em xin chịu hoàn toàn trách nhiệm.*

*Sinh viên,*

*Huỳnh Bá Hải*

# MỤC LỤC

[LỜI CÁM ƠN 2](#_Toc482588496)

[LỜI CAM ĐOAN 3](#_Toc482588497)

[MỤC LỤC 4](#_Toc482588498)

[DANH MỤC CÁC TỪ VIẾT TẮT 7](#_Toc482588499)

[DANH MỤC CÁC HÌNH VẼ 8](#_Toc482588500)

[DANH MỤC CÁC BẢNG 9](#_Toc482588501)

[Chương 1: TỔNG QUAN ĐỀ TÀI 10](#_Toc482588502)

[1.1. Lý do chọn đề tài 10](#_Toc482588503)

[1.2. Mục tiêu và nhiệm vụ 10](#_Toc482588504)

[1.1.1. Mục tiêu 10](#_Toc482588505)

[1.1.2. Nhiệm vụ 10](#_Toc482588506)

[1.3. Đối tượng nghiên cứu và phạm vi hệ thống 11](#_Toc482588507)

[1.3.1. Đối tượng nghiên cứu 11](#_Toc482588512)

[1.3.2. Phạm vi hệ thống 11](#_Toc482588513)

[1.4. Phương pháp phát triển 11](#_Toc482588514)

[1.4.1. Phương pháp nghiên cứu 11](#_Toc482588520)

[1.4.2. Phương pháp thi hành 11](#_Toc482588521)

[Chương 2: CƠ SỞ LÝ THUYẾT 12](#_Toc482588522)

[2.1. Android 12](#_Toc482588525)

[2.1.1. Tổng quan về Android 12](#_Toc482588526)

[2.1.2. Các đặc điểm của Android 12](#_Toc482588527)

[2.1.3. Cấu trúc của Android 13](#_Toc482588533)

[2.2. OCR 16](#_Toc482588534)

[2.1.1. Tổng quan về OCR 16](#_Toc482588538)

[2.1.2. Các ứng dụng của OCR 16](#_Toc482588539)

[2.1.3. Các thể loại 16](#_Toc482588540)

[2.1.4. Các kỹ thuật 17](#_Toc482588541)

[2.1.5. Độ chính xác 17](#_Toc482588542)

[2.3. Tesseract OCR 19](#_Toc482588543)

[2.4. Tess-two 20](#_Toc482588544)

[2.4.1. Tổng quan 20](#_Toc482588551)

[2.4.2. Yêu cầu cần thiết 20](#_Toc482588552)

[2.4.3. Cách sử dụng 20](#_Toc482588553)

[2.4.4. Ứng dụng trong đề tài 20](#_Toc482588554)

[2.5. Realm 21](#_Toc482588555)

[2.5.1. Tổng quan 21](#_Toc482588563)

[2.5.2. Yêu cầu cần thiết 21](#_Toc482588564)

[2.5.3. Cách sử dụng 21](#_Toc482588565)

[2.5.4. Ứng dụng trong đề tài 23](#_Toc482588566)

[2.6. Tra cứu từ điển 24](#_Toc482588567)

[Chương 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 25](#_Toc482588568)

[3.1. Mô tả hệ thống 25](#_Toc482588572)

[3.1.1. Yêu cầu chức năng 25](#_Toc482588577)

[3.1.2. Yêu cầu phi chức năng 25](#_Toc482588578)

[3.1.3. Yêu cầu hệ thống 26](#_Toc482588579)

[3.2. Phân tích thiết kế 26](#_Toc482588580)

[3.2.1. Biểu đồ hoạt động 26](#_Toc482588586)

[3.2.2. Biểu đồ tuần tự 28](#_Toc482588587)

[3.2.3. Thiết kế cơ sở dữ liệu 30](#_Toc482588588)

[Chương 4: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ 31](#_Toc482588589)

[4.1. Kết quả Demo 31](#_Toc482588594)

[4.1.1. Menu ứng dụng 31](#_Toc482588595)

[4.1.2. Tra từ bằng cách nhập 32](#_Toc482588596)

[4.1.3. Kết quả tra từ 33](#_Toc482588597)

[4.1.4. Xem lịch sử tìm kiếm 34](#_Toc482588598)

[4.1.5. Tra từ bằng Camera 35](#_Toc482588599)

[4.2. Đánh giá 36](#_Toc482588600)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 37](#_Toc482588601)

[1. Kết luận 37](#_Toc482588602)

[1.1. Những vấn đề đã đạt được 37](#_Toc482588603)

[1.2. Những vấn đề còn tồn tại 37](#_Toc482588604)

[2. Hướng phát triển 37](#_Toc482588605)

[TÓM TẮT LUẬN VĂN TỐT NGHIỆP 38](#_Toc482588606)

[TÀI LIỆU THAM KHẢO 40](#_Toc482588607)

# DANH MỤC CÁC TỪ VIẾT TẮT

OCR Optical Character Recognition – Nhận dạng ký tự quang học

# DANH MỤC CÁC HÌNH VẼ

[Hình 2.1: Cấu trúc Android 14](#_Toc482589466)

[Hình 2.2: File build.gradle (Project) 22](#_Toc482589467)

[Hình 2.3: File build.gradle (Application) 23](#_Toc482589468)

[Hình 3.1: Usecase - Tra cứu từ điển 25](#_Toc482589469)

[Hình 3.2: Sơ đồ hoạt động tra cứu từ bằng cách nhập tay 27](#_Toc482589470)

[Hình 3.3: Sơ đồ hoạt động tra cứu từ bằng Camera 28](#_Toc482589471)

[Hình 3.4: Sơ đồ hoạt động xem lịch sử tra cứu từ 29](#_Toc482589472)

[Hình 3.5: Sơ đồ tuần tự tra cứu từ bằng cách nhập tay 30](#_Toc482589473)

[Hình 3.6: Tra cứu từ bằng Camera 30](#_Toc482589474)

[Hình 3.7: Thiết kế cơ sở dữ liệu 31](#_Toc482589475)

[Hình 4.1: Menu ứng dụng 32](#_Toc482589476)

[Hình 4.2: Tra từ bằng các nhập tay 33](#_Toc482589477)

[Hình 4.3: Kết quả tra cứu từ 34](#_Toc482589478)

[Hình 4.4: Lịch sử tra cứu từ 35](#_Toc482589479)

[Hình 4.5: Tra từ bằng Camera 1 36](#_Toc482589480)

[Hình 4.6: Tra từ bằng Camera 2 36](#_Toc482589481)

[Hình 4.7: Tra từ bằng Camera 3 37](#_Toc482589482)

DANH MỤC CÁC BẢN

[Bảng 2.1: Đặc điểm của Android12](#_Toc482589721)

[Bảng 3.1: Mô tả chi tiết usecase tra từ bằng cách nhập tay35](#_Toc482589722)

[Bảng 3.2: Mô tả chi tiết usecase tra từ bằng camera35](#_Toc482589723)

[Bảng 3.3: Mô tả chi tiết usecase xem lịch sử tra cứu35](#_Toc482589724)

[Bảng 3.4: Bảng mô tả chi tiết cơ sở dữ liệu40](#_Toc482589725)

# Chương 1: TỔNG QUAN ĐỀ TÀI

Trong này sẽ trình bày tổng quan về đề tài, lý do chọn đề tài, tiếp theo sẽ trình bày về mục tiêu và nhiệm vụ để hoàn thành đề tài và phần cuối cùng trong chương này sẽ trình bày về các đối tượng cần nghiên cứu, phạm vi đề tài và phương pháp phát triển.

## Lý do chọn đề tài

Ngày nay các thiết bị số cầm tay như điện thoại di động, điện thoại thông minh, máy tính bảng có rất nhiều trên thị trường. Với giá thành ngày càng rẻ và các lợi ích đem lại, các thiết bị đó ngày càng được sử dụng nhiều và phổ biến. Một nền tảng được sử dụng trong các thiết bị đó là hệ điều hành thông minh Android của Google. Do vậy đi cùng việc phát triển và sử dụng các thiết bị phần cứng một cách hiệu quả thì việc phát triển các ứng dụng phần mềm chạy trên nền Android càng lúc càng tăng cao.

Nhờ sự phát triển của các ứng dụng phần mềm hiện nay mà việc học ngoại ngữ cũng trở nên đơn giản. Hiện tại có rất nhiều ứng dụng phần mềm hỗ trợ việc học ngoại ngữ và những ứng dụng từ điển cũng góp phần không nhỏ vào việc nâng cao hiệu quả học tập ngoại ngữ. Các ứng dụng từ điển hiện tại đều có điểm chung là yêu cầu người dùng nhập thủ công từ cần tra cứu vào ô tra cứu để có được kết quả, điều này gây cản trở không nhỏ nếu người dùng muốn tra cứu nhanh chóng hoặc muốn tra cứu nhiều từ cùng một lúc. Nếu có thể sử dụng camera để quét từ và nhận dạng chữ thì vấn đề này có thể được giải quyết. Vì vậy em quyết định lựa chọn đề tài “Ứng dụng tra từ điển Anh - Việt sử dụng OCR để nhận diện ký tự quang học trên hệ điều hành Android” nhằm đơn giản hóa việc tra cứu từ vựng Tiếng Anh chỉ bằng các thao tác đơn giản mang ý nghĩa thực tế.

## Mục tiêu và nhiệm vụ

### Mục tiêu

Mục tiêu của đề tài “Xây dựng ứng dụng tra từ điển Anh - Việt sử dụng OCR để nhận dạng ký tự quang học trên hệ điều hành Android” là tạo ra một ứng dụng di động tra từ điển Anh – Việt, ứng dụng công nghệ OCR để nhận dạng chữ quang học trong ảnh chụp được từ camera và từ kết quả nhận dạng này giúp người dùng tra cứu nghĩa của từng từ một. Ngoài ra, ứng dụng còn có thể giúp người dùng tra cứu từ bằng cách nhập bằng tay và xem lại lịch sử mình đã tra cứu.

### Nhiệm vụ

Để hoàn thành mục tiêu đề ra trong đề tài thì những nhiệm vụ phải hoàn thành đối với em là ứng dụng công nghệ OCR để nhận dạng ký tự từ ảnh chụp được từ camera của điện thoại và cho ra kết quả là chuỗi ký tự, phân tích file dữ liệu từ điển để tạo bộ dữ liệu từ điển cho ứng dụng phục vụ cho việc tra cứu.

## Đối tượng nghiên cứu và phạm vi đề tài



### Đối tượng nghiên cứu

Với nhiệm vụ đề ra trong đề tài là ứng dụng công nghệ OCR để nhận dạng ký tự quang học từ ảnh chụp được trên điện thoại và sử dụng kết quả là chuỗi ký tự nhận dạng được để thực hiện tra cứu thì các đối tượng cần nghiên cứu để hoàn thành nhiệm vụ này là:

* + OCR - Optical Character Reconition: công nghệ nhận dạng ký tự quang học
  + Tesseract OCR: mã nguồn mở về nhận dạng ký tự quang học
  + tess-two: một phần của công cụ Tesseract OCR cung cấp các API về nhận dạng ký tự quang học cho riêng nền tảng Android.
  + Realm: cơ sở dữ liệu trên nền tảng di động, thay thế cho các cơ sở dữ liệu truyền thống
  + Tra cứu từ điển

### Phạm vi đề tài

Trong phạm vi giới hạn về thời gian thực hiện đồ án nên em triển khai ứng dụng trong phạm vi giới hạn như sau: ứng dụng sẽ sử dụng thư viện bên thứ ba của mã nguồn mở Tesseract OCR là tess-two để thực hiện nhận dạng ký tự từ ảnh chụp được trên điện thoại hệ điều hành Android và sử dụng dữ liệu trong file dữ liệu từ điển mở Lingo để tra cứu từng từ một từ kết quả nhận dạng được từ ngôn ngữ Tiếng Anh sang ngôn ngữ Tiếng Việt.

## Phương pháp phát triển



### Nghiên cứu lý thuyết

Thu thập, phân tích và đánh giá các tài liệu liên quan đến OCR, Tesseract OCR, tess-two, xử lý camera, xử lý đa luồng, xử lý ảnh, lưu trữ, xử lý file dữ liệu từ điển.

### Các bước thực hiện đề tài

Các bước thực hiện để hoàn thành đề tài:

1 Xử lý camera để chụp ảnh trong lập trình Android

2 Ứng dụng thư viện bên thứ ba của Tesseract OCR là tess-two để nhận dạng ký tự quang học từ ảnh chụp được.

3 Phân tích xử lý file dữ liệu từ điển để tạo cơ sở dữ liệu phục vụ cho chức năng tra cứu từ bằng cách nhập bằng tay.

4 Sử dụng kết quả nhận dạng được từ ảnh chụp được và dữ liệu trong cơ sở dữ liệu để loại bỏ các từ không đúng chính tả và cho ra kết quả

5 Đánh giá kết quả.

### Phương pháp thực nghiệm

Hệ thống sẽ được triển khai trên các điện thoại hệ điều hành Android phiên bản 4.0 hoặc mới hơn.

# Chương 2: CƠ SỞ LÝ THUYẾT



## Android

### Tổng quan về Android

Android là một Hệ điều hành mã nguồn mở và là một hệ điều hành dựa trên Linux cho các thiết bị mobile như Smartphone và máy tính bảng. Ban đầu Android được phát triển bởi Công ty Android với sự hỗ trợ tài chính từ Google, sau đó được Google mua lại vào năm 2005.

Android đưa ra một phương pháp thống nhất để phát triển ứng dụng cho các thiết bị di động, nghĩa là các lập trình viên chỉ cần phát triển Android, và các ứng dụng khác có thể chạy trên các thiết bị khác nhau mà đã được trang bị Android.

Phiên bản beta của Android Software Development Kit (SDK) được công bố bởi Google vào năm 2007, trong khi phiên bản thương mại đầu tiên Android 1.0 được công bố 9/2008.

### Các đặc điểm của Android

Android là một hệ điều hành mạnh mẽ, và đang cạnh tranh mạnh mẽ với Apple 4GS và hỗ trợ nhiều tính năng tuyệt vời. Dưới đây liệt kê một số đặc điểm đó:

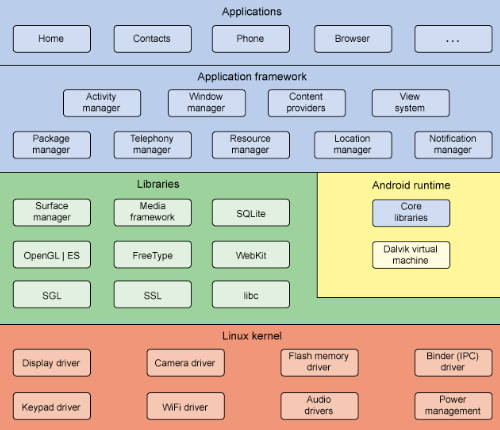
|  |  |
| --- | --- |
| Đặc điểm | Miêu tả |
| UI đẹp | Màn hình Android OS cơ bản cung cấp một giao diện người dùng đẹp và có tính thẩm mỹ cao |
| Connectivity | GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC và WiMAX. |
| Lưu trữ | SQLite, một relational database gọn nhẹ, được sử dụng cho mục đích lưu trữ dữ liệu |
| Hỗ trợ Media | H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, và BMP |
| Thông báo | SMS và MMS |
| Trình duyệt Web | Dựa trên thiết bị WebKit mã nguồn mở, đi kèm với thiết bị V8 JavaScript của Chrome hỗ trợ HTML5 và CSS3 |
| Multi-touch | Android hỗ trợ cho multi-touch mà đã được tạo ban đầu có sẵn cho các Handset như HTC Hero |
| Đa nhiệm | Người dùng có thể nhảy từ một tác vụ tới tác vụ khác và nhiều ứng dụng đa dạng có thể chạy đồng thời cùng một lúc |
| Widget tùy chỉnh | Widgets có thể thay đổi kích cỡ, vì thế người dùng có thể mở rộng để hiển thị nhiều nội dung hơn, hoặc thu nhỏ để tiết kiệm không gian |
| Đa ngôn ngữ | Hỗ trợ text đơn hướng và song hướng |
| GCM | Google Cloud Messaging (GCM) là một dịch vụ cho phép lập trình viên gửi thông điệp dữ liệu ngắn tới người dùng trên thiết bị Android, mà không cần một Sync Solution |
| Wi-Fi Direct | Một công nghệ cho phép các ứng dụng dò tìm và ghép cặp một cách trực tiếp, thông qua một kết nối peer-to-peer |
| Android Beam | Một công nghệ dựa trên NFC phổ biến cho phép người dùng chia sẻ tức thì, chỉ cần kích hoạt NFC của hai điện thoại với nhau |

Bảng 2.1: Đặc điểm của Android



### Cấu trúc của Android

Hệ điều hành Android là một ngăn xếp của các thành phần phần mềm mà có thể đại khái phân chia thành 5 khu vực và 4 lớp chính. Hình dưới đây minh họa sơ đồ cấu trúc Android.



Hình 2.1: Cấu trúc Android

* Lớp Linux Kernel trong Android: Linux Kernel là lớp thấp nhất. Nó cung cấp các chức năng cơ bản như quản lý tiến trình, quản lý bộ nhớ, quản lý thiết bị như: Camera, bàn phím, màn hình, … Ngoài ra, nó còn quản lý mạng, driver của các thiết bị, điều này gỡ bỏ sự khó khăn về giao tiếp với các thiết bị ngoại vi.
* Libraries trong Android: Phía trên Linux Kernel là tập hợp các bộ thư viện mã nguồn mở WebKit, bộ thư viện nổi tiếng libc, cơ sở dữ liệu SQLite hữu ích cho việc lưu trữ và chia sẻ dữ liệu, bộ thư viện thể phát, ghi âm về âm thanh, hoặc video. Thư viện SSL chịu trách nhiệm cho bảo mật Internet.
* Android Libraries: Phần này gồm các thư viện dựa trên Java. Nó bao gồm các Framework Library giúp xây dựng, vẽ đồ họa và truy cập cơ sở dữ liệu trở nên dễ dàng hơn.
* Android Runtime: Đây là thành phần thứ 3 trong cấu trúc, thuộc về lớp 2 tính từ dưới lên. Phần này cung cấp một thành phần quan trọng gọi là **Dalvik Virtual Machine** là một máy ảo Java đặc biệt, được thiết kế tối ưu cho Android.
* Application Framework: Lớp Application Framework cung cấp nhiều dịch vụ cấp cao hơn cho các ứng dụng trong các lớp Java. Các lập trình viên cũng được phép sử dụng các dịch vụ này trong các ứng dụng của họ.
* Applications: Bạn sẽ thấy tất cả các ứng dụng Android ở lớp trên cùng. Ứng dụng bạn viết sẽ được cài đặt vào lớp này. Ví dụ của những ứng dụng này là Contacts Books, Browser, Games, …

## OCR



### Tổng quan về OCR

Nhận dạng ký tự quang học (tiếng Anh: Optical Character Recognition, viết tắt là OCR), là loại phần mềm máy tính được tạo ra để chuyển các hình ảnh của chữ viết tay hoặc chữ đánh máy (thường được quét bằng máy scanner) thành các văn bản tài liệu. OCR được hình thành từ một lĩnh vực nghiên cứu về nhận dạng mẫu, trí tuệ nhận tạo và machine vision. Mặc dù công việc nghiên cứu học thuật vẫn tiếp tục, một phần công việc của OCR đã chuyển sang ứng dụng trong thực tế với các kỹ thuật đã được chứng minh.

Nhận dạng ký tự quang học (dùng các kỹ thuật quang học chẳng hạn như gương và ống kính) và nhận dạng ký tự số (sử dụng máy quét và các thuật toán máy tính) lúc đầu được xem xét như hai lĩnh vực khác nhau. Bởi vì chỉ có rất ít các ứng dụng tồn tại với các kỹ thuật quang học thực sự, bởi vậy thuật ngữ Nhận dạng ký tự quang học được mở rộng và bao gồm luôn ý nghĩa nhận dạng ký tự số.

Đầu tiên hệ thống nhận dạng yêu cầu phải được huấn luyện với các mẫu của các ký tự cụ thể. Các hệ thống "thông minh" với độ chính xác nhận dạng cao đối với hầu hết các phông chữ hiện nay đã trở nên phổ biến. Một số hệ thống còn có khả năng tái tạo lại các định dạng của tài liệu gần giống với bản gốc bao gồm: hình ảnh, các cột, bảng biểu, các thành phần không phải là văn bản

### Các ứng dụng của OCR

Các công cụ OCR đã được phát triển để ứng dụng vào nhiều lĩnh vực cụ thể như là:

* Nhập liệu cho các tài liệu trong kinh doanh: hóa đơn, passport, sao kê, biên lai…
* Nhận diện biển số xe một cách tự động
* Nhận dạng chữ viết tay trong thời gian thực để điều khiển máy tính
* Kiểm tra sự chính xác của hệ thống chống bot của CAPTCHA
* Công nghệ hỗ trợ cho người khiếm thị

### Các thể loại

Các thể loại của OCR:

* Optical character recognition (OCR) – targets typewritten text, one glyph or character at a time.
* Optical word recognition – targets typewritten text, one word at a time (for languages that use a space as a word divider). (Usually just called "OCR".)
* Intelligent character recognition (ICR) – also targets handwritten printscript or cursive text one glyph or character at a time, usually involving machine learning.
* Intelligent word recognition (IWR) – also targets handwritten printscript or cursive text, one word at a time. This is especially useful for languages where glyphs are not separated in cursive script.

### Các kỹ thuật

Các kỹ thuật trong OCR

* **Tiền xử lý**

Phần mềm OCR thường “tiền xử lý” các bức ảnh để nâng cao độ chính xác trong việc nhận dạng chữ, kỹ thuật bao gồm:

* De-skew
* Despeckle
* Binarisation
* Line removal
* Layout analysis or "zoning"
* Line and word detection
* Script recognition
* Character isolation or "segmentation"
* Normalise aspect ratio and scale
* **Nhận dạng ký tự**
* **Hậu xử lý**

### Độ chính xác

Các hệ thống nhận dạng ký tự viết tay đã đạt được những thành công lớn về mặt thương mại trong những năm gần đây. Trong số đó là thiết bị nhập cho những thiết bị hỗ trợ cá nhân (PDA) như những phần mềm chạy trên Palm OS. hãng Apple Newton đi tiên phong trong công nghệ này. Những giải thuật sử dụng trong những thiết bị này sử dụng những ưu điểm rằng thứ tự, tốc độ, và hướng của những đoạn dòng đơn lẻ đã được biết trước. Tương tự, người dùng có thể được yêu cầu sử dụng chỉ một vài loại kiểu chữ nhất định. Những phương pháp này không thể dùng được trong phần mềm scan tài liệu giấy, do đó sự nhận dạng chính xác văn bản in bằng tay vẫn là một vấn đề lớn đang được bỏ ngỏ. Với mức chính xác từ 80% đến 90%, những ký tự in bằng tay sạch sẽ có thể được nhận ra, nhưng độ chính xác đó vẫn tạo ra hàng tá lỗi mỗi trang, khiến cho công nghệ đó chỉ hiệu quả trong vài trường hợp nào đó. Sự đa dạng của OCR hiện nay được biết đến trong công nghiệp là ICR, (Intelligent Character Recognition - Nhận dạng Ký tự Thông minh).

Nhận dạng chữ viết tay là một lĩnh vực nghiên cứu sôi nổi, với tỷ lệ nhận dạng thậm chí còn thấp hơn cả văn bản in bằng tay. Tỷ lệ nhận dạng cao hơn của những bản viết tay chung chung hầu như là không thể nếu không sử dụng thông tin về ngữ pháp và văn cảnh. Ví dụ như, nhận dạng cả một chữ từ một cuốn từ điển thì dễ hơn là việc cố gắng lấy ra những ký tự rời rạc từ đoạn đó. Đọc dòng Tổng cộng của một tờ séc (luôn luôn được viết bằng số) là một ví dụ trong đó sử dụng những từ điển nhỏ hơn có thể tăng tỷ lệ nhận dạng rất nhiều. Kiến thức về ngữ pháp của một ngôn ngữ được scan cũng có thể giúp xác định một từ có thể là động từ hay danh từ, ví dụ như vậy, sẽ cho phép độ chính xác cao hơn. Hình dạng của chữ viết tay bản thân nó đã không chứa đủ thông tin về để nhận dạng chính xác (hơn 98%) tất cả những đoạn chữ viết tay.

## Tesseract OCR

**Gioi thieu**

Tesseract là một mã nguồn mở về công cụ nhận dạng ký tự quang học (OCR) mà người dùng có thể sử dụng dứoi giấy phép 2.0. Mã nguồn mở này người dùng có thể sử dụng một cách trực tiếp hoặc dành cho các lập trình viên thông qua việc sử dụng các để nhận dạng dữ liệu văn bản đánh máy, viết tay từ các bức ảnh. Tesseract hỗ trợ rất nhiều các ngôn ngữ khác nhau.

Hiện tại Tesseract giao diện người dùng được xây dựng sẵn, nhưng có rất nhiều các bên thứ 3 hỗ trợ cho việc này

**Cai dat**

Có 2 phần chính để cài đặt là: phần cài đặt công cụ và phần dữ liệu training cho một ngôn ngữ.

**Cài đặt trên Linux**

Tesseract hiện có thể cài đặt trực tiếp từ các bên phân phối Linux. Gói cài đặt được đặt tên là “tesseract” hoặc “tesseract-ocr” có thể được tìm thấy trên các kho chứa của bên phân phối Linux. Dữ liệu training có thể kèm theo khi cài đặt công cụ, nhưng nếu không có chúng ta có thể tải về dữ liệu training của một ngôn ngữ tại “<https://github.com/tesseract-ocr/tessdata>”, sau khi tải về dữ liệu này thì giải nén và sao chép file dữ liệu .traineddata vào thư mục có tên “tessdata”, thông thường thì là thư mục “/usr/share/tesseract-ocr/tessdata or /usr/share/tessdata”

Nếu Tesseract không có trên bên phát hành Linux hiện tại của máy hoặc chúng ta muốn có phiên bản mới nhất thì có thể biên dịch ra một công cụ tesseract cho riêng mình tại theo hướng dẫn tại “<https://github.com/tesseract-ocr/tesseract/wiki/Compiling>”

**Cài đặt trên macOS**

Chúng ta có thể cài Tesseract trên macOS bằng cách sử dụng MacPorts hoặc HomeBrew - hai công cụ giúp cài đặt Tesseract trên macOS

MacPorts

Để cài đặt Tesseract bằng MacPorts thì chạy dòng lệnh này:

sudo port install tesseract

Để cài đặt dữ liệu training về một ngôn ngữ bất kỳ thì thực thi dòng lệnh này:

sudo port install tesseract-<langcode>

trong đó langcode có thể được tìm thấy trong danh sách tại trang “<https://www.macports.org/ports.php?by=name&substr=tesseract->”

Homebrew

Để cài đặt Tesseract với Homebrew thì thực thi dòng lệnh sau:

brew install tesseract

**Windows**

Chúng ta có thể cài đặt Tesseract cho hệ điều hành Windows bằng công cụ “UB Mannheim”, công cụ này đã kèm theo dữ liệu training

**Các nền tảng khác:**

Tesseract có để được cài đặt trên nhiều nền tảng khác ngoài trừ các nền tảng được đề cập ở trên. Chúng ta có thể biên dịch ra một công cụ Tesseract cho những mục đích riêng khác nhau hoặc có thể sử dụng các dự án đã dựng sẵn sử dụng Tesseract dứoi đây

Bảng … là danh sách các ứng dụng có giao diện người dùng và những dự án khác của các nhà phát triển bên thứ 3 sử dụng Tesseract OCR :

## GUI

| **Tên** | **Linux** | **Mac** | **Windows** | **Giấy phép** | **Mô tả** |
| --- | --- | --- | --- | --- | --- |
| [Free-Ocr-Windows-Desktop](https://github.com/A9T9/Free-Ocr-Windows-Desktop/releases) |  |  | X | [GNU AGPL v3](https://github.com/A9T9/Free-Ocr-Windows-Desktop/blob/master/LICENSE) | Free OCR application for the Windows Desktop - Essentially a graphical user interface (GUI) for the Tesseract OCR engine. The application also includes support for reading and OCR'ing PDF files |
| [YAGF](http://sourceforge.net/projects/yagf-ocr/) | X |  |  | GPL v3 | A graphical front-end for cuneiform and tesseract |
| [gImageReader](https://github.com/manisandro/gImageReader) | X |  | X | GPL v3 | A graphical GTK frontend to tesseract-ocr |
| [SunnyPage OCR](http://www.sunnypage.ge/en/) |  |  | X | Proprietary | A GUI frontend for Tesseract OCR engine with automatic adjustment of image brightness, image processing and PDF support. |
| [VietOCR](http://vietocr.sourceforge.net/) | X | X | X | Apache 2.0 | A GUI frontend for Tesseract OCR engine. Supports optical character recognition for Vietnamese and other languages supported by Tesseract |
| [OCRFeeder](https://wiki.gnome.org/action/show/Apps/OCRFeeder) | X |  |  | GPL v3 | OCRFeeder is a document layout analysis and optical character recognition system |
| [PDF OCR X](http://solutions.weblite.ca/pdfocrx/) |  | X | X | Proprietary | PDF OCR is a simple drag-and-drop utility for Mac OS X and Windows, that converts your PDFs and images into text documents or searchable PDF files |
| [Lector](https://github.com/zdenop/lector) | X |  | X | GPL v2 | A graphical ocr solution for GNU/Linux based on Python, Qt4 and tessaract OCR |
| [Tesseract-OCR QT4 gui](https://github.com/zdenop/tesseract-ocr-qt4gui) | X |  |  | Apache 2.0 | Tesseract-OCR QT4 gui is a simple GUI for tesseract |
| [Lime OCR](http://code.google.com/p/lime-ocr/) |  |  | X | GPL v3 | A simple, free OCR software for Windows using tesseract-ocr engine |
| [Ocrivist](http://www.ocrivist.com/) | X |  |  | GPL v3 | Ocrivist is a utility which makes it possible to scan and OCR books and other printed documents to PDF or Djvu format |
| [Tesseract-GUI](http://tesseract-gui.sourceforge.net/) | X |  |  | GPL v2 | Tessract-GUI is not a front-end for tesseract-ocr, it is just a graphical way to use it with simple image manipulation through ImageMagick |
| [QTesseract](http://code.google.com/p/qtesseract/) | X |  |  | LGPL v3 | QT GUI for the Tesseract OCR |
| [TessOCR(KISI)](http://djvu.life.coocan.jp/TessOCR/doc/tessOCR_eng.html) |  | X |  | Apache 2.0 | A free OCR tool |
| [pmOCR](http://github.com/deajan/pmocr/) | X |  |  | BSD | Batch OCR tool, also file monitor event OCR with tesseract |
| [tesseract4java](http://github.com/tesseract4java/tesseract4java) | X | X | X | GPLv3 | A cross-platform GUI for training and running Tesseract with advanced features like batch recognition and accuracy evaluation |

Các dịch vụ OCR trưc tuyến:

* OCR.net: dịch vụ chuyển đổi PDF và ảnh thành TEXT
* WeOCR: một nền tảng cho các hệ thống web OCR cho phép mọi người sử dụng nhận dạng ký tự qua mạng
* CustomOCR
* Free OCR
* i2OCR
* Indic-OCR OCR Service: một dịch vụ OCR trực tuyến cho ngôn ngữ Ấn Độ
* Obserbot: dịch vụ OCR trực tuyến, từ điển, dịch vụ chuyển ngữ

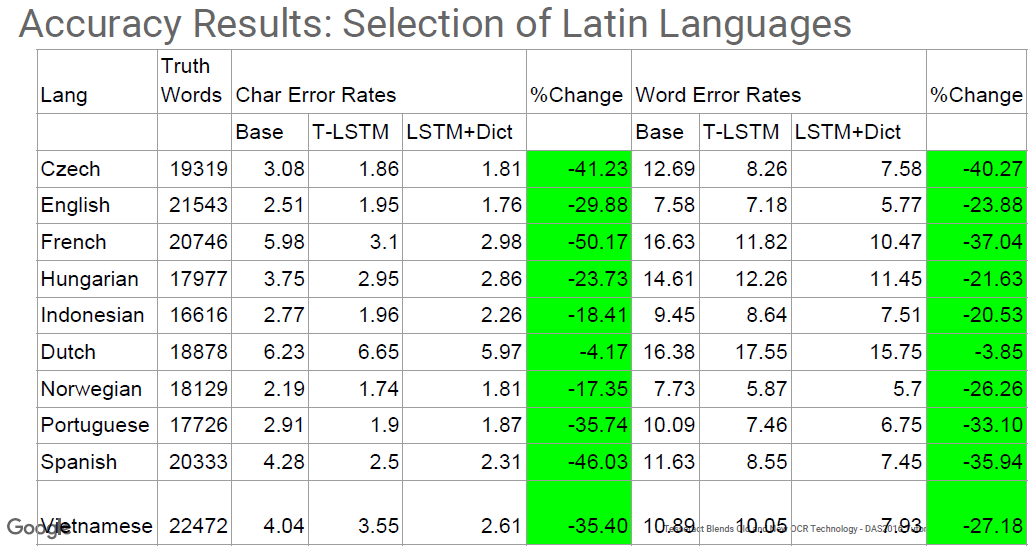
MOBILE:

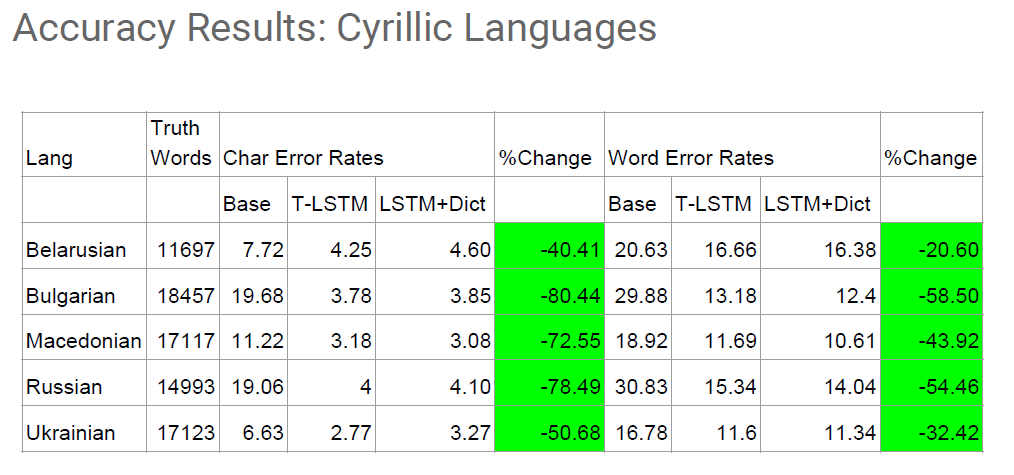
* Android:
  + tess-two: một phân nhánh của mã nguồn mở Tesseract OCR cho hệ điều hành Android trong đó có thêm các tính năng bổ sung.
  + Textfairy: một ứng dụng Android về nhận dạng ký tự quang học, source code tại: github.com
  + Character Recognition: một ứng dụng Android về nhận dạng ký tự quang học, source code tại: gitorious.org
  + tesseract-android-tools: tập các API về OCR cho Android
  + Across India: một ứng dụng có thể giúp người dùng chụp các bức ảnh từ các bảng tín hiện ngôn ngữ Ấn Độ hoặc Tiếng Anh và chuyển ngữ nó thành ngôn ngữ mà bạn có thể hiểu.
* IOS:
  + Tesseract-OCR-iOS: Tesseract-OCR-iOS là một framework cho iOS 7+, được biên dịch cho armv7s và arm64
  + OCR-iOS-Example: một ví dụ đơn giản về cách sử dụng nhận dạng kí tự quang học trên iOS
* Các hệ điều hành khác:
  + ScanBizCards: giải pháp di động về quét thẻ trong hoạt động kinh doanh

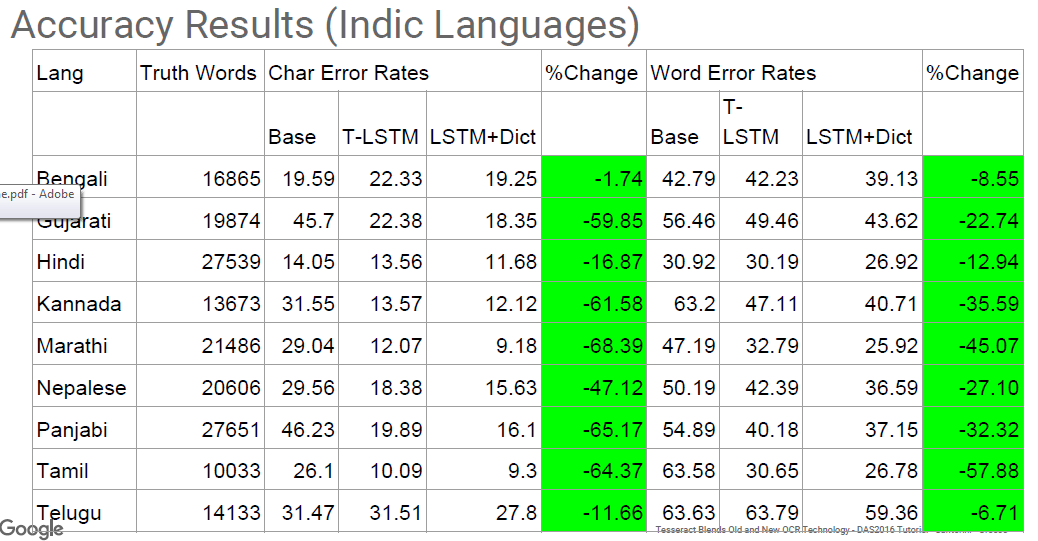
Các dự án khác:

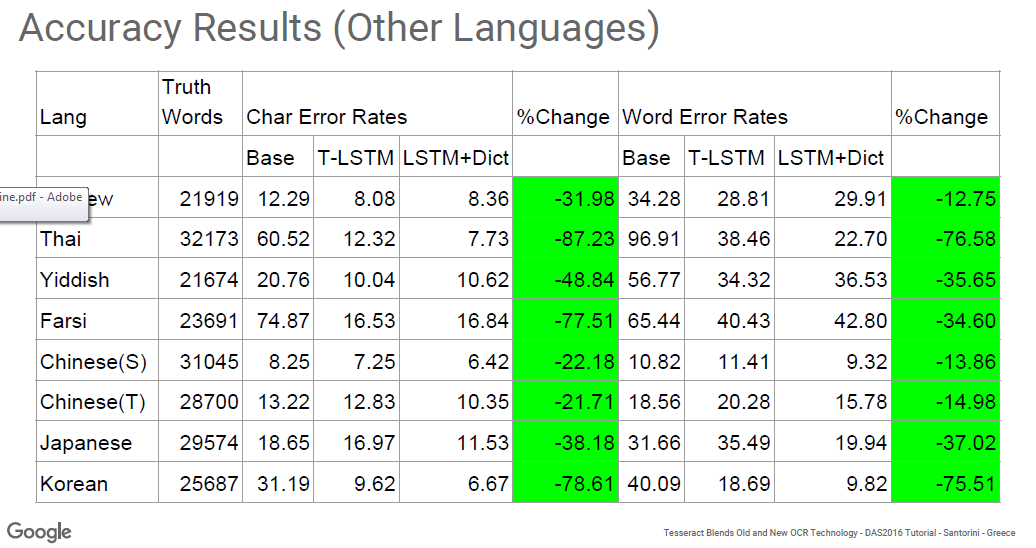
* [ocr-fileformat](https://github.com/UB-Mannheim/ocr-fileformat) - Validate and transform between OCR file formats (hOCR, ALTO, PAGE, FineReader)
* [Tess4J](https://github.com/nguyenq/tess4j) - A Java JNA wrapper for Tesseract OCR API.
* [Traineddata inspector](https://mazoea.com/te/traineddata/) - to inspect some of the internals of traineddata files
* [TopOCR](http://www.topocr.com/) - high Quality OCR for Cameras with tesseract-ocr support (paid product)
* [Simple OCR Web Server](https://github.com/ybur-yug/python_ocr_tutorial) using python, flask, tesseract-ocr, and leptonica
* [Display OCR](https://github.com/arturaugusto/display_ocr) is OpenCV-Python + python-tesseract real-time image preprocess and OCR of 7 segments font.
* [OpenOCR](https://github.com/tleyden/open-ocr) makes it simple to host your own OCR REST API.
* <https://github.com/guitarmind/tesseract-web-service> is An implementation of RESTful web service for tesseract-OCR using tornado
* [RasterEdge .NET Image SDK - OCR Recognition](http://www.rasteredge.com/dotnet-imaging/addon-ocr-sdk/) is robust, high-performance recognition application of royalty-free distribution for desktop or server applications.
* [DevScope OCR SDK](http://www.devscope.net/products/DevScopeOCR) is an Optical Character Recognition toolkit engine based on Tesseract OCR v3 that allows to develop applications using Microsoft .NET framework
* [Paperwork](https://github.com/jflesch/paperwork) - using OCR to grep dead trees the easy way (requires pyocr)
* [Aletheia](http://www.primaresearch.org/tools.php) - An Advanced Document Layout and Text Ground-Truthing System for Production Environments
* [gscan2pdf](http://gscan2pdf.sourceforge.net/) a GUI to produce PDFs or DjVus from scanned documents
* [Audiveris](http://audiveris.kenai.com/) is an open-source Optical Music Recognition software which processes the image of a music sheet to automatically provide symbolic music information in MusicXML standard.
* [Ocrivist](https://code.google.com/p/ocrivist/) is a utility which makes it possible to scan and OCR books and other printed documents to PDF or Djvu format.
* [thu-ipv6-login](https://code.google.com/p/thu-ipv6-login/) a python script for IPv6 authentication in Tsinghua University with support for OCR of authcode
* [Wolfram Mathematica 9.0](http://www.wolfram.com/mathematica/) use tesseract for [recognizing text](https://groups.google.com/d/msg/tesseract-ocr/NmxFclHcsAE/-KaiT5oJ8oQJ)
* [node-dv](https://github.com/creatale/node-dv) is a node.js library for processing and understanding scanned documents
* [hocr-tools](https://github.com/tmbdev/hocr-tools) - python tools for manipulating and evaluating the hOCR format for representing multi-lingual OCR results by embedding them into HTML. They include hocr-pdf tool for creating searchable pdf.
* [PyPDFOCR](https://github.com/virantha/pypdfocr) - Tesseract-OCR based PDF filing
* [OCRmyPDF](https://github.com/jbarlow83-hh/OCRmyPDF) - Adds OCR text layer to scanned PDF files and images, allowing them to be searched. Processes pages in parallel on multi-core CPUs. Keeps exact resolution of original embedded images without recompressing JPEGs, when possible. Includes image several preprocessing options, detailed documentation, and support for many exotic PDFs.
* [ChronoScan](http://www.chronoscan.org/) is a complete suite for document Scanning & Data Entry
* [speedy-ocr](http://www.donaldmarang.org/speedy-ocr.html) utility to simplify scanning and OCR focus to help blind and visually impaired community. It is part of [Vinux project](http://vinuxproject.org/).
* [Project VIRAL](http://apps.man.poznan.pl/trac/varico/wiki) Varico Invoice Recognition with Assisted Learning
* [Bindery](http://blender3d.github.com/Bindery/): A simple GUI for binding post processed scanned pages into digital documents
* [Clarify](http://code.google.com/p/clarify): Clarify helps you OCR 'image-only' PDFs. Your input is a PDF that you normally cannot extract text from. The output is text. Clarify is a python module that wraps up tesseract-ocr, xpdf and netpbm. Requirements: python, tesseract-ocr, xpdf, netpbm
* [hOcr2Pdf.NET](http://hocrtopdf.codeplex.com/documentation): hOcr2Pdf.NET is a library that programmers can use to create highly compressed, searchable pdf's for applications. Requirements: .NET 2.0 or higher, Tesseract 3.0, JBig2.exe
* [PDFBeads](http://rubyforge.org/projects/pdfbeads): convert scanned images to a single searchable PDF file based on hOCR files. Requirements: ruby, RMagick, hpricot
* [ExactImage/hocr2pdf](http://www.exactcode.com/site/open_source/exactimage/hocr2pdf/): creates a Searchable PDF from hOCR input. Requirements: libagg
* [HocrConverter](https://github.com/jbrinley/HocrConverter): creates PDFs and plain text from hOCR documents. Requirements: python, reportlab
* [HocrToPdf.java](http://www.acoveo.com/acoveo/files/HocrToPdf.java): java source for very basic hOCR to PDF converter. Compiled version can be found at project [modi2hocr](http://code.google.com/p/modi2hocr/source/browse/trunk/). Requirements: java, jericho, iText2
* [hOcr2Pdf.NET](http://hocrtopdf.codeplex.com/): is a .NET library to convert .hocr html produced by Tesseract or Cuneiform into searchable pdfs using HtmlAgilityPack and iTextSharp. Requirements: C#.
* [Tally-Ho](http://code.google.com/p/tallyho/): Tally-Ho is a screen reader intended for sites like google books
* [Mayan EDMS](http://rosarior.github.com/mayan/): Document management system with tesseract as it's base
* [Olena](http://git.lrde.epita.fr/?p=olena.git;a=summary): a generic and efficient image processing platform (tesseract is used in its part called [scribo](http://git.lrde.epita.fr/?p=olena.git;a=tree))
* [ocrodjvu](http://jwilk.net/software/ocrodjvu) is a wrapper for OCR systems, that allows you to perform OCR on DjVu files
* [PaRADIIT](https://sites.google.com/site/paradiitproject/home) (Pattern Redundancy Analysis for Document Image Indexation & Transcription) is a project initiated and sponsored by 2 successive Google DH awards. It aims to turn ancient books, especially from the Renaissance, into accessible digital libraries.
* [The ISRI Analytic Tools](https://github.com/eddieantonio/isri-ocr-evaluation-tools) consist of 17 tools for measuring the performance of and experimenting with OCR output.
* [pdf2pdfocr](https://github.com/LeoFCardoso/pdf2pdfocr) is a tool to OCR a PDF (or supported images) and add a text layer in the original file making it a searchable PDF. It is a python script that uses tesseract and other open source tools. Linux, macOS and Windows supported.
* [Indic Messenger](https://indic-ocr.github.io/indicmessenger) A Facebook chat bot which can OCR images containing Indian/English text and transliterate it to other Indian scripts.
* [LibreOCR](https://indic-ocr.github.io/LibreOCR/) A [LibreOffice](http://www.libreoffice.org/) extension which can convert an image to OCT and open in the Writer application.

ĐỘ CHÍNH XÁC VÀ HIỆU SUẤT









SỬ DỤNG LỆNH

## Tess-two



### Tổng quan

Tess-two là một nhánh của Tesseract Tools cho Android (tesseract-android-tools) và thêm một vài chức năng bổ sung. Tesseract Tools cho Android là một tập các APIs và các build files cho Tesseract OCR và các thư viện xử lý ảnh

* Tess-two có thể hoạt động với:
  + Tesseract 3.05
  + Leptonica 1.74.1
  + libjpeg 9b
  + libpng 1.6.25
* Mã nguồn của tess-two được để dưới thư mục: tess-two/jni
* Module tess-two chứ các công cụ để biên dịch Tesseract và các thư viện Leptonica sử dụng trên nền tảng Android. Chúng cung cấp các Java API để biên dịch Tesseract và các thư viện Leptonica

### Yêu cầu cần thiết

* Hệ điều hành Android 2.3 hoặc có phiên bản cao hơn
* Một ‘trained data file’ cho một ngôn ngữ. Các file dữ liệu phải được chép vào thiết bị Android dưới một thư mục con có tên là “tessdata”

### Cách sử dụng

Để sử dụng tess-two vào trong ứng dụng Android, chúng ta chỉnh sửa file build.gradle của module app bằng cách thêm một dependency ở bên ngoài như sau:

dependencies {

compile 'com.rmtheis:tess-two:6.3.0'

}

### Ứng dụng trong đề tài

* Trong ứng dụng tra từ điển, **tess-two** được sử dụng để nhận diện hình ảnh chụp được và cho ra kết quả từ nhận dạng được dưới dạng String.

## Realm



### Tổng quan

* The Realm Mobile Platform:  
  Khái niệm cốt lõi của The Realm Mobile Platform là một vật chứa dữ liệu hạng nhẹ với tên gọi ‘Realm’. Giống như một cơ sở dữ liệu, dữ liệu trong Realm có thể được truy vấn, lọc, liên kết … Không giống như cơ sở dữ liệu truyền thống mặc dù các objects trong Realm thực hiện đầy đủ các chức năng như cơ sở dữ liệu truyền thống, Realm đồng bộ hóa liền mạch giữa thiết bị và ứng dụng và có thể được truy cập một cách an toàn qua các luồng.
* The Realm Mobile Database  
  Cơ sở dữ liệu Realm được xây dựng từ nền tảng để chạy trên các thiết bị di động. Không giống như các cơ sở dữ liệu truyền thống, các đối tượng trong Realm là các “native objects”. Chúng ta không phải sao chép dữ liệu ra khỏi cơ sở dữ liệu, chỉnh sửa và lưu lại – chúng ta luôn làm việc trực tiếp với các objects trong Realm. Nếu một tiến trình chỉnh sửa thông tin một object trong Realm thì ngay lập tức các tiến trình các có thể được thông báo về sự thay đổi

### Yêu cầu cần thiết

* Android studio phiên bản 1. 5 hoặc cao hơn
* JDK phiên bản 7.0 hoặc cao hơn
* Phiên bản hiện tại của Android SDK
* Android phiên bản 2.3 hoặc cao hơn

### Cách sử dụng

Cách cài đặt Realm trong Android Project

**Bước 1**: thêm một dependency vào file build.gradle cấp độ project

buildscript **{**

repositories **{**

jcenter**()**

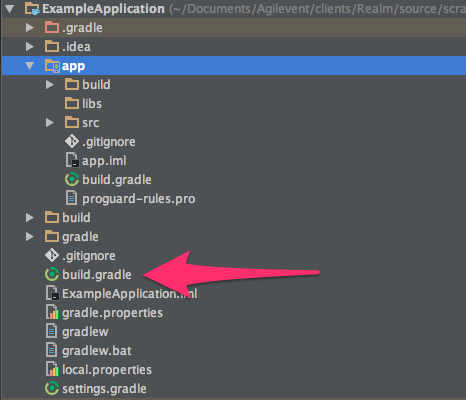
**}**

dependencies **{**

classpath "io.realm:realm-gradle-plugin:3.1.4"

**}**

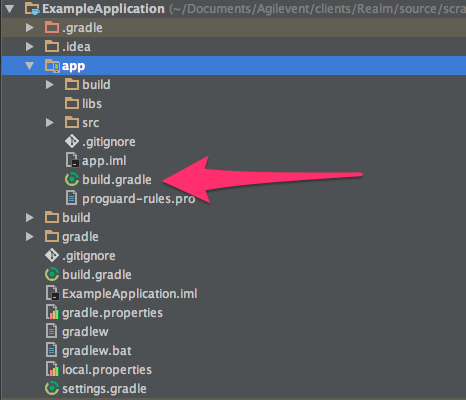
**}**



Hình 2.2: File build.gradle (Project)

**Bước 2**: Apply realm-android plugin vào trên cùng của file build.gradle cấp độ application

apply **plugin:** 'realm-android'



Hình 2.3: File build.gradle (Application)

### Ứng dụng trong đề tài

Trong ứng dụng tra từ điển, Realm được sử dụng để lưu trữ bộ dữ liệu từ điển và sử dụng một số hàm dựng sẵn có trong Realm để thực hiện một số các thao tác truy vấn cơ bản một cách nhanh chóng như: tìm kiếm, thêm dữ liệu, xóa dữ liệu, sắp xếp …

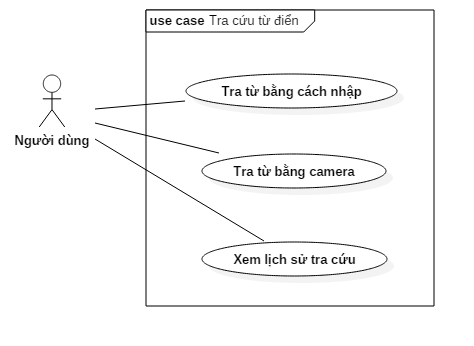
# Chương 3: ỨNG DỤNG TRA TỪ ĐIỂN ANH-VIỆT SỬ DỤNG OCR ĐỂ NHẬN DẠNG KÝ TỰ QUANG HỌC



## Mô tả hệ thống



### Yêu cầu chức năng



Hình 3.1: Usecase - Tra cứu từ điển

Tác nhân: Người dùng

Chức năng:

* Tra cứu bằng cách nhập: người dùng nhập từ cần tìm kiếm vào ô tìm kiếm và nhấn “Tìm kiếm” để tra cứu từ
* Tra cứu từ bằng camera: người dùng tra cứu từ bằng cách dùng camera để quét từ cần tra và nhấn “Chụp”, hệ thống sẽ phân tích và cho ra các từ nhận dạng được từ bức ảnh vừa chụp, người dùng kích chọn từ cần tra cứu để xem nghĩa
* Xem lịch sử tra cứu: các từ đã tra được lưu lại thành một danh sách và người dùng có thể xem lại lịch sử tra cứu này

|  |  |
| --- | --- |
| Tên Usecase | Tra từ bằng cách nhập |
| Mã Usecase | UC01 |
| Yêu cầu cấp cao | Cho phép người dùng tra cứu từ bằng cách nhập từ khóa vào ô tìm kiếm |
| Tác nhân | Người dùng |
| Mô tả | Người dùng nhập từ khóa vào ô tìm kiếm sau đó nhấn “Tìm kiếm”, hệ thống sẽ truy vấn cơ sở dữ liệu và cho ra kết quả cần truy vấn |
| Điều kiện trước | Người dùng mở màn hình tra cứu bằng cách nhập tay |
| Điều kiện sau |  |

Bảng 3.1: Mô tả chi tiết usecase tra từ bằng cách nhập tay

|  |  |
| --- | --- |
| Tên Usecase | Tra từ bằng camera |
| Mã Usecase | UC02 |
| Yêu cầu cấp cao | Cho phép người dùng tra cứu từ bằng cách chụp ảnh và chọn từ cần tra |
| Tác nhân | Người dùng |
| Mô tả | Người dùng chụp ảnh có chữ quang học, hệ thống sẽ dựa vào tess-two để nhận diện kí tự quang học, sau đó sẽ dùng thuật toán loại nhiễu để nâng cao kết quả nhận diện, sau đó người dùng chọn từ cần tra để xem nghĩa của từ cần tra. |
| Điều kiện trước | Người dùng mở màn hình tra cứu bằng Camera |
| Điều kiện sau |  |

Bảng 3.2: Mô tả chi tiết usecase tra từ bằng camera

|  |  |
| --- | --- |
| Tên Usecase | Xem lịch sử tra cứu |
| Mã Usecase | UC03 |
| Yêu cầu cấp cao | Cho phép người dùng xem lại lịch sử tra cứu |
| Tác nhân | Người dùng |
| Mô tả | Người dùng mở màn hình kết quả tra cứu để xem lại các từ mình đã tra |
| Điều kiện trước | Người dùng mở màn hình xem kết quả tra cứu |
| Điều kiện sau |  |

Bảng 3.3: Mô tả chi tiết usecase xem lịch sử tra cứu

### Yêu cầu phi chức năng

Yêu cầu phi chức năng của ứng dụng:

* Chương trình phải chạy ổn định
* Giao diện đơn giản dễ sử dụng và thân thiện với người dùng

### Yêu cầu hệ thống

Yêu cầu môi trường để chạy ứng dụng

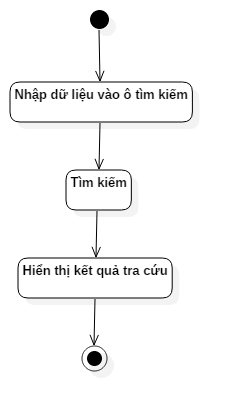
* Hệ điều hành Android phiên bản mới hơn 4.0
* Hệ thống đủ dung lượng để lưu trữ file dữ liệu từ điển

## Phân tích thiết kế



### Biểu đồ hoạt động

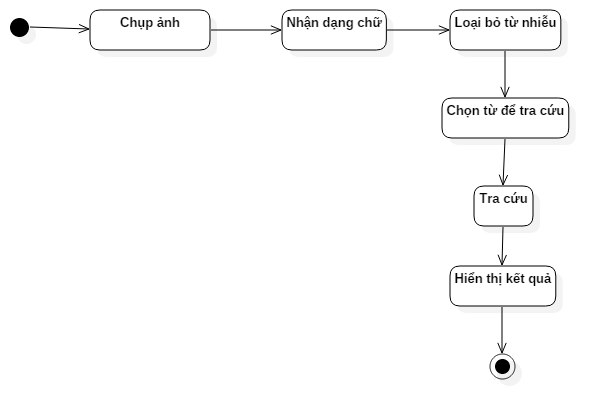
Tra cứu bằng cách nhập từ



Hình 3.2: Sơ đồ hoạt động tra cứu từ bằng cách nhập tay

* Nhập dữ liệu vào ô tìm kiếm: người dùng mở màn hình tra từ bằng cách nhập và nhập từ khóa vào ô tìm kiếm sau đó nhấn ‘Tìm kiếm’
* Tìm kiếm: hệ thống sẽ tìm kiếm kết quả theo từ khóa người dùng nhập vào
* Hiển thị kết quả: ứng dụng hiển thị kết quả của từ cần tra sau khi tìm kiếm trong cơ sở dữ liệu của ứng dụng.

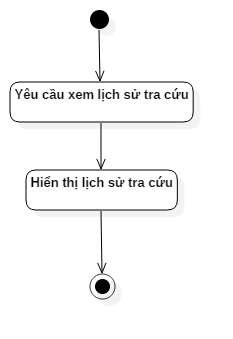
Tra cứu bằng camera



Hình 3.3: Sơ đồ hoạt động tra cứu từ bằng Camera

* **Chụp ảnh**: Người dùng mở màn hình tra từ bằng camera sau đó chụp ảnh có chứa từ cần tra cứu
* **Nhập dạng chữ**: sau khi người dùng chụp ảnh xong, ứng dụng sẽ thực hiện nhận dạng các chữ có trong ảnh chụp được và cho ra kết quả là một chuỗi các chữ nhận dạng được
* **Loại bỏ từ nhiễu** (từ không đúng chính tả): sau khi có kết quả từ việc nhận dạng, ứng dụng tiếp tục tra cứu từng từ trong chuỗi kết quả nhận dạng và loại bỏ các từ không có trong cơ sở dữ liệu từ điển, cuối cùng cho ra kết quả nhận dạng đã được loại bỏ các từ sai chính tả
* **Chọn từ để tra cứu**: kết quả nhận dạng sau khi đã được loại bỏ các từ nhiễu thì sẽ hiện lên màn hình và cho phép người dùng chạm vào từng từ để tra cứu nghĩa
* **Tra cứu**: khi người dùng chạm vào từ cần tra cứu, ứng dụng sẽ tra cứu từ đó trong cơ sở dữ liệu
* **Hiển thị kết quả**: sau khi tra cứu theo từ khóa người dùng chọn

Xem lịch sử tra cứu



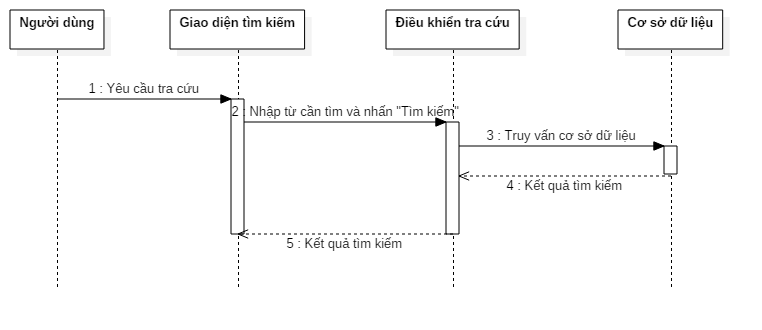
Hình 3.4: Sơ đồ hoạt động xem lịch sử tra cứu từ

* Yêu cầu xem lịch sử tra cứu: người dùng mở màn hình xem lịch sử tra cứu
* Hiến thị lịch sử tra cứu: ứng dụng hiển thị danh sách các từ mà người dùng đã tra cứu.

### Biểu đồ tuần tự

a Tra cứu bằng cách nhập bằng tay

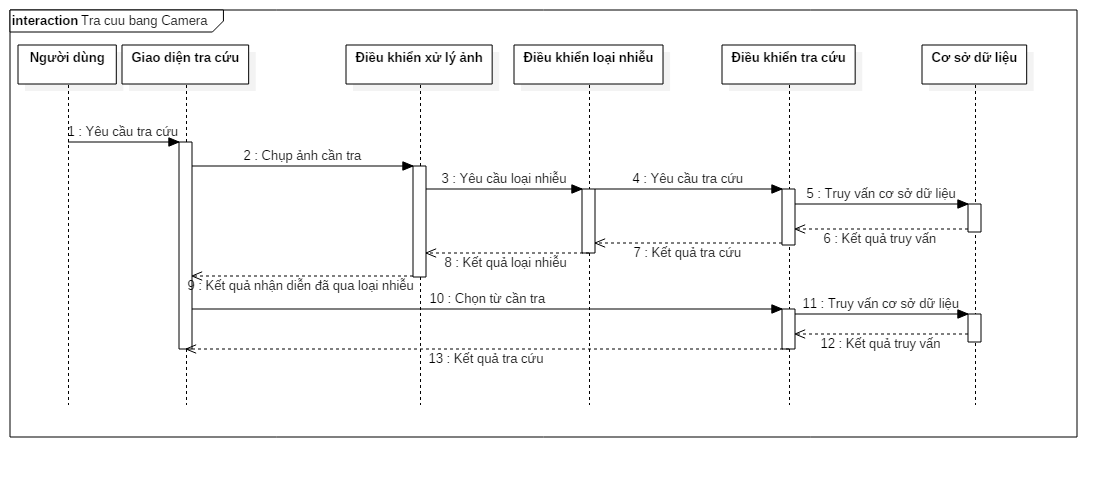
Hình 3.5 là sơ đồ tuần từ tra cứu từ bằng cách nhập bằng tay. Người dùng mở giao diện tìm kiêm và nhập từ khóa cần tìm vào ô tìm kiếm sau đó nhấn nút “Tìm kiếm”, hệ thống sẽ truy vấn cơ sở dữ liệu theo từ khóa và cho ra kết quả tìm kiếm.



Hình 3.5: Sơ đồ tuần tự tra cứu từ bằng cách nhập tay

b Tra cứu bằng Camera

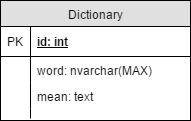
Hình 3.6 là sơ đồ tuần tự mô tả việc tra cứu từ bằng cách nhận diện chữ quang học từ ảnh chụp trên camera của điện thoại. Theo sơ đồ ta có thể thấy người dùng mở giao diện tra cứu từ bằng ảnh chụp, chụp lại ảnh có chữ cần tra, sau đó ứng dụng sẽ nhận diện chữ quang học có trong ảnh chụp và cho ra kết quả là một chuỗi ký tự. Từ chuỗi ký tự nhận dạng được, ứng dụng sẽ loại bỏ từ nhiễu bằng các bóc tách các chữ có trong chuỗi ký tự và tìm kiếm trong cơ sở dữ liệu từ điển, nếu không tìm thấy kết quả trong cơ sở dữ liệu thì từ đó sẽ bị loại bỏ. Sau khi loại bỏ các từ sai chính tả ứng dụng sẽ trả về chuỗi kết quả ra màn hình. Từ chuỗi kết quả này, người dùng có thể chạm vào từng chữ để tra cứu nghĩa của từ đó.



Hình 3.6: Tra cứu từ bằng cách nhận diện ảnh chụp từ Camera

### Thiết kế cơ sở dữ liệu

Hình 3.7 mô tả cơ sở dữ liệu từ điển của ứng dụng, bao gồm 3 trường: id, word và mean. Phần mô tả cụ thể từng trường được mô tả trong bảng 3.4



Hình 3.7: Thiết kế cơ sở dữ liệu

Bảng dữ liệu từ điển

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên cột | Kiểu dữ liệu | NULL | Khóa | Mô tả |
| 1 | id | INT | Không | Khóa chính | Mã từ |
| 2 | word | NVARCHAR(MAX) | Không | Không | Từ |
| 3 | mean | TEXT | Không | Không | Nghĩa của từ  (Định dạng HTML) |

Bảng 3.4: Bảng mô tả chi tiết cơ sở dữ liệu

# Chương 4: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

Nội dung chương này sẽ trình bày môi trường triển khai, các màn hình ứng dụng và phần cuối cùng sẽ trình bày về đánh giá ứng dụng



## Kết quả Demo

Phần này sẽ trình bày về các màn hình trong ứng dụng

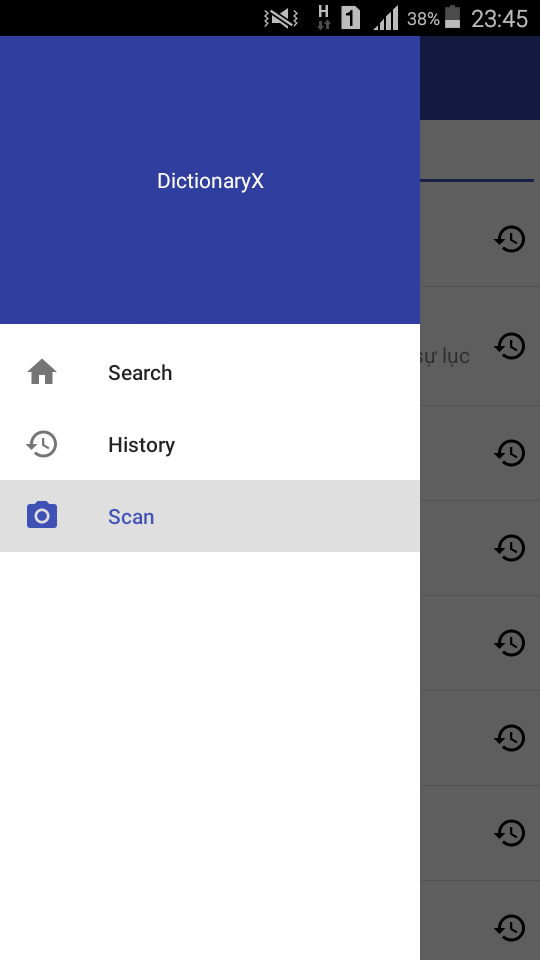
### Menu ứng dụng

Hình 4.1 mô tả màn hình Menu ứng dụng. Menu ứng dụng có 3 items chính đó là:

+ Search: người dùng nhấn vào menu item này để thực hiện chức năng tra cứu từ bằng cách nhập bằng tay

+ History: người dùng nhấn vào menu item này để thực hiện chức năng xem lịch sử tra cứu từ

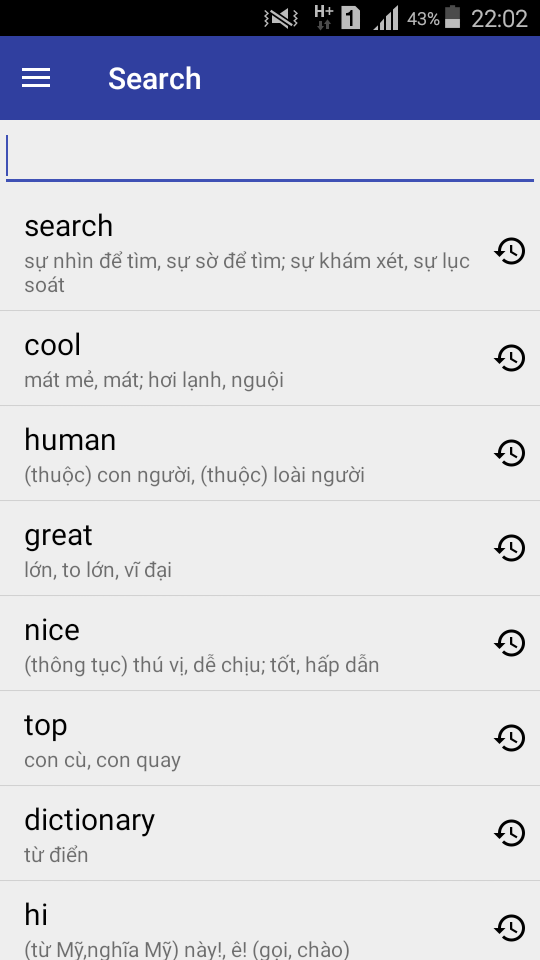
+ Scan: người dùng nhấn vào menu item này để thực hiện chức năng tra cứu từ bằng cách nhận diện ký tự quang học trong ảnh chụp từ camera của điện thoại



Hình 4.1: Menu ứng dụng

### Tra từ bằng cách nhập bằng tay

Hình 4.2 mô tả màn hình tra cứu từ bằng tay. Ở màn hình này người dùng có một ô textbox để người dùng nhập từ khóa tìm kiếm và phần dưới là danh sách các từ khóa trong mà người dùng đã tra cứu trước đây, từ danh sách này người dùng có thể thực hiện xem lại nội dung định nghĩa của các từ này bằng cách chọn một từ trong danh sách này.



Hình 4.2: Tra từ bằng các nhập tay

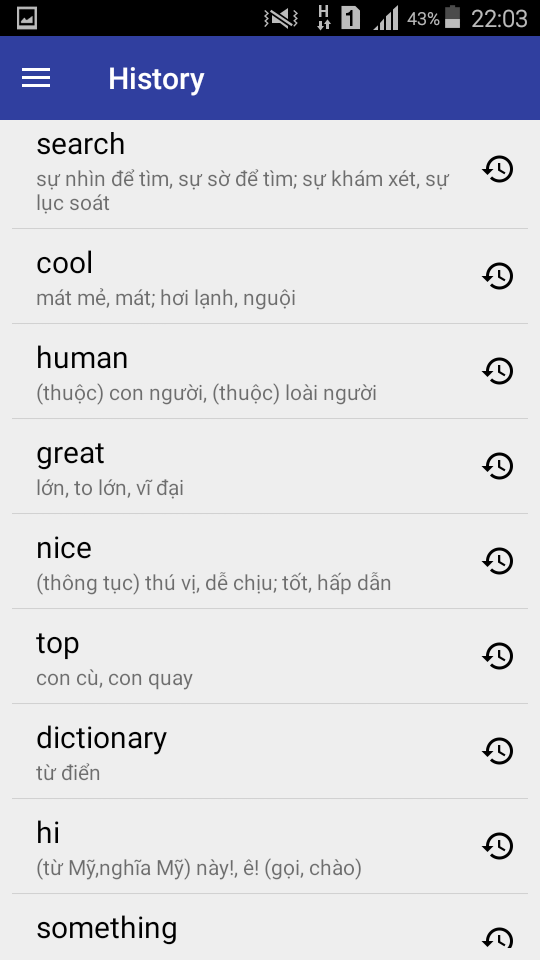
### Kết quả tra từ

Hình 4.3 mô tả màn hình kết quả tra từ. Sau khi nhập tư khóa vào ô tìm kiếm và nhấn enter, ứng dụng sẽ cho kết quả như hình dưới(ví dụ ở đây là từ khóa “great”)



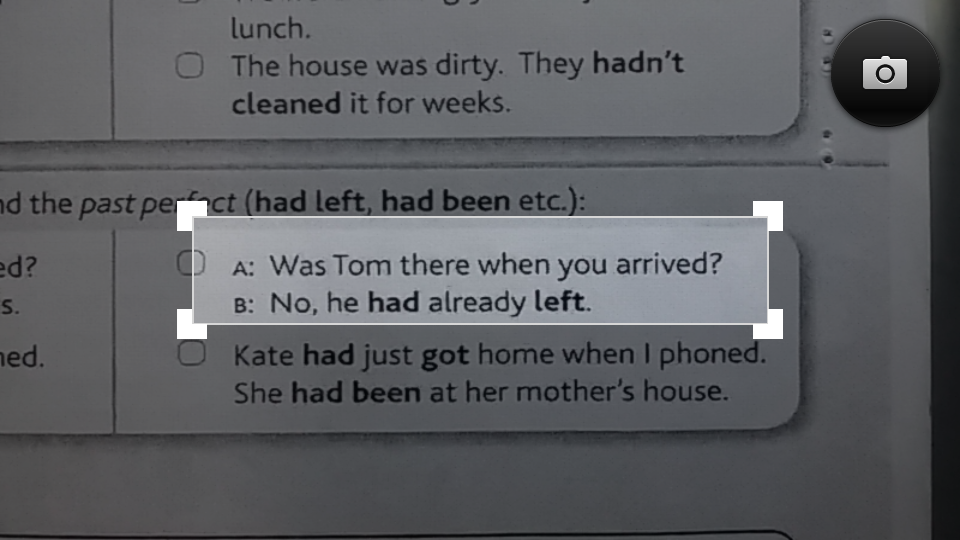
Hình 4.3: Kết quả tra cứu từ

### Xem lịch sử tìm kiếm

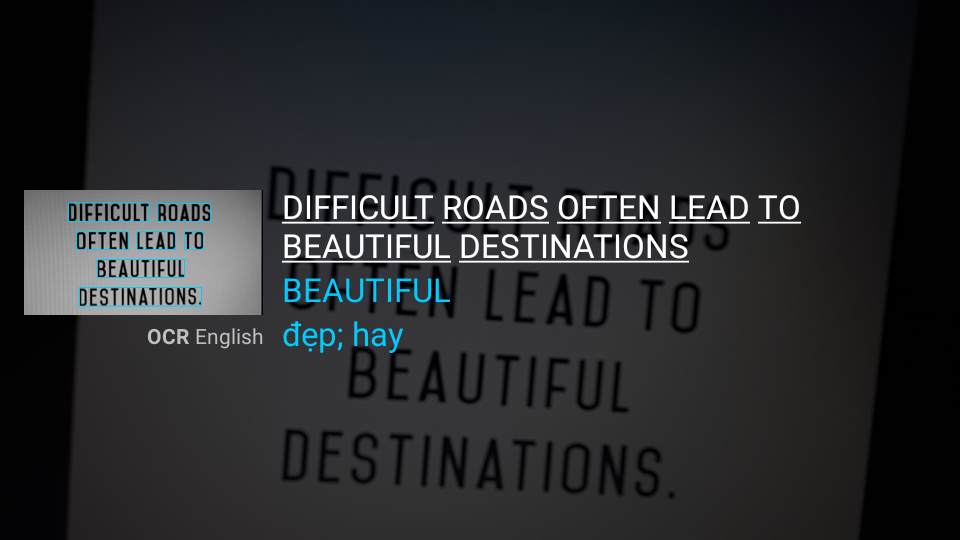


Hình 4.4: Lịch sử tra cứu từ

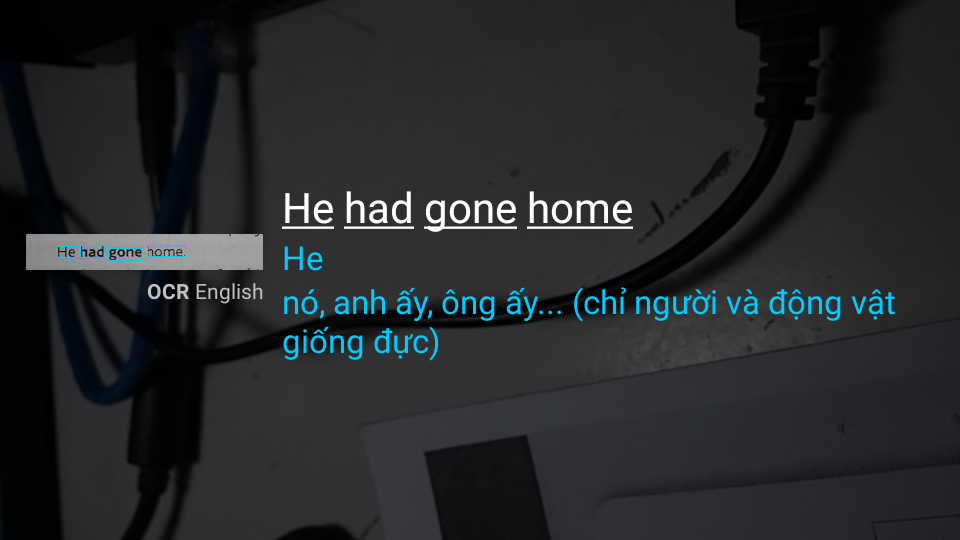
### Tra từ bằng Camera



Hình 4.5: Tra từ bằng Camera 1



Hình 4.6: Tra từ bằng Camera 2



Hình 4.7: Tra từ bằng Camera 3

## Đánh giá

* Ứng dụng tra từ bằng cách scan từ và chụp từ để tra cải thiện đáng kể thời gian tra cứu từ so với các ứng dụng từ điển truyền thống
* Chức năng tra từ bằng cách nhập tay và tra từ bằng camera hoạt động ổng định, đáp ứng kịp thời yêu cầu tra từ
* Xét về nội dung từ điển thì hiện tại ứng dụng bị giới hạn nội dung định nghĩa của từ và phụ thuộc vào dữ liệu của file cơ sở dữ liệu từ điển mở Lingo, chính vì điều này mà một số từ bị thiếu định nghĩa và một số từ định nghĩa chưa được chính xác
* Xét về bộ nhớ của ứng dụng thì hiện tai ứng dụng phải chứ file dữ liệu từ điểnm các file dữ liệu “trained data files” cho tess-two và ứng dụng Realm để lưu trữ cơ sở dữ liệu nên bộ nhớ hiện tại của ứng dụng là lớn.
* Hiện tại thuật toán loại bỏ từ bị nhiễu là kiểm tra từ đó có chứa trong file cơ sở dữ liệu từ điển và trả về kết quả, chính vì điều này nên thuật toán này phụ thuộc vào bộ dữ liệu và phải truy vấn đơn lẻ nhiều lần nên gây chậm ứng dụng.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

### Những vấn đề đã đạt được

* Hoàn thành ứng dụng tra cứu từ điển với chức năng tra cứu bằng cách nhập tay và tra cứu bằng camera, ngoài ra còn cho phép người dùng xem lại lịch sử tra cứu từ
* Giải pháp loại bỏ từ nhiễu để tối ưu hóa kết quả tìm kiếm đã được thực hiện
* Ứng dụng hoàn thiện với giao diện đơn giản dễ sử dụng

### Những vấn đề còn tồn tại

* Hiện tại thuật toán loại bỏ từ nhiễu còn chậm và phụ thuộc vào file dữ liệu từ điển
* Ứng dụng chỉ mới hỗ trợ dịch từ khóa một chiều từ Tiếng Anh sang Tiếng Việt, vẫn chưa hỗ trợ được nhiều ngôn ngữ
* Bộ nhớ phụ thuộc vào file dữ liệu từ điển và file dữ liệu phục vụ cho tess-two và cơ sở dữ liệu Realm nên còn lớn.

## Hướng phát triển

* Nâng cấp chức năng nhận dạng để có thể nhận dạng các vật thể có hình chữ cái, phục vụ cho việc học ngoại ngữ của các em thiếu nhi
* Hỗ trợ nhiều ngôn ngữ hơn nữa như: Tiếng Nhật, Tiếng Trung, Tiếng Hàn …
* Nâng cấp chức năng nhận dạng ký tự để có thể nhận dạng các chữ viết tay
* Thêm chức năng nhận dạng ký tự quang học từ file có sẵn trong máy
* Thêm chức năng dịch đoạn văn để ứng dụng có thể dịch đoạn văn
* Nâng cấp bộ dữ liệu từ điển để có thể đưa ra đúng nghĩa hơn, đa dạng câu ví dụ … khi tra cứu từ

# TÓM TẮT LUẬN VĂN TỐT NGHIỆP

Nội dung đồ án được chia ra 4 chương, mỗi chương có thực hiện một số nội dung khác nhau, cụ thể như sau:

**Chương 1: Tổng quan đề tài**

Chương này trình bày tổng quan về đồ án, bao gồm:

* Lý do chọn đề tài
* Mục tiêu và nhiệm vụ
* Đối tượng nghiên cứu và phạm vi hệ thống
* Phương pháp phát triển

**Chương 2: Cơ sở lý thuyết**

Chương này trình bày các vấn đề về lý thuyết có trong đồ án, bao gồm:

* Android
* OCR
* Tesseract OCR
* Tesstwo
* Tra cứu từ điển
* Loại bỏ từ nhiễu

**Chương 3: Phân tích thiết kế hệ thống**

Chương này trình bày các vấn đề về phân tích thiết kế ứng dụng, bao gồm:

* Mô tả hệ thống:
  + Yêu cầu về chức năng
  + Yêu cầu phi chức năng
  + Yêu cầu hệ thống
* Phân tích thiết kế:
  + Biểu đồ hoạt động
  + Biểu đồ tuần tự
  + Thiết kế cơ sở dữ liệu

**Chương 4: Triển khai và đánh giá kết quả**

Chương này trình bày các vấn đề liên quan đến kết quả thực hiện của chương trình và đánh giá kết quả của quá trình thực hiện, bao gồm:

* Kết quả demo
* Đánh giá

# TÀI LIỆU THAM KHẢO

[1] Ngô Trung Việt (sách dịch), Phân tích và thiết kế Tin học- Hệ thống Quản ‎ lí, Kinh doanh, Nghiệp vụ, Nhà XB Giao thông vận tải 1995.

[2] Đặng Văn Đức, Giáo trình nhập môn UML, NXB Lao động xã hội, 2003.

[3] Nguyễn Văn Ba, Phát triển hướng đối tượng với UML 2.0 và C++, NXB Đại học Quốc gia Hà Nội, 2005.

[4] Nhận dạng ký tự quang học - vi.wikipedia.org

[5] Optical character recognition - en.wikipedia.org

[6] Tesseract OCR - github.com

[7] Tesseract (software) - en.wikipedia.org

[8] tess-two - github.com

[9] Experimental optical character recognition app – github.com

[10] Ray Smith, Google Inc. Training LSTM networks on 100 languages and test results - https://github.com/tesseract-ocr/docs/blob/master/das\_tutorial2016/7Building%20a%20Multi-Lingual%20OCR%20Engine.pdf