

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ**

BÁO CÁO TỔNG KẾT

ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN

**NGHIÊN CỨU CHẾ TẠO THIẾT BỊ
HỖ TRỢ NGƯỜI KHIẾM THỊ ĐỌC SÁCH
BẰNG CÔNG NGHỆ NHẬN DẠNG HÌNH ẢNH
TSV2015-14**

Thuộc nhóm ngành khoa học: Công nghệ thông tin và Truyền thông

Cần Thơ, 12/2015

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ**

BÁO CÁO TỔNG KẾT

ĐỀ TÀI NGHIÊN CỨU KHOA HỌC CỦA SINH VIÊN

**NGHIÊN CỨU CHẾ TẠO THIẾT BỊ
HỖ TRỢ NGƯỜI KHIẾM THỊ ĐỌC SÁCH
BẰNG CÔNG NGHỆ NHẬN DẠNG HÌNH ẢNH
TSV2015-14**

Thuộc nhóm ngành khoa học: Công nghệ thông tin và Truyền thông

Sinh viên thực hiện: Phạm Nguyễn Hải Âu

Nam, Nữ: Nam

Dân tộc: Kinh

Lớp, khoa: DI13Y9A1, Khoa Công nghệ thông tin & Truyền thông

Năm thứ: 3 Số năm đào tạo: 4,5

Ngành học: Truyền thông và mạng máy tính

Người hướng dẫn: TS. Phạm Nguyên Khang

Cần Thơ, 12/2015

DANH SÁCH THÀNH VIÊN THAM GIA

HỌ VÀ TÊN	ĐƠN VỊ
Phạm Nguyễn Hải Âu	MSSV: B1304528 Lớp: Truyền thông và mạng máy tính (DI13Y9A1) Khóa: 39
Đào Minh Tân	MSSV: B1204069 Lớp: Kỹ thuật phần mềm (DI1296A2) Khóa: 38
Huỳnh Nhật Minh	MSSV: 1111311 Lớp: Kỹ thuật phần mềm (DI1196A1) Khóa: 37

ĐƠN VỊ PHỐI HỢP CHÍNH

- ✓ **Khoa Công nghệ thông tin & Truyền thông – Đại Học Cần Thơ**
- ✓ **Phòng Quản lý Khoa học – Đại Học Cần Thơ**

MỤC LỤC

	Trang
DANH SÁCH THÀNH VIÊN THAM GIA	3
ĐƠN VỊ PHỐI HỢP CHÍNH	3
MỤC LỤC	4
DANH MỤC HÌNH	6
DANH MỤC BẢNG	7
DANH MỤC TỪ VIẾT TẮT	8
THÔNG TIN KẾT QUẢ NGHIÊN CỨU CỦA ĐỀ TÀI	9
THÔNG TIN VỀ SINH VIÊN	12
PHẦN I: MỞ ĐẦU.....	14
1. TỔNG QUAN TÌNH HÌNH NGHIÊN CỨU TRONG VÀ NGOÀI NƯỚC	14
2. TÍNH CẤP THIẾT CỦA ĐỀ TÀI.....	15
3. MỤC TIÊU CỦA ĐỀ TÀI	16
4. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU	16
5. PHƯƠNG PHÁP NGHIÊN CỨU	16
PHẦN II: NỘI DUNG & KẾT QUẢ NGHIÊN CỨU	17
CHƯƠNG 1: TỔNG QUAN VỀ XỬ LÝ HÌNH ẢNH VÀ NHẬN DẠNG KÝ TỰ QUANG HỌC TRÊN SÁCH IN TIẾNG VIỆT	17
1.1. Giới thiệu hệ thống xử lý ảnh	17
1.2. Một số vấn đề cơ bản trong xử lý ảnh:	18
1.3. Tách sườn ảnh bằng thuật toán Canny.....	19
1.4. Biến đổi Hough (Hough Line Transform)	20
1.5. Ảnh văn bản sách in Tiếng Việt.....	22
1.6. Nhận dạng ký tự quang học trên sách in Tiếng Việt	22

CHƯƠNG 2: TỔNG QUAN VỀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN	23
2.1. Xử lý ngôn ngữ tự nhiên	23
2.2. Các bước xử lý ngôn ngữ:.....	23
2.3. Một số bài toán và ứng dụng trong xử lý ngôn ngữ tự nhiên	23
CHƯƠNG 3: XÂY DỰNG THIẾT BỊ VÀ CÀI ĐẶT CHƯƠNG TRÌNH.....	25
3.1. Mô tả giải pháp	25
3.2. Xây dựng thiết bị và chương trình	26
CHƯƠNG 4: ĐÁNH GIÁ KẾT QUẢ	37
PHẦN III: KẾT LUẬN VÀ KIẾN NGHỊ.....	41
1. KẾT QUẢ ĐÓNG GÓP	41
2. TRIỂN KHAI ỨNG DỤNG	41
3. HƯỚNG PHÁT TRIỂN	41
TÀI LIỆU THAM KHẢO	43
PHỤ LỤC.....	44
1. OpenCV	44
2. Tesseract	48
3. Praat:	49
4. Arduino:	51
5. Boost:	52
6. fswebcam:	52

DANH MỤC HÌNH

	Trang
Hình 1: Hệ thống xử lý ảnh	17
Hình 2: Các loại đường biên.....	19
Hình 3: Đường thẳng trong hệ tọa độ Descartes và hệ tọa độ cực	21
Hình 4: Biểu diễn Hough của đường thẳng trong hệ tọa độ cực	21
Hình 5: Giao điểm các đường biểu diễn Hough của đường thẳng	22
Hình 6: Mô hình thu nhận ảnh.....	22
Hình 7: Mô hình hệ thống	25
Hình 8: Thiết bị thực tế.....	29
Hình 9: Giao diện chính của chương trình	30
Hình 10: Lưu đồ giải thuật chụp và xử lý ảnh trước khi nhận dạng	31
Hình 11: Ghi âm nhiều từ cùng lúc	33
Hình 12: Đánh dấu âm khoảng âm thanh và khoảng im lặng	35
Hình 13: Mô hình xây dựng TTS	36
Hình 14: Kết quả module tự động xoay ảnh.....	37
Hình 15: Module tự động xác định vị trí văn bản	38
Hình 16: Kết quả tách ảnh thành hai phần	39
Hình 17: So sánh văn bản nhận dạng được từ ảnh	39
Hình 18: Kết quả tách âm thanh với Praat.....	40
Hình 19: Kết quả tách câu thành từng file âm thanh tương ứng với từng từ	40

DANH MỤC BẢNG

	Trang
<i>Bảng 1: Các thành phần trong mô hình hệ thống</i>	<i>26</i>
<i>Bảng 2: Chức năng của thiết bị.....</i>	<i>27</i>

DANH MỤC TỪ VIẾT TẮT

STT	Thuật ngữ	Diễn giải
1.	OpenCV	Open Source Computer Vision
2.	BSD	Berkeley Software Distribution
3.	3D	Three-Dimensional
4.	GPU	Graphics Processing Unit
5.	API	Application program interface
6.	RGB	Red green blue
7.	GNU GPL	GNU General Public License
8.	TTS	Text to speech
9.	ROI	Region of Image

THÔNG TIN KẾT QUẢ NGHIÊN CỨU CỦA ĐỀ TÀI

1. Thông tin chung:

- Tên đề tài: **Nghiên cứu chế tạo thiết bị hỗ trợ người khiếm thị đọc sách bằng công nghệ nhận dạng hình ảnh**
- Sinh viên thực hiện: **Phạm Nguyễn Hải Âu**
- Lớp: Truyền thông và Mạng máy tính
- Khoa: Công nghệ Thông tin và Truyền thông
- Năm thứ: **3** - Số năm đào tạo: **4.5**
- Người hướng dẫn: **TS. Phạm Nguyên Khang**

2. Mục tiêu đề tài:

Nghiên cứu chế tạo thiết bị hỗ trợ người khiếm thị đọc sách bằng công nghệ nhận dạng hình ảnh với chức năng chính là hỗ trợ đọc sách in Tiếng Việt. Thiết bị sẽ giúp người khiếm thị đọc được sách in bằng cách nhận dạng hình ảnh trực tiếp thông qua camera và đọc đoạn văn bản đó. Người khiếm thị sẽ nghe đoạn văn bản qua tai nghe (headphone) trên thiết bị. Đặc biệt là thiết bị phải có giá thành thấp. Thiết bị dễ sử dụng đối với người khiếm thị.

3. Tính mới và sáng tạo:

Đây là một loại thiết bị mới hỗ trợ người khiếm thị đọc sách in Tiếng Việt. Ứng dụng công nghệ nhận dạng hình ảnh bằng Tesseract vào việc nhận dạng chữ in Tiếng Việt. Đề xuất các phương pháp xử lý góc nghiêng để nhận dạng hình ảnh chính xác, tách ảnh để tăng tốc độ xử lý. Đề xuất một phương pháp đơn giản để xây dựng bộ TTS Tiếng Việt. Sử dụng lập trình thiết bị nhúng giúp cho người khiếm thị dễ dàng sử dụng hơn.

4. Kết quả nghiên cứu:

Nhóm nghiên cứu đã tìm hiểu công nghệ nhận dạng hình ảnh và xử lý hình ảnh (dựa trên thư viện mã nguồn mở OpenCV), công nghệ nhận dạng chữ in (dựa trên thư viện mã nguồn mở Tesseract). Đồng thời tìm hiểu và cải tiến công nghệ chuyển đổi văn bản sang giọng nói (dựa trên thư viện mã nguồn mở TTS). Bên cạnh đó, tìm hiểu

quy trình vận hành hệ thống trên nền tảng Linux và quy trình vận hành hệ thống trên thiết bị nhúng Raspberry Pi. Nghiên cứu quy trình thiết kế, bảo trì và đảm bảo chất lượng phần mềm, chất lượng thiết bị. Thêm chức năng đọc tiếng Anh. Kết hợp xử lý đọc Tiếng Việt và Tiếng Anh trong cùng văn bản.

5. Đóng góp về mặt kinh tế - xã hội, giáo dục và đào tạo, an ninh, quốc phòng và khả năng áp dụng của đề tài:

5.1. Về phát triển kinh tế - xã hội

Thiết bị giúp người khiếm thị tiết kiệm công sức và thời gian đọc chữ nổi. Khi người khiếm thị tiếp cận với sách dễ dàng hơn, người khiếm thị sẽ có thể học hỏi được những kiến thức quý báu từ sách, nhờ đó người khiếm thị có thể tham gia vào các hoạt động phát triển kinh tế. Đặc biệt, với chi phí sản xuất rẻ (3.000.000 đồng), người khiếm thị có hoàn cảnh khó khăn dễ dàng tiếp cận thiết bị.

5.2. Về khoa học và đào tạo

Chế tạo được thiết bị hỗ trợ người khiếm thị đọc sách. Thiết bị giúp người khiếm thị tiếp cận sách dễ dàng, từ đó người khiếm thị dễ dàng học tập, tìm hiểu tri thức qua sách báo. Áp dụng kỹ thuật xử lý ảnh và nhận dạng ký tự quang học (OCR) để chuyển hình ảnh thành văn bản. Đề xuất phương pháp xử lý ảnh bị nghiêng, tách ảnh lớn thành ảnh nhỏ để tăng tốc độ xử lý. Nghiên cứu xử lý âm thanh để tách câu dài thành từng tiếng riêng lẻ. Xây dựng một công cụ đọc sách hoàn chỉnh bằng phần mềm mã nguồn mở. Thiết bị tạo tiền đề, làm cơ sở để chế tạo các thiết bị và phần mềm khác hỗ trợ người khiếm thị trong đời sống.

5.3. Về triển khai ứng dụng

Thiết bị có thể triển khai thí nghiệm tại các trung tâm người khuyết tật. Công bố chương trình dưới dạng mã nguồn mở trên Internet.

6. Công bố khoa học của sinh viên từ kết quả nghiên cứu của đề tài

Cần Thơ, ngày ... tháng ... năm 20...

Sinh viên chịu trách nhiệm thực hiện đề tài

Phạm Nguyễn Hải Âu

Nhận xét của người hướng dẫn về những đóng góp khoa học của sinh viên thực hiện đề tài (*phần này do người hướng dẫn ghi*):

Đóng góp khoa học của đề tài:

Áp dụng kỹ thuật xử lý ảnh và nhận dạng ký tự quang học (OCR) để chuyển hình ảnh thành văn bản. Đề xuất phương pháp xử lý ảnh bị nghiêng, tách ảnh lớn thành ảnh nhỏ để tăng tốc độ xử lý.

Nghiên cứu xử lý âm thanh để tách câu dài thành từng tiếng riêng lẻ.

Xây dựng một công cụ đọc sách hoàn chỉnh bằng phần mềm mã nguồn mở.

Cần Thơ, ngày ... tháng ... năm 20...

Xác nhận của Trường Đại học Cần Thơ
(ký tên và đóng dấu)

Giáo viên hướng dẫn
(ký, họ và tên)

Phạm Nguyên Khang

THÔNG TIN VỀ SINH VIÊN

CHỊU TRÁCH NHIỆM CHÍNH THỰC HIỆN ĐỀ TÀI

I. SƠ LƯỢC VỀ SINH VIÊN:

Họ và tên: **Phạm Nguyễn Hải Âu**

Sinh ngày: **04** tháng **03** năm **1995**.

Nơi sinh: **Hiệp Hưng, Phụng Hiệp, Cần Thơ**

Lớp: **Truyền thông và mạng máy tính 1**

Khóa: **39**

Khoa: **Công nghệ Thông tin và Truyền thông**

Địa chỉ liên hệ: **31A/9, Nguyễn Văn Trỗi (Mộ Thân), Phường Xuân Khánh, Quận Ninh Kiều, TP Cần Thơ.**

Điện thoại: **0989 108 862**

Email: **aub1304528@student.ctu.edu.vn**

II. QUÁ TRÌNH HỌC TẬP (kê khai thành tích của sinh viên từ năm thứ 1 đến năm đang học):

❖ Năm thứ 1:

- Ngành học: Truyền thông và mạng máy tính
- Khoa: Công nghệ thông tin và Truyền thông
- Kết quả xếp loại học tập: Xuất sắc (HK1: 3,61; HK2: 3,75)
- Sơ lược thành tích:
 - + Đạt giải Ba Sáng tạo Thanh Thiếu niên – Nhi đồng cấp Quốc Gia
 - + Đạt giải Khuyến khích Nhân tài Đất Việt
 - + Giấy khen cấp trường “Đạt giải quốc gia”
 - + Giấy chứng nhận của hoàn thành tốt “Tiếp sức mùa thi 2014”
 - + Giấy khen cấp trường “Đạt thành tích Xuất sắc trong năm học 2013-2014”.
 - + Giấy khen “Đã có nhiều thành tích đóng góp trong công tác Đoàn và phong trào thanh niên năm học 2013-2014”.
 - + Giấy khen về công tác Đoàn và phong trào thanh niên 2013
 - + Giấy chứng nhận hoàn thành tốt “Mùa hè xanh 2014”
 - + Giấy chứng nhận của hoàn thành tốt “Tiếp sức mùa thi 2014”

+ Chứng nhận tham gia “Tiếp sức đến trường 2014”

+ Đạt danh hiệu Đoàn viên ưu tú.

❖ **Năm thứ 2:**

- Ngành học: Truyền thông và mạng máy tính

- Khoa: Công nghệ thông tin và Truyền thông

- Kết quả xếp loại học tập: Xuất sắc (HK1: 3,75; HK2: 3,75)

- Sơ lược thành tích:

+ Giấy khen cấp trường “Đạt thành tích Xuất sắc trong năm học 2014-2015”

+ Giấy khen “Đã có thành tích tốt trong việc ứng dụng CNTT tình nguyện hỗ trợ hoạt động Đoàn - Hội”

+ 02 Giấy khen về công tác Đoàn và phong trào thanh niên 2014

+ Giấy chứng nhận của hoàn thành tốt “Tiếp sức mùa thi 2015”

Cần Thơ, ngày ... tháng ... năm 20...

Xác nhận của Trường Đại học Cần Thơ
(ký tên và đóng dấu)

Sinh viên chịu trách nhiệm chính
thực hiện đề tài
(ký, họ và tên)

Phạm Nguyễn Hải Âu

PHẦN I: MỞ ĐẦU

1. Tổng quan tình hình nghiên cứu trong và ngoài nước

❖ Ngoài nước

Trên thế giới, đã có rất nhiều thiết bị hỗ trợ người khuyết tật. Đặc biệt là người khiếm thị. Những thiết bị hỗ trợ người khiếm thị đã được quan tâm phát triển rất nhiều:

- Roy Shilkrot, Jochen Huber, Meng Ee Wong, Pattie Maes, Suranga Nanayakkara, *FingerReader: A Wearable Device to Explore Printed Text on the Go*.
- Prof. Amnon Shashua, Ziv Aviram, Dr. Yonatan Wexler, Erez Naaman, Ram Ben Yehuda, *OrCam* - thiết bị di động với camera thông minh hỗ trợ người thị giác kém, <http://www.orcam.com>

Những thiết bị này hỗ trợ người bị các tật về mắt (không mù hoàn toàn) có thể nhận biết rõ cảnh quan xung quanh hay các đồ vật. Đồng thời các thiết bị này cũng cung cấp chức năng đọc văn bản tiếng Anh.

Tuy nhiên những thiết bị này có giá quá cao (đối với OrCam là 3500 USD), hơn nữa các thiết bị trên không hỗ trợ đọc sách tiếng Việt nên những người khiếm thị có hoàn cảnh khó khăn ở Việt Nam khó lòng tiếp cận được.

❖ Trong nước

Ở Việt Nam, hiện nay đã có một số thiết bị, phần mềm hỗ trợ người khiếm thị, một số công trình liên quan đến nhận dạng ảnh, nhận dạng chữ viết:

- TS. Nguyễn Bá Hải, *SPKT Eye*, Trường ĐH Sư phạm Kỹ thuật TPHCM
- Vũ Đình Sáng, Đinh Đình Trọng, cán bộ hướng dẫn PGS.TS. Trần Cao Đệ, *Nhận dạng chữ viết tay trực tuyến rồi rạc tiếng việt*, Luận văn tốt nghiệp, Đại Học Cần Thơ.

SPKT Eye của TS. Nguyễn Bá Hải được đánh giá cao về độ khả thi của nó. Thiết bị này liên tục được cải tiến về giá thành. Nhờ đó người khiếm thị rất ưu chuộng.

Tuy nhiên thiết bị này chỉ tập chung hỗ trợ cho người khiếm thị di chuyển một cách dễ dàng hơn, vẫn chưa thể hỗ trợ để có thể đọc được chữ trên sách, báo.

2. Tính cấp thiết của đề tài

Sách là kho tàng kiến thức đồ sộ nhất của nhân loại. Việc đọc sách mang lại cho chúng ta kiến thức về mọi lĩnh vực trong đời sống vì vậy nhu cầu đọc sách là rất lớn. Tuy nhiên đối với người khiếm thị, việc đó là một sự khó khăn rất lớn.

Theo Bách khoa toàn thư mở Wikipedia: “Thống kê trên thế giới vào năm 2002 cho thấy ước tính hiện nay có khoảng 161 triệu người mù lòa, trong đó có 124 triệu người khiếm thị và 37 triệu người mù, 90% trong số đó đang sống ở các nước đang phát triển (11,6 triệu người ở khu vực Đông Nam Á, 9,3 triệu người ở khu vực Tây Thái Bình Dương, 6,8 triệu người ở Châu Phi), 1,4 triệu trẻ em dưới 15 tuổi bị mù... Ngoài ra trên thế giới còn có hàng triệu người khác bị mù chức năng vì tật khúc xạ (cận thị, viễn thị, loạn thị), 80% người mù trên 50 tuổi. Cứ 5 giây trôi qua lại có thêm một người bị mù và cứ một phút trôi qua lại có một đứa trẻ vĩnh viễn không nhìn thấy ánh sáng. Hàng năm cả thế giới phải tiêu tốn hơn 42 tỷ đôla Mỹ cho việc chữa trị các bệnh về mắt. Khoảng 75% trong số các bệnh gây mù có thể tránh được bằng các phương pháp điều trị hoặc phòng ngừa. Thống kê riêng ở Anh về tuổi khởi phát bệnh mù lòa cho thấy 31% người trả lời nói rằng bắt đầu mắc bệnh khi còn ở độ tuổi lao động (từ 17 đến 59 tuổi), nhưng 56% cho rằng bệnh khởi phát từ khi 60 tuổi trở về sau. Chỉ có 8% bắt đầu bị bệnh khi mới dưới 16 tuổi.”

Tháng 8 năm 2014, tổ chức y tế thế giới WHO công bố ước tính có khoảng 285 triệu người bị khiếm thị trên toàn thế giới. Trong đó có 39 triệu người mù và 246 triệu người suy giảm thị lực. Trong Khoảng 90% người khiếm thị sống ở các quốc gia và vùng lãnh thổ có thu nhập thấp. Việt Nam cũng được xếp vào trong nhóm các nước này.

Các thiết bị hỗ trợ người khiếm thị đọc sách ở nước ngoài có giá thành quá cao (ví dụ: thiết bị Orcam có giá 3500 USD). Người khiếm thị ở các nước có thu nhập thấp như Việt Nam sẽ khó mà tiếp cận được. Hơn nữa, các thiết bị này không hỗ trợ tiếng Việt nên việc tiếp cận chúng đối với người khiếm thị Việt càng khó khăn hơn.

Vì vậy nhóm đề xuất đề tài “Nghiên cứu chế tạo thiết bị hỗ trợ người khiếm thị đọc sách bằng công nghệ nhận dạng hình ảnh” để giải quyết những vấn đề nêu trên.

3. Mục tiêu của đề tài

Nghiên cứu chế tạo thiết bị hỗ trợ người khiếm thị đọc sách bằng công nghệ nhận dạng hình ảnh với giá thành rẻ có các chức năng cơ bản sau:

Thiết bị sẽ giúp người khiếm thị đọc được sách in bằng cách nhận dạng hình ảnh trực tiếp bằng camera và đọc đoạn văn bản đó.

Người khiếm thị sẽ nghe đoạn văn bản qua tai nghe (headphone) trên thiết bị.

4. Đối tượng và phạm vi nghiên cứu

❖ Đối tượng nghiên cứu:

- Công nghệ nhận dạng hình ảnh và xử lý hình ảnh (dựa trên thư viện mã nguồn mở OpenCV).
- Công nghệ nhận dạng chữ in (dựa trên thư viện mã nguồn mở Tesseract).
- Công nghệ chuyển đổi văn bản sang giọng nói (dựa trên thư viện mã nguồn mở Text To Speech – TTS).
- Lập trình nhúng trên tảng Raspberry Pi.

❖ Phạm vi nghiên cứu:

- Nghiên cứu công nghệ xử lý ảnh và nhận dạng hình ảnh để giúp nhận dạng đúng vị trí văn bản chữ in trên giấy;
- Nghiên cứu công nghệ nhận dạng chữ in để chuyển đổi hình ảnh sang văn bản;
- Nghiên cứu công nghệ chuyển đổi văn bản sang giọng nói tiếng Việt.

5. Phương pháp nghiên cứu

- Phân tích nhu cầu cập nhật thông tin, tri thức của người khiếm thị;
- Phân tích yêu cầu, chức năng, và các đặc điểm của hệ thống;
- Nghiên cứu các công nghệ, thư viện nguồn mở có khả năng giúp ích cho hệ thống.
- Thiết kế cơ sở dữ liệu tập trung cho hệ thống;
- Thiết kế mô hình xử lý, mô hình chức năng cho hệ thống;
- Xây dựng tài liệu phân tích thiết kế;
- Lập trình xây dựng các phân hệ, kiểm thử và chỉnh sửa;
- Triển khai và đánh giá.

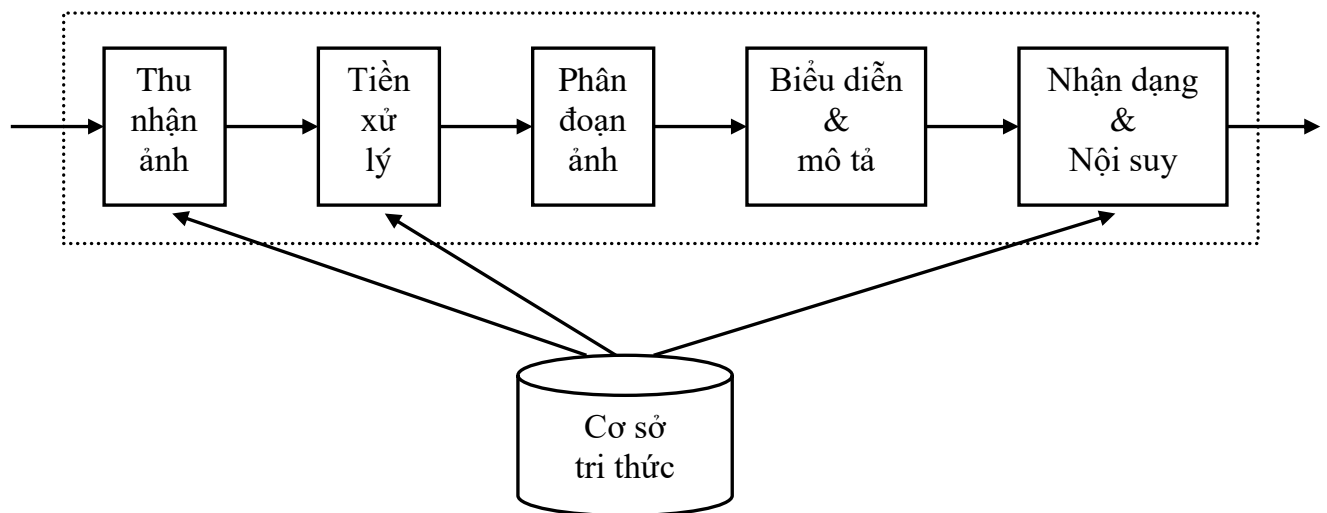
PHẦN II: NỘI DUNG & KẾT QUẢ NGHIÊN CỨU

CHƯƠNG 1: TỔNG QUAN VỀ XỬ LÝ HÌNH ẢNH VÀ NHẬN DẠNG KÝ TỰ QUANG HỌC TRÊN SÁCH IN TIẾNG VIỆT

1.1. Giới thiệu hệ thống xử lý ảnh

Xử lý ảnh là một lĩnh vực mới mẻ trong ngành khoa học máy tính tuy nhiên tốc độ phát triển của nó rất nhanh, thu hút đông đảo nhiều nhà nghiên cứu, sinh viên, học sinh tham gia nghiên cứu.

Các bước cơ bản trong xử lý ảnh:



Hình 1: Hệ thống xử lý ảnh

- Thu nhận hình ảnh (Image Acquisition): Đầu tiên ảnh tự nhiên (hình ảnh sách in) được thu nhận qua thiết bị thu (Camera, máy chụp ảnh,...) chuyển thành ảnh số (ảnh màu hoặc đen trắng) và lưu trên thiết bị lưu trữ (bộ nhớ flash, đĩa...).
- Tiền xử lý hình ảnh (Image Processsing): Ảnh số sau khi thu nhận có thể bị nhiễu, độ tương phản thấp, ảnh bị nghiêng... Khiến các quá trình phân tích, nhận dạng không hiệu quả nên cần đưa vào bộ tiền xử lý để nâng cao chất lượng ảnh (lọc nhiễu, nâng cao độ tương phản, xoay ảnh về vị trí cân bằng...).
- Phân đoạn hay phân vùng ảnh (Image Segmentation): Ảnh sau khi tiền xử lý cần được tách thành các thành phần để biểu diễn, phân tích hoặc nhận dạng.

- Biểu diễn và mô tả ảnh (Image Representation & Description): Đầu ra ảnh sau phân đoạn chứa các điểm ảnh của vùng ảnh (đã phân đoạn) và sự liên kết với các vùng lân cận. Việc biến đổi các thông tin này thành dạng thích hợp là cần thiết cho các bước xử lý tiếp theo. Việc chọn các tính chất để thể hiện ảnh gọi là trích chọn đặc trưng kết hợp với tách đặc tính của ảnh làm cơ sở để phân biệt lớp đối tượng.
- Nhận dạng và nội suy ảnh (Image Recognition & Interpretation): Nhận dạng là quá trình xác định ảnh bằng cách so sánh với mẫu chuẩn đã được học (hoặc lưu trữ) từ trước. Nội suy là phán đoán theo ý nghĩa trên cơ sở nhận dạng.

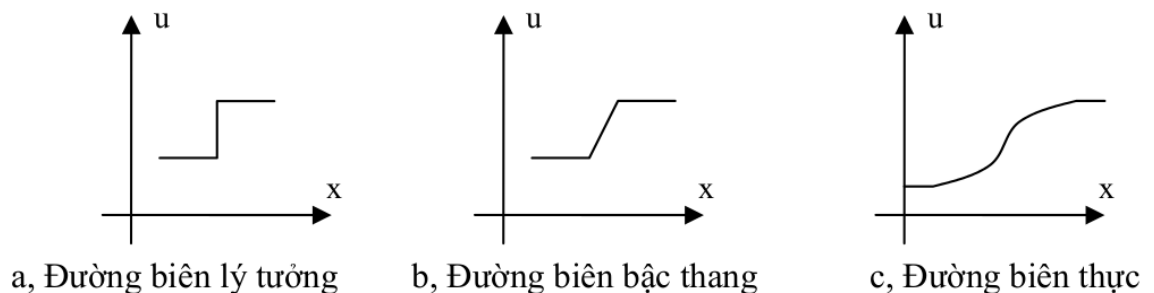
Cơ sở tri thức là cơ sở để phục vụ cho các bước xử lý nhằm làm đơn giản hóa các phương pháp toán học, bắt chước quy trình tiếp nhận và xử lý hình ảnh của con người, để tiện cho việc xử lý.

1.2. Một số vấn đề cơ bản trong xử lý ảnh:

- ❖ Điểm ảnh (Picture Element): ảnh tự nhiên là ảnh liên tục về không gian và độ sáng. Để xử lý bằng máy tính, ảnh cần được số hóa. Số hóa ảnh là sự biến đổi gần đúng một ảnh liên tục thành một tập điểm ảnh phù hợp với ảnh thật về vị trí và độ sáng. Mỗi điểm ảnh (Pixel) gồm một cặp tọa độ (x,y) và mức xám (hoặc màu). Kích thước giữa các điểm ảnh được chọn thích hợp sao cho mắt người cảm nhận được sự liên tục về không gian và màu gần như ảnh thật. Mỗi phần tử trong ma trận ảnh được gọi là một phần tử ảnh.
- ❖ Độ phân giải của ảnh (Image Resolution): là mật độ điểm ảnh được ấn định trên một ảnh số được hiển thị.
- ❖ Mức xám của ảnh: là cường độ sáng của điểm ảnh (x,y) được gán bằng giá trị số tại điểm đó.
 - Các thang giá trị mức xám thông thường là: 16, 32, 128, 256.
 - Ảnh đen trắng: là ảnh có hai màu đen trắng (không có màu khác) với mức xám ở các điểm ảnh có thể khác nhau.
 - Ảnh nhị phân: ảnh chỉ có hai mức đen trắng, mỗi điểm ảnh nhị phân chỉ có thể là mức 0 hoặc 1.

- Ảnh màu: để tạo ảnh màu có thể dùng các hệ màu. Ví dụ trong hệ màu RGB (Red - Blue - Green), dùng 3 byte để mô tả mức màu. Khi đó có $2^{(8*3)}$ màu (xấp xỉ 16 triệu màu).

- ❖ Tách nhiễu và phân ngưỡng:
- ❖ Điểm Biên: Một điểm ảnh được coi là điểm biên nếu có sự thay đổi nhanh hoặc đột ngột về mức xám (hoặc màu). Ví dụ trong ảnh nhị phân, điểm đen gọi là điểm biên nếu lân cận nó có ít nhất một điểm trắng.
- ❖ Đường biên (đường bao: boundary): tập hợp các điểm biên liên tiếp tạo thành một đường biên hay đường bao.
- ❖ Ý nghĩa của đường biên trong xử lý: ý nghĩa đầu tiên: đường biên là một loại đặc trưng cục bộ tiêu biểu trong phân tích, nhận dạng ảnh. Thứ hai, người ta sử dụng biên làm phân cách các vùng xám (màu) cách biệt. Ngược lại, người ta cũng sử dụng các vùng ảnh để tìm đường phân cách.
- ❖ Tầm quan trọng của biên: để thấy rõ tầm quan trọng của biên, xét ví dụ sau: khi người họa sĩ muốn vẽ một danh nhân, họa sĩ chỉ cần vẽ vài đường nét tốc họa mà không cần vẽ một cách đầy đủ.
- ❖ Mô hình biểu diễn đường biên, theo toán học: điểm ảnh có sự biến đổi mức xám $u(x)$ một cách đột ngột theo hình dưới:



Hình 2: Các loại đường biên

1.3. Tách sườn ảnh bằng thuật toán Canny

Phương pháp tách sườn bằng thuật toán Canny được phát triển bởi John F. Canny vào năm 1986. Phương pháp này được sử dụng rộng rãi trong việc tách đường biên.

Phương pháp này sử dụng hai mức ngưỡng cao và thấp. Ban đầu ta dùng mức ngưỡng cao để tìm điểm bắt đầu của biên, sau đó chúng ta xác định hướng

phát triển của biên dựa vào các điểm ảnh liên tiếp có giá trị lớn hơn mức ngưỡng thấp. Ta chỉ loại bỏ các điểm có giá trị nhỏ hơn mức ngưỡng thấp. Các đường biên yếu sẽ được chọn nếu chúng được liên kết với các đường biên khỏe.

Phương pháp Canny bao gồm các bước sau:

Bước 1. Trước hết dùng bộ lọc Gaussian để làm mịn ảnh.

Bước 2. Sau đó tính toán gradient của đường biên của ảnh đã được làm mịn.

Bước 3. Tiếp theo là loại bỏ những điểm không phải là cực đại.

Bước 4. Bước cuối cùng là loại bỏ những giá trị nhỏ hơn mức ngưỡng.

Phương pháp này hơn hẳn các phương pháp khác do ít bị tác động của nhiễu và cho khả năng phát hiện các biên yếu. Nhược điểm của phương pháp này là nếu chọn ngưỡng quá thấp sẽ tạo ra biên không đúng, ngược lại nếu chọn ngưỡng quá cao thì nhiều thông tin quan trọng của biên sẽ bị loại bỏ. Căn cứ vào mức ngưỡng đã xác định trước, ta sẽ quyết định những điểm thuộc biên thực hoặc không thuộc biên. Nếu mức ngưỡng càng thấp, số đường biên được phát hiện càng nhiều (nhưng kèm theo là nhiễu và số các đường biên giả cũng xuất hiện càng nhiều). Ngược lại nếu ta đặt mức ngưỡng càng cao, ta có thể bị mất những đường biên mờ hoặc các đường biên sẽ bị đứt đoạn. Phương pháp

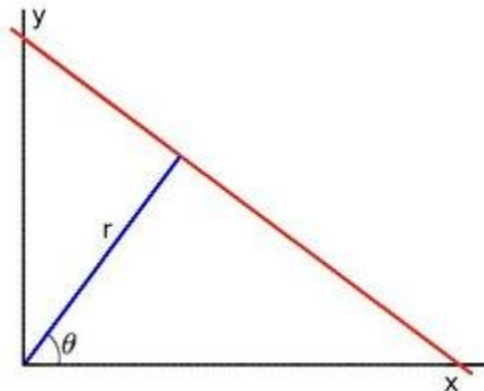
Phương pháp Canny có các ưu điểm sau:

- Cực đại hóa tỷ số tín hiệu trên nhiễu làm cho việc phát hiện các biên thực càng chính xác.
- Đạt được độ chính xác cao của đường biên thực.
- Làm giảm đến mức tối thiểu số các điểm nằm trên đường biên nhằm tạo ra các đường biên mỏng, rõ.

1.4. Biến đổi Hough (Hough Line Transform)

Biến đổi Hough là một phương pháp tìm các đường thẳng trong một hình ảnh. Biến đổi Hough ban đầu là một biến đổi thẳng, đây là một cách tương đối nhanh để tìm một ảnh nhị phân của các đường thẳng. Biến đổi này có thể được tổng quát xa hơn cho các trường hợp hơn là chỉ cho các đường đơn giản.

Như chúng ta biết trong hệ tọa độ Descartes, mỗi đường thẳng được xác định bằng cặp giá trị (x, y) và trong hệ tọa độ cực, nó được xác định bởi cặp giá trị (r, θ) :



Hình 3: Đường thẳng trong hệ tọa độ Descartes và hệ tọa độ cực

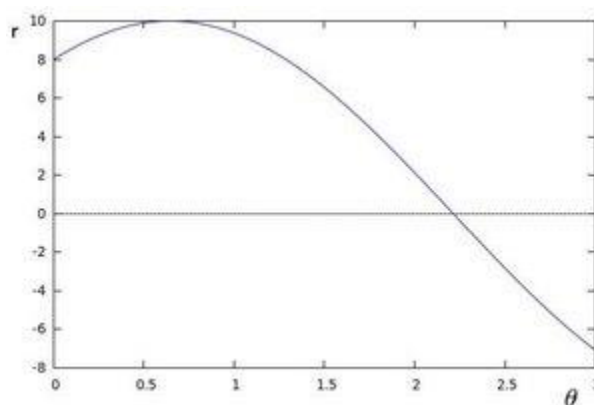
Mối liên hệ giữa (x, y) và (r, θ) là:

$$r = x \cdot \cos \theta + y \cdot \sin \theta$$

Vì vậy, nếu ta có n điểm (x_i, y_i) thuộc một đường thẳng xác định bởi (r, θ) thì ta có phương trình:

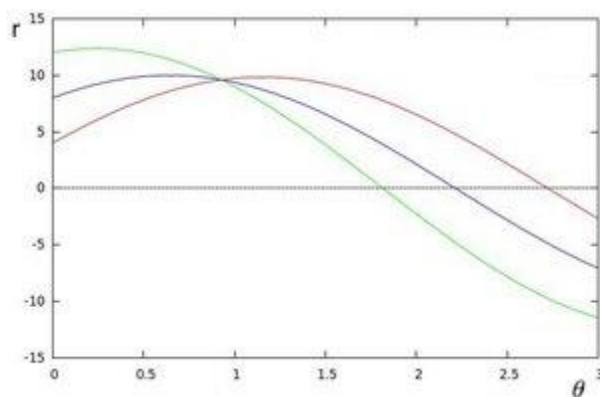
$$r = x_i \cdot \cos \theta + y_i \cdot \sin \theta, \text{ với } i = 0..n$$

Biến đổi Hough sẽ ánh xạ n điểm này thành n đường \sin đi qua (r, θ) trong hệ tọa độ cực:



Hình 4: Biểu diễn Hough của đường thẳng trong hệ tọa độ cực

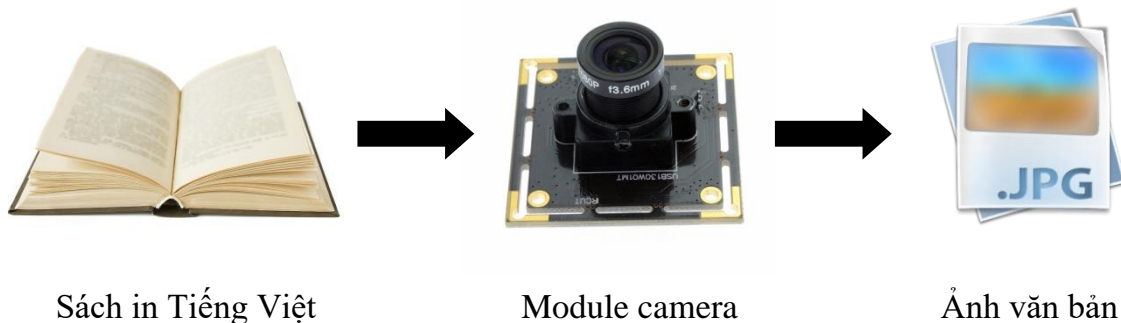
Các đường *sin* này có giao điểm là (r, θ) xác định duy nhất một đường thẳng trong mặt phẳng. Do đó, mỗi đường thẳng đi qua điểm (x, y) xác định một và chỉ một cặp (r, θ) :



Hình 5: Giao điểm các đường biểu diễn Hough của đường thẳng

1.5. Ảnh văn bản sách in Tiếng Việt

Ảnh văn bản sách in Tiếng Việt được đề cập trong đề tài là các file ảnh số thu được bằng module camera lưu trữ dưới định dạng JPG hoặc PNG...



Hình 6: Mô hình thu nhận ảnh

1.6. Nhận dạng ký tự quang học trên sách in Tiếng Việt

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 thị giác máy tính. 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.

CHƯƠNG 2: TỔNG QUAN VỀ XỬ LÝ NGÔN NGỮ TỰ NHIÊN

2.1. Xử lý ngôn ngữ tự nhiên

Xử lý ngôn ngữ tự nhiên (*Natural language processing* - NLP) là một nhánh của trí tuệ nhân tạo tập trung vào các ứng dụng trên ngôn ngữ của con người. Trong trí tuệ nhân tạo thì xử lý ngôn ngữ tự nhiên là một trong những phần khó nhất vì nó liên quan đến việc phải hiểu ý nghĩa ngôn ngữ-công cụ hoàn hảo nhất của tư duy và giao tiếp.

2.2. Các bước xử lý ngôn ngữ:

- Phân tích hình thái: từng từ sẽ được phân tích và các ký tự không phải chữ (như các dấu câu) sẽ được tách ra khỏi các từ. Trong tiếng Việt, dấu cách được dùng để phân tách các tiếng (âm tiết – không phải từ) cũng giống như các ngôn ngữ khác như: tiếng Trung, Hàn, Nhật thì việc phân tách từ là việc không đơn giản.
- Phân tích cú pháp: dãy các từ sẽ được biến đổi thành các cấu trúc thể hiện sự liên kết giữa các từ này. Sẽ có những dãy từ bị loại do vi phạm các luật văn phạm.
- Phân tích ngữ nghĩa - Thêm ngữ nghĩa vào các cấu trúc được tạo ra bởi bộ phân tích cú pháp.
- Tích hợp văn bản - Ngữ nghĩa của một câu riêng biệt có thể phụ thuộc vào những câu đứng trước, đồng thời nó cũng có thể ảnh hưởng đến các câu phía sau.
- Phân tích thực nghĩa - Cấu trúc thể hiện điều được phát ngôn sẽ được thông dịch lại để xác định nó thật sự có nghĩa là gì.

Tuy nhiên, ranh giới giữa 5 bước xử lý này cũng rất mong manh. Chúng có thể được tiến hành từng bước một, hoặc tiến hành cùng lúc - tùy thuộc vào giải thuật và ngữ cảnh cụ thể.

2.3. Một số bài toán và ứng dụng trong xử lý ngôn ngữ tự nhiên

Nhận dạng chữ viết: có hai kiểu nhận dạng, thứ nhất là nhận dạng chữ in, ví dụ nhận dạng chữ trên sách giáo khoa rồi chuyển nó thành dạng văn bản điện tử như dưới định dạng doc của Microsoft Word chẳng hạn. Phức tạp hơn

là nhận dạng chữ viết tay, có khó khăn bởi vì chữ viết tay không có khuôn dạng rõ ràng và thay đổi từ người này sang người khác

Nhận dạng tiếng nói: nhận dạng tiếng nói rồi chuyển chúng thành văn bản tương ứng, tổng hợp tiếng nói là sự trợ giúp tốt cho người khiếm thị, nhưng ngược lại nó là bước cuối cùng trong giao tiếp giữa robot với người.

Tìm kiếm thông tin: đặt câu hỏi và chương trình tự tìm ra nội dung phù hợp nhất. Thông tin ngày càng đầy lên theo cấp số nhân, đặc biệt với sự trợ giúp của Internet việc tiếp cận thông tin trở nên dễ dàng hơn bao giờ hết

CHƯƠNG 3: XÂY DỰNG THIẾT BỊ VÀ CÀI ĐẶT CHƯƠNG TRÌNH

3.1. Mô tả giải pháp

Thiết bị hỗ trợ người khiếm thị đọc sách gồm các thành phần:

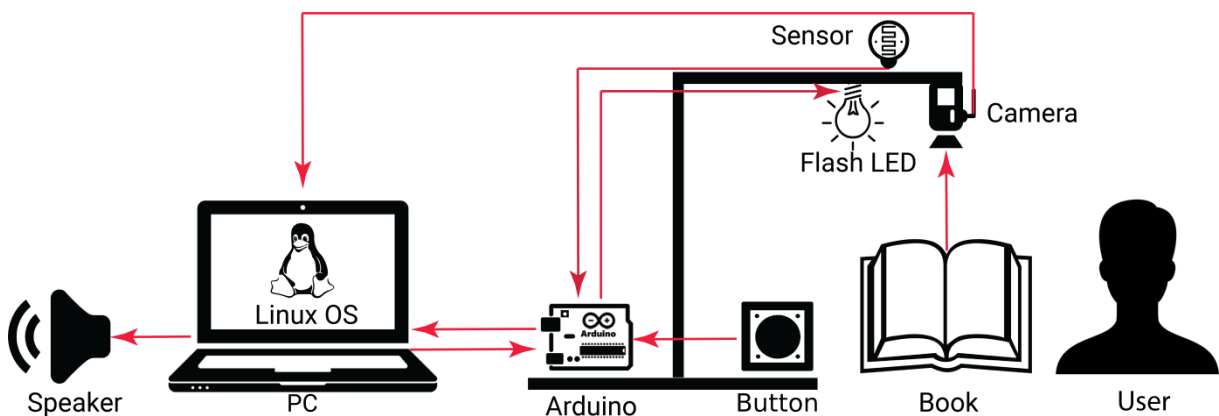
* Thiết bị phần cứng

- Module USB camera (độ phân giải 8 Mpx) dùng để thu nhận hình ảnh
- Các thành phần khác như: LED công suất, MOSFET, nút bấm, quang trở, điện trở, nguồn, cáp kết nối và Kit Arduino UNO R3 được nạp code để đồng bộ với chương trình trên máy tính để nhận tín hiệu từ nút bấm và điều khiển các thiết bị khác hoạt động hỗ trợ quá trình nhận dạng sách hiệu quả hơn.

* Chương trình chạy trên máy tính gồm các module:










- Module chụp và xử lý ảnh trước khi nhận dạng
- Module nhận dạng vị trí văn bản
- Module chuyển đổi hình ảnh sang văn bản
- Module chuyển văn bản sang giọng nói

Trong chương này, chúng tôi trình bày chi tiết về các module kể trên. Phần cài đặt hệ thống cũng sẽ được trình bày ở cuối chương.



Hình 7: Mô hình hệ thống

Bảng 1: Các thành phần trong mô hình hệ thống

TT	Tên đối tượng	Ký hiệu	Mô tả	Ghi chú
1	User		Người khiếm thị (mù hoặc mắc phải các tật về mắt)	
2	Button		Nút điều khiển bắt đầu “đọc sách”	
3	Arduino		Board mạch Arduino dùng để nhận tín hiệu từ nút, cảm biến và điều khiển đèn Flash	
4	Sensor		Đo cường độ ánh sáng từ môi trường	
5	Flash (LED)		Đèn Flash (LED) trợ sáng giúp đọc sách trong môi trường bóng tối hoặc thiếu sáng	
6	Camera		Module camera giao tiếp qua USB dùng để chụp ảnh	
7	Book		Các loại sách in Tiếng Việt	
8	PC		Máy tính chạy hệ điều hành Linux Ubuntu đã cài đặt sẵn chương trình điều khiển	
9	Speaker		Loa dùng để phát âm thanh	

3.2. Xây dựng thiết bị và chương trình

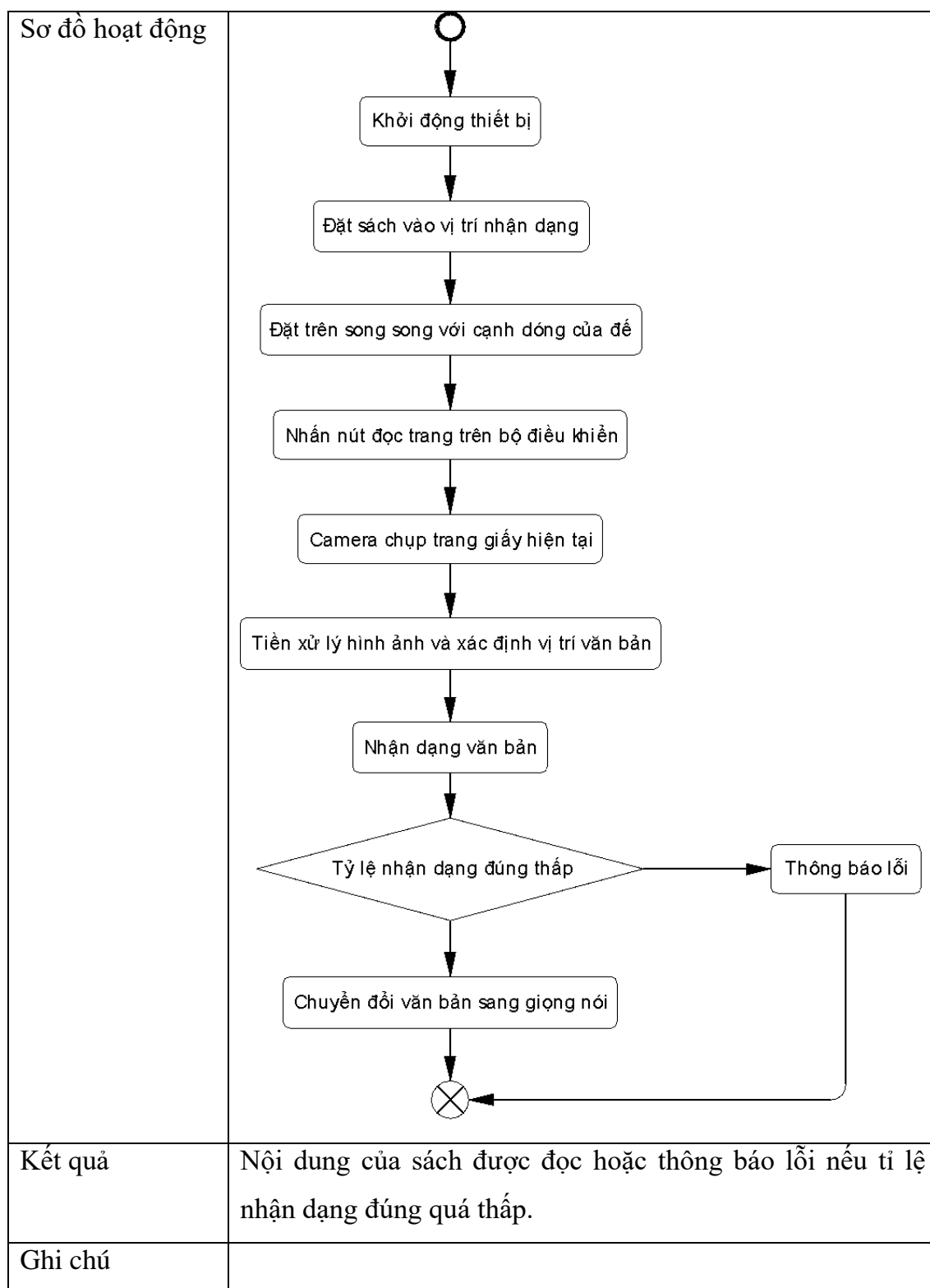
Trong phần này, chúng tôi sẽ giới thiệu về phần mềm và thiết bị “VReader” - một thiết bị và phần mềm hỗ trợ người khiếm thị đọc sách.

Phần mềm được viết bằng ngôn ngữ lập trình C++ trên nền tảng hệ điều hành mã nguồn mở Linux Ubuntu, sử dụng thư viện GTK 3.0 để thiết kế giao diện, OpenCV 3.0.0 để xử lý hình ảnh, Tesseract để nhận dạng chữ in Tiếng Việt và một

sổ thư viện lập trình C++ và chương trình như: Boost, fswebcam,... Riêng module chuyển đổi văn bản sang giọng nói được xây dựng bằng công cụ hỗ trợ xử lý ngôn ngữ tự nhiên Praat và ngôn ngữ lập trình Java.

Bảng 2: Chức năng của thiết bị

Mã yêu cầu	VR01
Tên chức năng	Đọc sách tiếng Việt
Đối tượng sử dụng	Tất cả người sử dụng thiết bị (Người kiểm thị)
Tiền điều kiện	Các linh kiện được kết nối và sẵn sàng hoạt động
Mô tả	Đọc sách in trên các tài liệu in bằng Tiếng Việt
Đầu vào	ảnh chụp của sách in
Đầu ra	Âm thanh đọc từ văn bản đã được nhận dạng từ hình ảnh sách in
Cách xử lý	<p>Bước 1: Mở thiết bị lên</p> <p>Bước 2: Chờ đến khi thiết bị được khởi động xong, người sử dụng đặt sách vào đế (vùng quét của Cammera)</p> <p>Bước 2: Điều chỉnh tài liệu, và có thể sử dụng “chặn sách” để tạo điều kiện tốt nhất cho việc nhận dạng của thiết bị</p> <p>Bước 3: Người dùng nhấn nút “Điều khiển” của hệ thống để bắt đầu tiến trình nhận dạng và đọc văn bản được nhận dạng. Sau khi đọc xong phần mềm sẽ quay lại trạng thái ở Bước 2</p>



Môi trường vận hành:

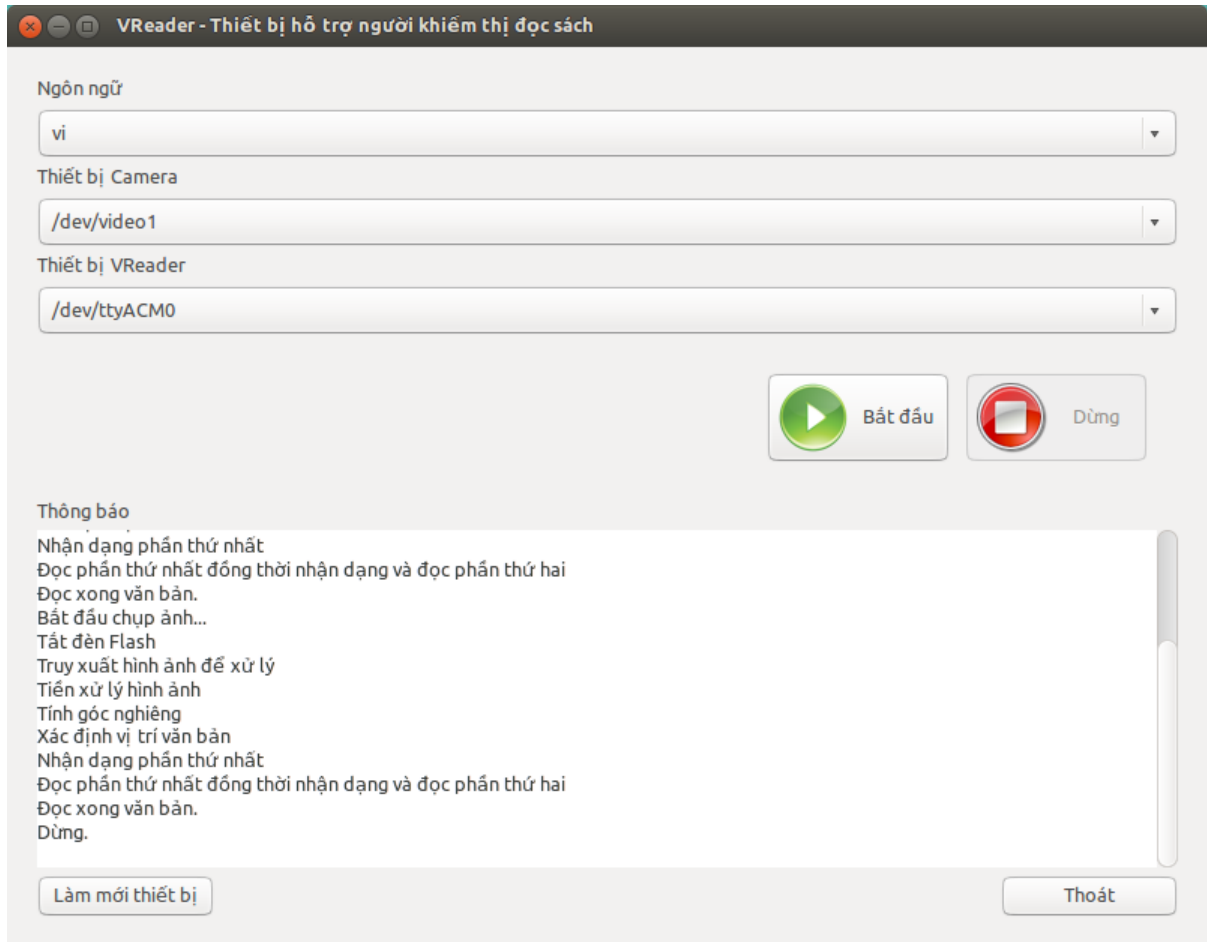
- CPU: 1.8 GHz
- RAM: 4GB
- HDD: 10 GB
- Hệ điều hành: Linux Ubuntu
- Yêu cầu cài đặt các gói hỗ trợ: OpenCV 3.0, GTKmm 3.0, Tesseract, Boost, fswebcam

Các thiết bị phần cứng bao gồm:

- Module USB camera (độ phân giải 8 Mpx)
- Kit Arduino UNO R3 (được nạp code để đồng bộ với chương trình trên máy tính)
- Các thành phần khác như: LED công suất, MOSFET, nút bấm, quang trở, điện trở, nguồn, cáp kết nối...



Hình 8: Thiết bị thực tế



Hình 9: Giao diện chính của chương trình

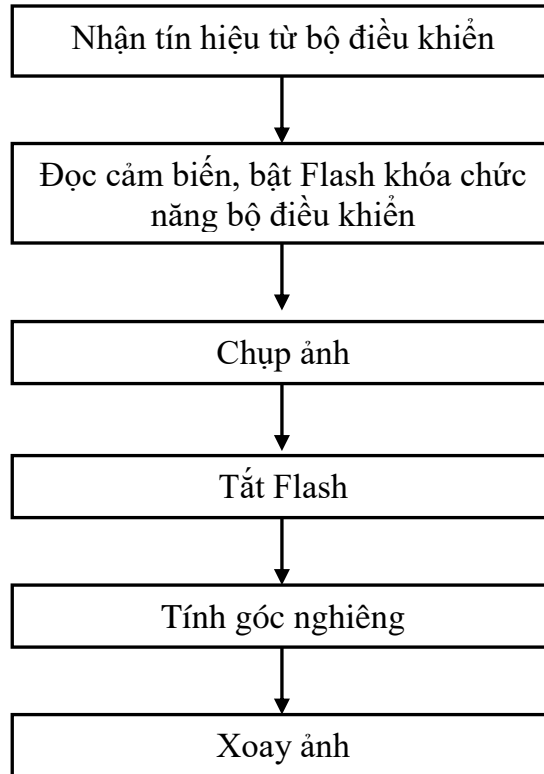
3.2.1. Module chụp và xử lý ảnh trước khi nhận dạng:

Module này có chức năng đồng bộ với thiết bị phần cứng để chụp ảnh và tiền xử hình ảnh.

Khi chưa bắt đầu tiến trình đọc, nếu nút bấm trên bộ điều khiển được nhấn, thiết bị sẽ vô hiệu chức năng của nút bấm khi chưa kết thúc tiến trình, thiết bị đọc giá trị cảm biến (quang trở) để điều khiển đèn Flash (LED công suất) mở độ sáng phù hợp với điều kiện sáng của môi trường. Khi Flash được mở thành công, thiết bị sẽ gửi tín hiệu để báo cho chương trình trên máy tính bắt đầu chụp ảnh. Khi chụp ảnh xong, chương trình trên máy tính sẽ gửi tín hiệu xuống thiết bị để tắt đèn Flash.

Sau đó, chương trình bắt đầu thực hiện tiền xử lý hình ảnh đã chụp bằng một số thuật toán được hỗ trợ trong thư viện OpenCV.

Bước tiếp theo là xác định góc nghiêng của các dòng chữ và xoay ảnh về vị trí cân bằng bằng thuật toán tách sườn ảnh Canny và biến đổi Hough được hỗ trợ bởi thư viện OpenCV.



Hình 10: Lưu đồ giải thuật chụp và xử lý ảnh trước khi nhận dạng

3.2.2. Module xác định vị trí văn bản

Sau khi ảnh được xoay về vị trí cân bằng, phần ảnh này có thể chứa các đối tượng gây nhiễu cho quá trình nhận dạng. Chương trình tiếp tục thực hiện bước xác định trí văn bản và tiến hành tách hình ảnh thành hai phần để xử lý song song. Quá trình này cũng giúp cho thời gian nhận dạng được rút ngắn.

Để xác định vùng ảnh có khả năng chứa văn bản, nhóm nghiên cứu đề xuất giải pháp nhị phân hóa ảnh để làm rõ biên. Tiếp theo là tìm biên và tìm hình chữ nhật nhỏ nhất chứa toàn bộ vùng hình ảnh chứa văn bản. Thao tác tìm biên sử dụng thuật toán *findContours()* được hỗ trợ trong thư viện OpenCV.

3.2.3. Module Nhận dạng chữ in Tiếng Việt

Đến đây, chúng ta đã có hình ảnh có thể nhận dạng tuy nhiên để tăng tốc độ đọc, chúng tôi sử dụng phương pháp xử lý song song. Giả sử chúng ta có ảnh P có thời gian nhận dạng là t và thời gian để đọc văn bản nhận dạng được là t' . Tổng thời gian nhận dạng toàn bộ P và đọc văn bản trên P là $t+t'$.

Bằng cách tìm một vị trí thích hợp để cắt hình ảnh thu được ở bước “Xác định vị trí văn bản” thành hai ảnh P_1, P_2 . Với t_1, t_2 ($t=t_1+t_2$) lần lượt là thời gian nhận dạng P_1, P_2 và t'_1, t'_2 ($t' = t'_1+t'_2$) lần lượt là thời gian đọc văn bản nhận dạng được trên P_1, P_2 (giả sử thời gian đọc P_1, P_2 , nhận dạng P_1, P_2 đều bằng nhau). Sau đó tiến hành nhận dạng P_1 mất t_1 , nhận dạng P_2 và đọc P_1 song song mất $t_2=t'_1$. Cuối cùng là quá trình đọc P_2 mất t'_2 . Vậy tổng thời gian đọc và nhận dạng P là $t_1 + t_2 + t'_2 = t + (1/2)t'$. Nhanh hơn so với đọc toàn bộ P mất $t + t'$. Tuy nhiên thực tế thì quá trình đọc 1/2 văn bản lâu hơn quá trình nhận dạng 1/2 hình ảnh. Do đó chúng tôi không cắt ở vị trí 1/2 hình ảnh mà cắt ở vị trí thích hợp trong khoảng 1/5 - 2/5 chiều cao hình ảnh.

Mục tiêu của ta là tìm một vị trí để cắt ảnh sao cho hạn chế ở mức thấp nhất cắt vào phần ảnh chứa văn bản.

Phương pháp tìm vị trí cắt thích hợp:

Tìm một hình chữ nhật có chiều rộng bằng với chiều rộng của ảnh và chiều cao 1 pixel. Lần lượt cho hình chữ nhật này trượt trên ảnh và đếm số điểm ảnh trắng trong vùng ảnh (ROI) đó. Như vậy, công việc này khá đơn giản bởi đây là bài toán tìm min. Ta dễ dàng tìm được vị trí cắt thích hợp.

3.2.4. Module chuyển văn bản sang giọng nói

Đề tài đề xuất một phương pháp xây dựng bộ TTS đơn giản bằng cách ghi âm liên tục nhiều từ và tách âm thanh tự động.

Module này sử dụng công cụ Praat để tổng hợp tiếng nói và ngôn ngữ lập trình Java để xây dựng bộ chuyển đổi văn bản sang giọng nói (Text-to-speech).

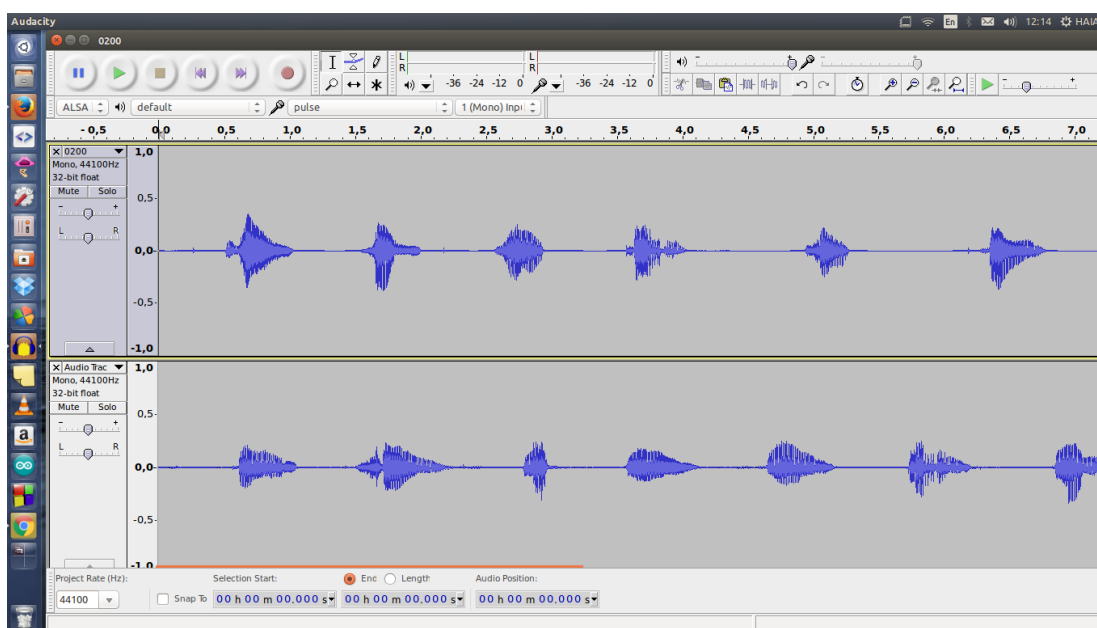
Bước 1: Chuẩn bị danh sách từ phổ biến trong Tiếng Việt

Sử dụng nguồn dữ liệu tất cả bài viết trong Wikipedia Tiếng Việt để tính tần suất xuất hiện của các từ. Sắp xếp các từ theo thứ tự tần suất giảm dần, chọn ra những từ tiếng Việt phổ biến nhất để thu âm.

Chia danh sách từ thành từng bộ 10 từ để dễ dàng ghi âm và xử lý.

Bước 2: Ghi âm từng nhiều từ cùng lúc

Công cụ sử dụng để ghi âm trong đề tài là phần mềm mã nguồn mở Audacity. Việc ghi âm được thực hiện liên tục, tương ứng với từng bộ từ đã chuẩn bị. Trong quá trình ghi âm môi trường có thể có tiếng ồn, để đảm bảo quá trình tách được chính xác, phải tiến hành khử nhiễu của âm thanh bằng tính năng “Noise Removal”. Tính năng này xử lý một cách tự động và hàng loạt. Sau đó chương trình sẽ lưu tự động với mã số.

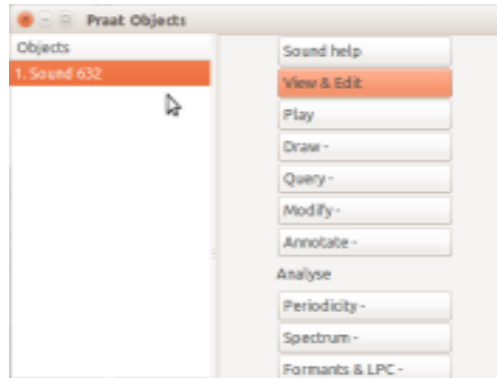


Hình 11: Ghi âm nhiều từ cùng lúc

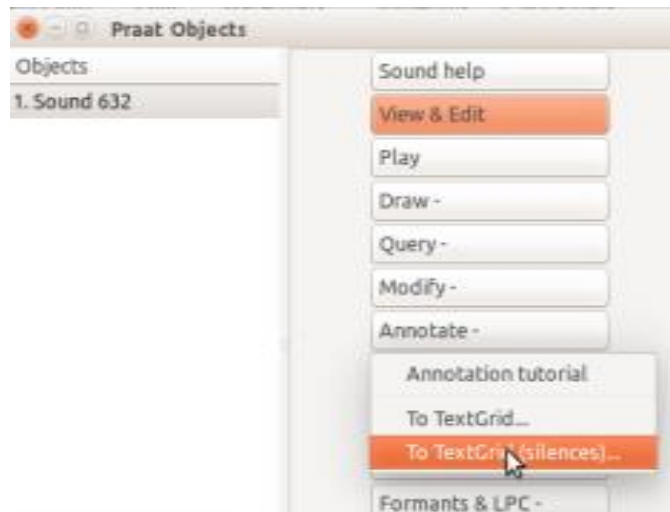
Bước 3: Sử dụng Praat để tách âm thanh của từng từ trong đoạn âm thanh:

Để tách âm thanh của từng từ trong một đoạn âm thanh của một câu hoặc một đoạn văn ta cần có: file âm thanh (của câu hoặc đoạn văn), file văn bản (chứa nội dung của các từ tương ứng với đoạn âm thanh của câu hoặc đoạn văn đó), file script - ở đây là file mã nguồn sẽ thực hiện việc tách âm thanh dựa vào Praat

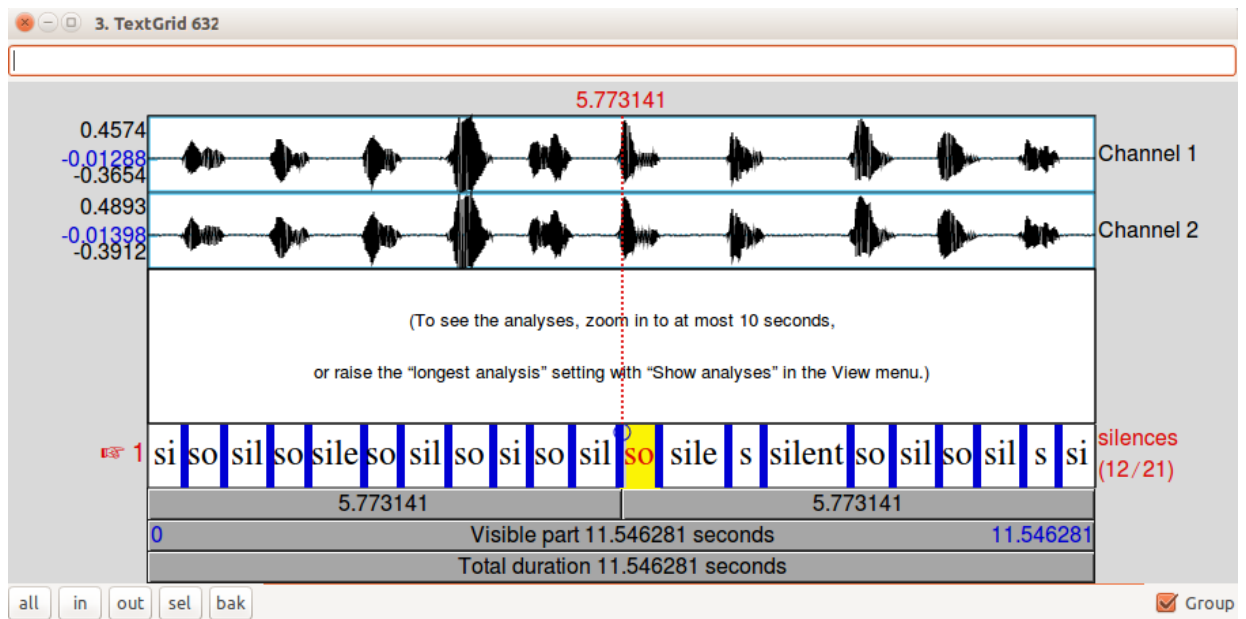
- Mở file âm thanh để xử lý
 - + Đầu tiên, trên giao diện chính của Praat chọn Open > Read from file...
 - + Chọn vào file âm thanh cần xử lý
 - + Kết quả sau khi thêm file âm thanh



- Thực hiện đánh dấu để phân biệt âm thanh của các từ và khoảng im lặng
 - + Chọn file âm thanh từ khung Objects
 - + Sau đó chọn vào Annotate > To TextGrid (silences)...



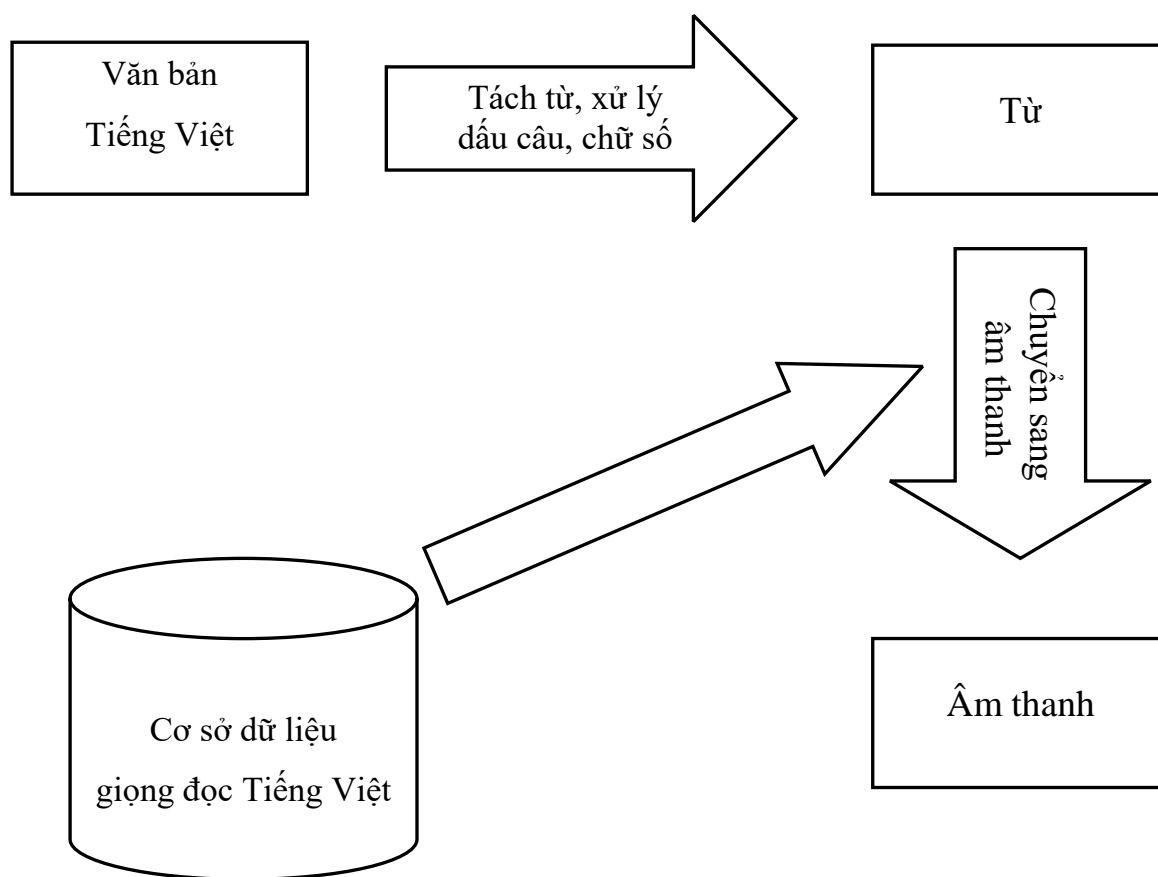
- + Một hộp thoại hiện lên, giữ các thông số mặc định và nhấn OK
- + Một file TextGrid được tạo ra tương ứng với file âm thanh
- + Kết quả của bước thực hiện trên: (các khoảng có âm thanh và im lặng được đánh dấu)



Hình 12: Đánh dấu âm khoảng âm thanh và khoảng im lặng

- Sau khi các vị trí được xác định, ta tiến hành tách đoạn âm thanh thành các file âm thanh của các từ riêng biệt
 - + Thêm file Long Sound: từ giao diện chính của phần mềm Praat ta chọn Open > Open Long Sound File... (hoặc nhấn Ctrl+L)
 - + Chọn đến địa chỉ của File âm thanh của đoạn/câu văn
 - + Mở tiếp file script để thực hiện việc tách file: Vào Praat > Open Praat Script ...
 - + Chọn đến Script để tách âm thanh
 - + Trên cửa sổ hiện lên mã nguồn script chọn Run > Run
 - + Nhập thông tin của file Text chứa nội dung của các từ trong đoạn âm thanh và vị trí lưu đầu ra , nhấn Ok để thực hiện
 - + Kết quả là các file âm thanh riêng biệt sẽ được tạo ra đúng với các âm thanh của từng từ trong đoạn âm thanh đầu vào và mang tên tương ứng với nội dung trong file văn bản đã chuẩn bị

Bước 4: Xây dựng bộ TTS



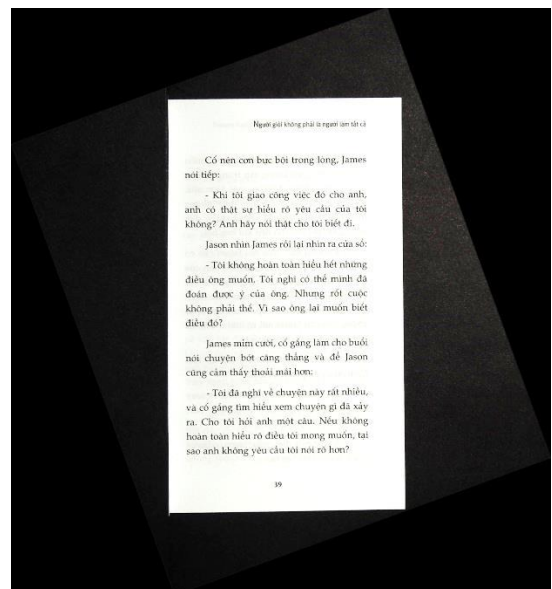
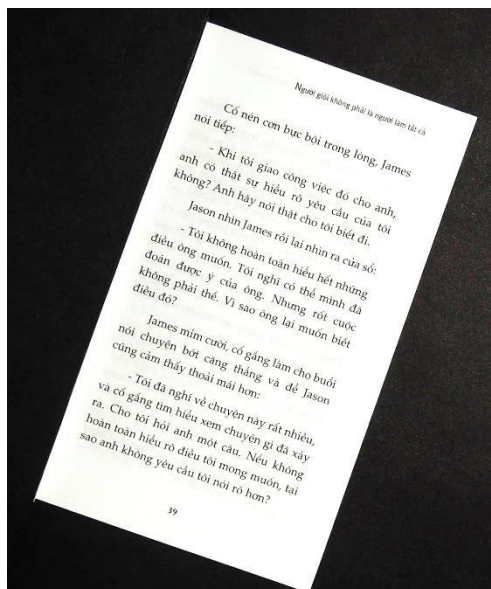
Hình 13: Mô hình xây dựng TTS

CHƯƠNG 4: ĐÁNH GIÁ KẾT QUẢ

Bộ test sử dụng: 410 ảnh với nhiều góc nghiêng khác nhau có bố cục một cột, chữ đen, nền trắng, font chữ Times New Roman. Thời gian thực thi tập trung chủ yếu vào khâu nhận dạng. Thời gian xử lý hình ảnh (xoay, xác định vị trí) không đáng kể (trung bình khoảng 1 giây). Với tùy chọn nhận dạng mặc định của Tesseract: 15 từ / giây, độ chính xác là 93.5%. Với tùy chọn nhận dạng với OSD của Tesseract cho tốc độ khoảng 30 từ / giây thì độ chính xác giảm xuống còn 93.3%. Đối với bộ Test có chứa văn bản có nhiều bố cục phức tạp, chữ màu sáng, nền quá tối, có nhiều hình ảnh, bảng biểu sẽ ảnh hưởng đến kết quả nhận dạng.

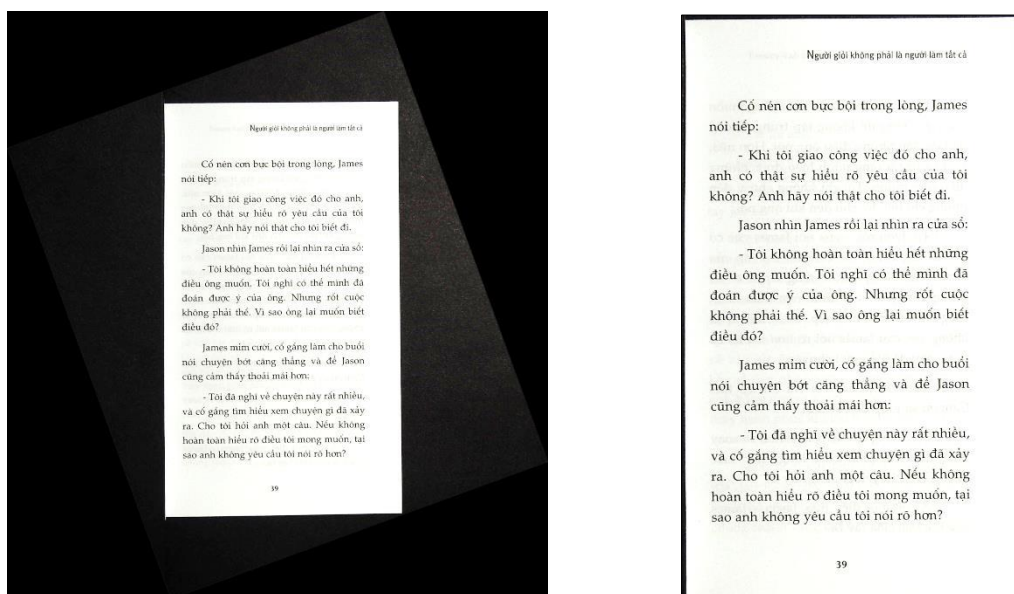
Kết quả xử lý ảnh nghiêng

Chương trình có thể tự động xoay ảnh nghiêng ± 20 độ, sai lệch là ± 1 độ. Kết quả sai lệch này không làm ảnh hưởng đến khả năng nhận dạng của Tesseract.



Hình 14: Kết quả module tự động xoay ảnh

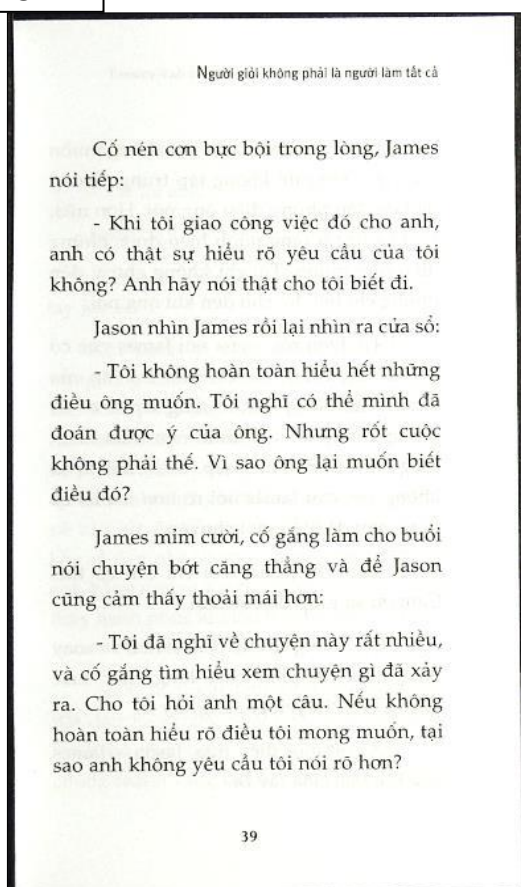
Kết quả module xác định vị trí văn bản:



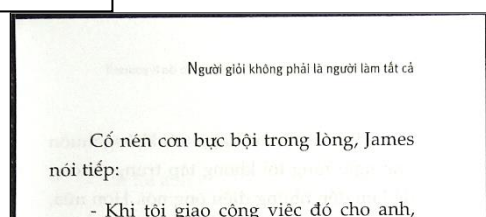
Hình 15: Module tự động xác định vị trí văn bản

Kết quả tách ảnh thành hai phần:

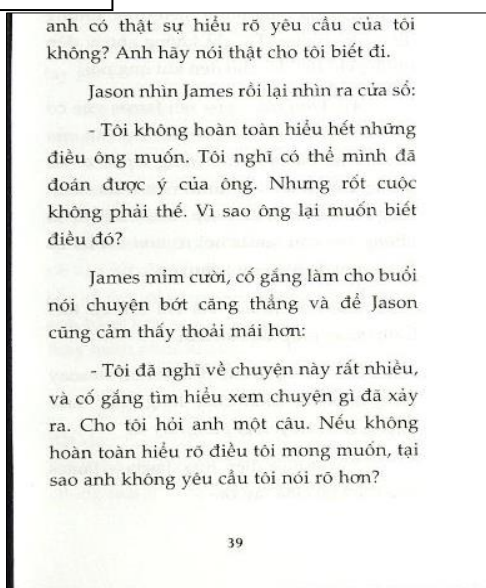
Ảnh gốc



Phần 1

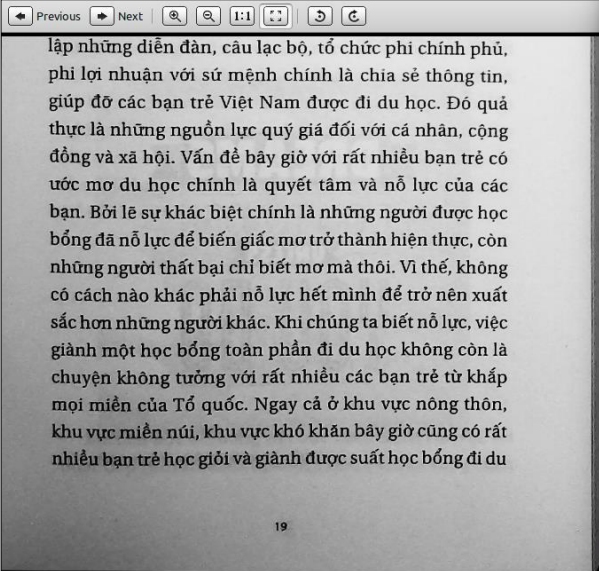
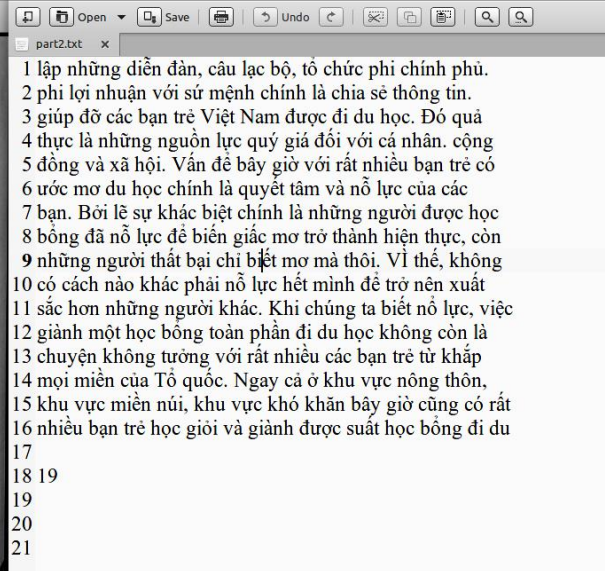


Phần 2



Hình 16: Kết quả tách ảnh thành hai phần

Kết quả module nhận dạng:

Hình ảnh	Văn bản nhận dạng được
	

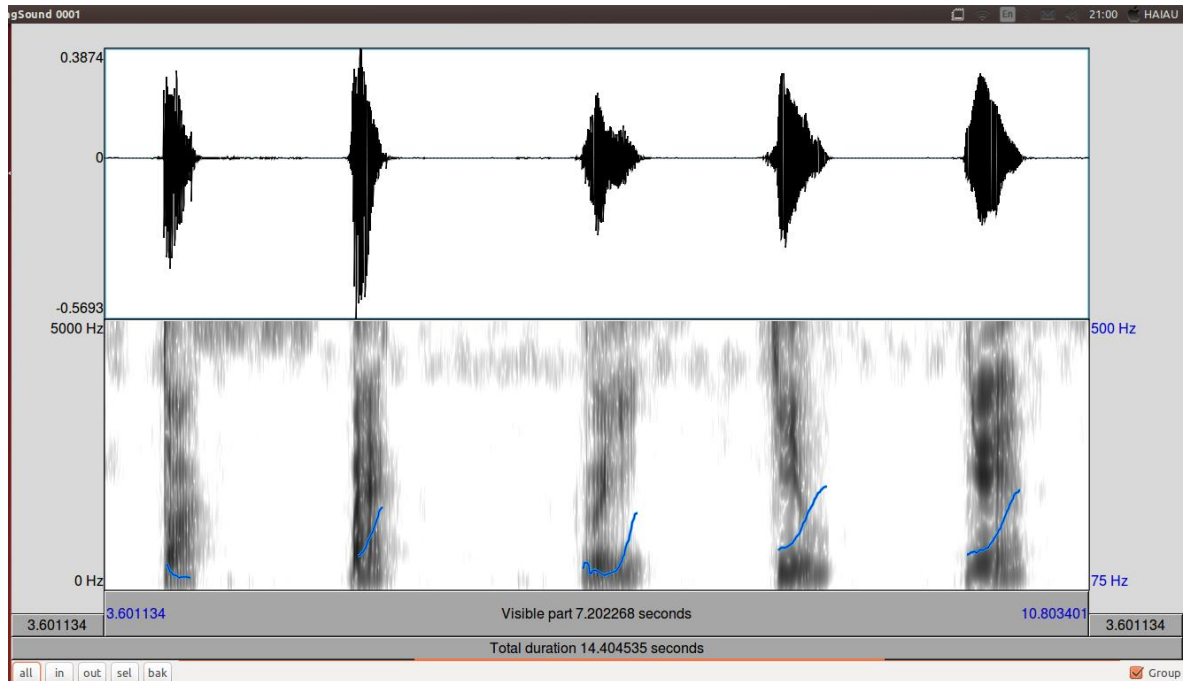
Hình 17: So sánh văn bản nhận dạng được từ ảnh

Kết quả tách câu thành từng file âm thanh tương ứng với từng từ:

Đối với cách thu âm thu âm thủ công từng từ một, người dùng sẽ tốn thời gian cho việc xử lý và lưu trữ thành từng file. Thời gian trung bình để ghi âm và xử lý một file là 30 giây. Vậy nếu ghi âm khoảng 10 nghìn từ, thời gian này sẽ là 5000 phút làm việc liên tục.

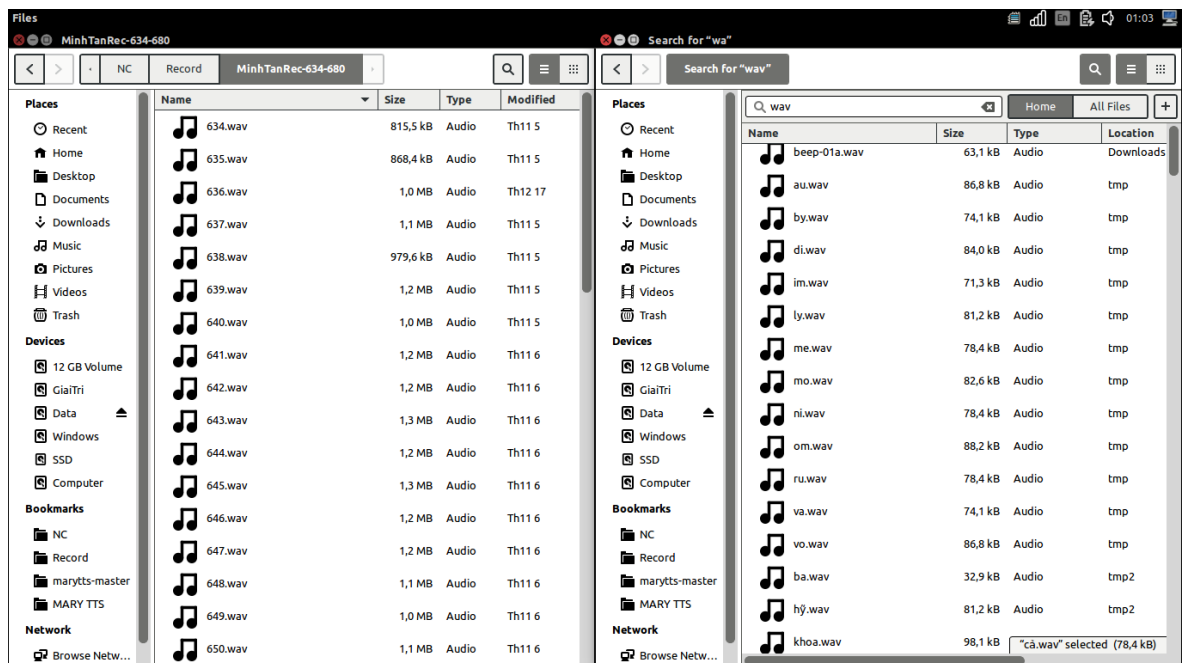
Đối với phương pháp ghi âm đề xuất trong đề tài, thời gian ghi âm mỗi bộ 10 từ là 60 giây. Thời gian ghi âm 10 nghìn từ là 1000 phút. Thời gian để chạy script tách từ mất 30 phút. Tổng thời gian thực hiện với phương pháp này là 1030 phút xử lý liên tục.

Như vậy, phương pháp được thực hiện trong đề tài tiết kiệm thời gian hơn phương pháp ghi âm thủ công xấp xỉ 5 lần.



Hình 18: Kết quả tách âm thanh với Praat

Kích thước của bộ ngữ liệu tiếng nói tổng hợp được là hơn 9.200 file với dung lượng là khoảng 400 MB. Với số lượng dữ liệu này, có thể đáp ứng hầu hết các loại văn bản thông dụng. Đối với những từ không có trong cơ sở dữ liệu, đề tài sử dụng phương pháp đánh vần từ đó để người khiếm thị đoán nghĩa của từ.



Hình 19: Kết quả tách câu thành từng file âm thanh tương ứng với từng từ

PHẦN III: KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết quả đóng góp

1.1. Về khoa học và đào tạo

Thiết bị được thiết bị hỗ trợ người khiếm thị đọc sách. Thiết bị giúp người khiếm thị tiếp cận sách dễ dàng, từ đó người khiếm thị dễ dàng học tập, tìm hiểu tri thức qua sách báo. Thiết bị tạo tiền đề, làm cơ sở để chế tạo các thiết bị và phần mềm khác hỗ trợ người khiếm thị trong đời sống.

1.2. Về phát triển kinh tế

Thiết bị giúp người khiếm thị tiết kiệm công sức và thời gian đọc chữ nổi. Khi người khiếm thị tiếp cận với sách dễ dàng hơn, người khiếm thị sẽ có thể học hỏi được những kiến thức quý báu từ sách, nhờ đó người khiếm thị có thể tham gia vào các hoạt động phát triển kinh tế. Đặc biệt, với chi phí sản xuất rẻ (3.000.000 đồng), người khiếm thị có hoàn cảnh khó khăn dễ dàng tiếp cận thiết bị.

1.3. Về xã hội

Thiết bị giúp người khiếm thị sẽ cảm thấy tự tin hơn khi mình hoàn toàn có thể tiếp cận kho tàng kiến thức của nhân loại. Thiết bị góp phần xoa dịu nỗi đau cho người khiếm thị đồng thời cũng cải thiện đời sống tinh thần cho người khiếm thị. Khi người khiếm thị có thể tiếp cận được tri thức, có thể tham gia các hoạt động phát triển kinh tế thì người khiếm thị không còn là gánh nặng cho gia đình và xã hội nữa.

2. Triển khai ứng dụng

Thiết bị có thể triển khai thí nghiệm tại các trung tâm người khuyết tật.

Công bố chương trình dưới dạng mã nguồn mở trên Internet.

3. Hướng phát triển

Trong tương lai, nhóm sẽ phát triển tiếp thiết bị:

- Cải thiện độ chính xác của thiết bị bằng cách trang bị Camera độ nét cao hơn.
- Cải thiện thuật toán để tăng tốc độ xử lý.
- Cải thiện chất lượng giọng đọc, thêm nhiều giọng đọc hơn.

- Sử dụng Raspberry Pi thay cho việc sử dụng máy tính giúp cho việc di chuyển dễ dàng hơn.
- Hỗ trợ nhiều loại font chữ, bố cục, màu chữ hơn...
- Giao diện đơn giản hơn, có thể cho ứng dụng tự chạy sau khi khởi động.
- Điều khiển bằng khẩu lệnh.

TÀI LIỆU THAM KHẢO

- [1] Michael Margolis, *Arduino Cookbook*, 2011
- [2] John Boxall, *Arduino Workshop*, 2013
- [3] Gary Bradski and Adrian Kaehler, *Learning OpenCV*, 2008
- [4] Nguyễn Trọng Vinh, *Nghiên cứu một số kỹ thuật hiệu chỉnh góc nghiêng của ảnh*, Đại Học Lạc Hồng, 2011.
- [5] Pascal van Lieshou, *PRAAT Short Tutorial*, V. 4.2.1, 2003
- [6] Samarth Brahmbhatt, *Practical OpenCV*
- [7] Hồ Tú Bảo và Lương Chi Mai, *Về xử lý tiếng Việt trong công nghệ thông tin*
- [8] Nguyễn Vĩnh An, *So sánh một số phương pháp phát hiện biên*, 2015
- [9] Will Styler, *Using Praat for Linguistic Research*, Version: 1.6, 2015
- [10] Intel Corporation, *OpenCV 3.0.0 [Online]*, 2015, Available:
<http://docs.opencv.org/3.0.0/index.html>

PHỤ LỤC

1. OpenCV

Thư viện OpenCV

OpenCV (Open Source Computer Vision Library: <http://opencv.org>) là một thư hàm lập trình mã nguồn mở trong lĩnh vực thị giác máy tính (Computer Vision) và máy học (Machine Learning) được phân phối dưới giấy phép BSD được phát triển bởi trung tâm nghiên cứu của Intel ở Nizhny Novgorod (Nga), sau đó được hỗ trợ bởi Willow Garage và hiện nay được phát triển tiếp tục bởi Itseez.

OpenCV được viết bằng ngôn ngữ lập trình C và C++ hỗ trợ đa nền tảng (cross-platform), có thể phát triển một cách chủ động với nhiều ngôn ngữ khác nhau như C, C++, Python, Java,... Các hàm trong OpenCV dễ sử dụng giúp cho các nhà phát triển triển khai các ứng dụng một cách dễ dàng và nhanh chóng. OpenCV được thiết kế hướng đến các ứng dụng tính toán tối ưu và trọng tâm chính là các ứng dụng thời gian thực.

Thư viện này gồm hơn 2500 thuật toán tối ưu trong lĩnh vực thị giác máy tính như nhận dạng khuôn mặt, nhận dạng chuyển động người, kiểm tra sản phẩm trong công nghiệp, chuẩn đoán hình ảnh trong y học, an ninh, giao diện người dùng, nhận dạng hình ảnh, thực tế ảo, robot học, ...

Các thành phần của OpenCV

OpenCV được module hóa với nhiều thư viện chia sẻ (shared library) và thư viện tĩnh (static library), bao gồm:

- + **core** – module chính định nghĩa các cấu trúc dữ liệu cơ bản bao gồm mảng dày đặc và đa chiều (Mat) và các hàm cơ bản được dùng trong tất cả các module khác.
- + **imgproc** – module hỗ trợ xử lý hình ảnh bao gồm các bộ lọc hình ảnh tuyến tính và phi tuyến, các hàm biến đổi đổi hình thái học, chuyển đổi hệ màu, tính toán histogram, ...
- + **video** – module phân tích video bao gồm các hàm tính toán chuyển động, trừ nền (background subtraction) và các thuật toán theo dõi đối tượng...

- + ***calib3d*** – các thuật xử lý hình thái học đa góc nhìn, hiệu chuẩn từ camera đơn hoặc camera kép, phán đoán đối tượng chụp, các thuật toán xử lý nổi tương ứng, xây dựng lại cấu trúc 3D của đối tượng.
- + ***features2d*** – các thuật toán làm nổi bật sự dò tìm, mô tả sự ăn khớp.
- + ***objdetect*** – nhận dạng các đối tượng và thể hiện của các lớp đã định nghĩa sẵn như (ví dụ như khuôn mặt, mắt, con người, xe cộ, ...)
- + ***highgui*** – một interface đơn giản và dễ sử dụng để xây dựng giao diện người dùng đơn giản.
- + ***videoio*** – một interface đơn giản và dễ sử dụng để quay video, mã hóa và giải mã video.
- + ***gpu*** – các thuật toán tăng tốc GPU từ các module OpenCV khác.
- + Các module khác như: imgcodecs, FLANN, Python bindings, ...

Cách cài đặt thư viện OpenCV trên nền tảng Linux

❖ Bước 1: Cài đặt trình biên dịch và các gói cần thiết

Để cài đặt OpenCV, yêu cầu phải cài đặt các gói sau:

- GCC 4.4.x hoặc mới hơn
- CMake 2.8.7 hoặc mới hơn
- Git
- GTK+2.x hoặc mới hơn, bao gồm phần header (libgtk2.0-dev)
- pkg-config
- Python 2.6 hoặc mới hơn và Numpy 1.5 hoặc mới hơn với gói phát triển (python-dev, python-numpy)
- ffmpeg hoặc gói phát triển libav: libavcodec-dev, libavformat-dev, libswscale-dev
- [Tùy chọn] libtbb2 libtbb-dev
- [Tùy chọn] libdc1394 2.x
- [Tùy chọn] libjpeg-dev, libpng-dev, libtiff-dev, libjasper-dev, libdc1394-22-dev

Các gói này có thể được cài đặt bằng các lệnh sau:

- Cài đặt trình biên dịch:

```
$ sudo apt-get install build-essential
```

- Các cài đặt bắt buộc:

```
$ sudo apt-get install cmake git libgtk2.0-dev pkg-  
config libavcodec-dev libavformat-dev libswscale-  
dev
```

- Các cài đặt tùy chọn:

```
$ sudo apt-get install python-dev python-numpy  
libtbb2 libtbb-dev libjpeg-dev libpng-dev libtiff-  
dev libjasper-dev libdc1394-22-dev
```

❖ Bước 2: Tải về mã nguồn OpenCV:

- Cách 1: Tải về tại địa chỉ <http://opencv.org/downloads.html>
- Cách 2: Dùng Git với cú pháp như sau:

```
$ cd ~  
$ git clone https://github.com/Itseez/opencv.git  
$ git clone  
https://github.com/Itseez/opencv_contrib.git
```

Ghi chú: Để tiện thao tác trong các bước sau, hướng dẫn này đặt mã nguồn opencv trong thư mục \$HOME của người dùng.

❖ Bước 3: Biên dịch mã nguồn OpenCV với CMake:

- Chuyển đến thư mục ~/opencv và tạo một thư mục để chứa các file sinh ra trong quá trình biên dịch:

```
$ cd ~/opencv  
$ mkdir build  
$ cd build
```

- Thiết lập CMake:

```
$ cmake -D CMAKE_BUILD_TYPE=Release -D  
CMAKE_INSTALL_PREFIX=/usr/local ..
```

- Biên dịch:

```
$ make -j7
```

- Cài đặt:

```
$ sudo make install
```

Cách sử dụng thư viện OpenCV:

❖ Sử dụng header của OpenCV:

Để sử dụng các thành phần trong OpenCV mã nguồn C++ cần bao gồm tập tin core.hpp. Các khai báo như sau:

```
#include "opencv2/core.hpp"
```

❖ Khai báo không gian tên cv:

Tất cả lớp và hàm trong OpenCV được đặt trong không gian tên là “cv”. Vì vậy, để truy cập các hàm này cần chỉ rõ không gian tên cv:: hoặc khai báo không gian tên cv bằng cú pháp:

```
using namespace cv;
```

❖ Cấu trúc một chương trình OpenCV đơn giản:

```
#include "iostream"

#include "opencv2/core.hpp" //Sử dụng thư viện
opencv

using namespace cv; //Sử dụng namespace cv
int main(int, char**) {
    //Tạo đối tượng ma trận image
    Mat image;
    //Đọc hình ảnh với hệ màu RGB
    //lưu vào đối tượng image
    image = imread("/home/haiau/myimage.png",
CV_LOAD_IMAGE_COLOR);
    //Kiểm tra hình ảnh mở thành công không
    if(!image.data) {
        std::cout    << "Could not read the image"
        << std::endl;
        return -1;
    }
}
```

```

        //Tạo cửa sổ với tên "My image"
        //có kích thước tự động
        namedWindow("My image", WINDOW_AUTOSIZE );
        //Hiển thị hình ảnh lên cửa sổ "My image".
        imshow("My image", image);

        //Đợi nhấn một phím
        waitKey(0);
        return 0;
    }

```

2. Tesseract

Giới thiệu Tesseract:

Tesseract là một thư viện mã nguồn mở hỗ trợ nhận dạng ký tự quang học (OCR - Optical Character Recognition) được phát hành dưới giấy phép mã nguồn mở Apache 2.0. Nó có thể được sử dụng trực tiếp bằng dòng lệnh hoặc thông qua các lời gọi hàm sử dụng API. Tesseract hỗ trợ đa nền tảng và hỗ trợ rất nhiều ngôn ngữ khác nhau như C++, Java, C#, ...

Cài đặt Tesseract:

Bước 1: Cài đặt Tesseract bằng lệnh:

```
$ sudo apt-get install tesseract-ocr
```

Hoặc:

```
$ sudo apt-get install tesseract-vie
```

Bước 2: Xem hướng dẫn sử dụng tesseract bằng lệnh:

```
$ tesseract -help
```

Cú pháp thực thi Tesseract:

Cú pháp chung:

```
$ tesseract imagename textname [-l lang]
```

Ví dụ: Cú pháp sau nhận dạng Tiếng Việt trong hình ảnh myimage.png và xuất kết quả ra file out.txt

```
$ tesseract myimage.png out -l vie
```


3. Praat:

Giới thiệu Praat:

Praat (theo tiếng Hà Lan có nghĩa là “nói”), là một gói phần mềm dùng để xử lý ngôn ngữ tự nhiên được phát hành dưới giấy phép mã nguồn mở GNU GPL. Praat được thiết kế và phát triển bởi Paul Boersma và David Weenink, trường Đại học Amsterdam. Praat có thể chạy trên nhiều nền tảng khác nhau bao gồm Unix, Linux, Mac và Windows. Praat hỗ trợ và đáp ứng nhiều yêu cầu trong công việc nghiên cứu của các nhà khoa học liên quan đến lĩnh vực xử lý tín hiệu tiếng nói, cung cấp các chức năng để trích chọn tham số phục vụ cho nhiều mục đích nghiên cứu khác nhau; ngoài ra Praat cũng hỗ trợ tổng hợp ngôn ngữ tự nhiên và ngôn ngữ script giúp cho việc xây dựng các công cụ và chương trình linh hoạt hơn.



Praat được thiết kế để cung cấp cho những mục đích khác nhau với giao diện dễ sử dụng, có nhiều lựa chọn cho việc học bằng cách thử nghiệm, tham khảo hướng dẫn sử dụng và khả năng phân tích phong phú. Với Praat chúng ta có thể

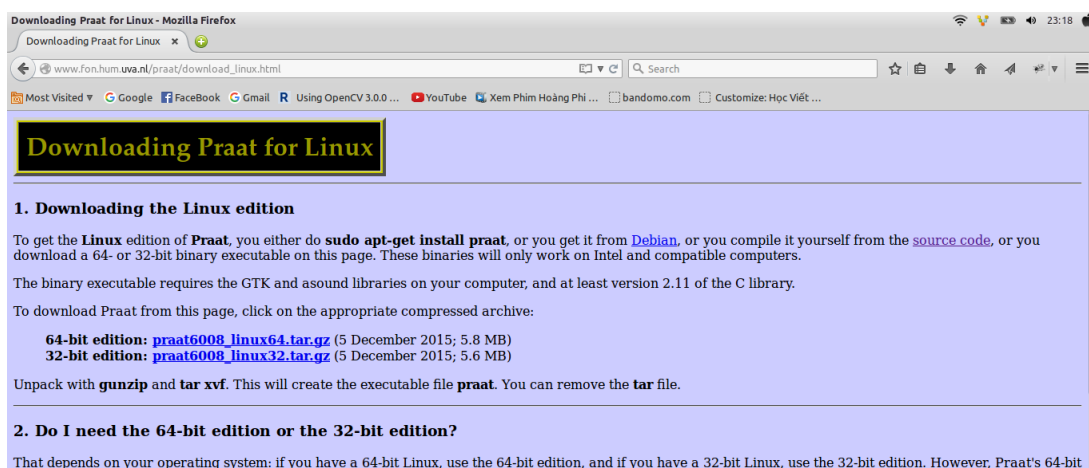
- Tạo ra các đường dạng sóng, đường phổ rộng và hẹp
- Thu âm, chỉnh sửa, xuất ra các đoạn âm thanh riêng biệt thông qua việc phân tích
- Lấy thông tin về độ cao, cường độ, định dạng, sự rung động...
- Tăng cường các vùng tần số nhất định, phân tách và rán nhãn cho các từ, vần hoặc các âm vị.
- Mang công việc của chúng ta trên các khung hình đồ họa.

Cài đặt Praat:

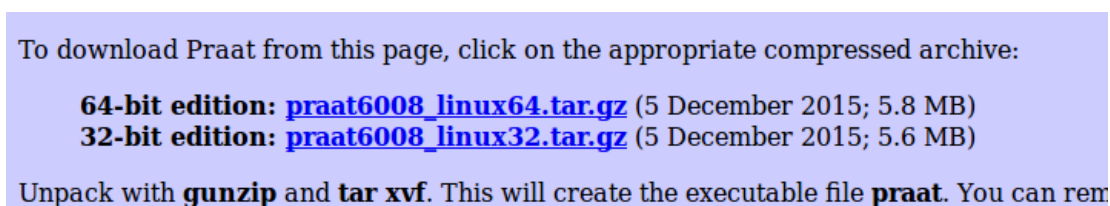
Bài nghiên cứu thực hiện trên nền tảng Linux nên chúng tôi sẽ hướng dẫn cài phần mềm Praat trên Linux (cụ thể là Ubuntu 14.04)

- Vào trình duyệt web truy cập vào địa chỉ:

http://www.fon.hum.uva.nl/praat/download_linux.html



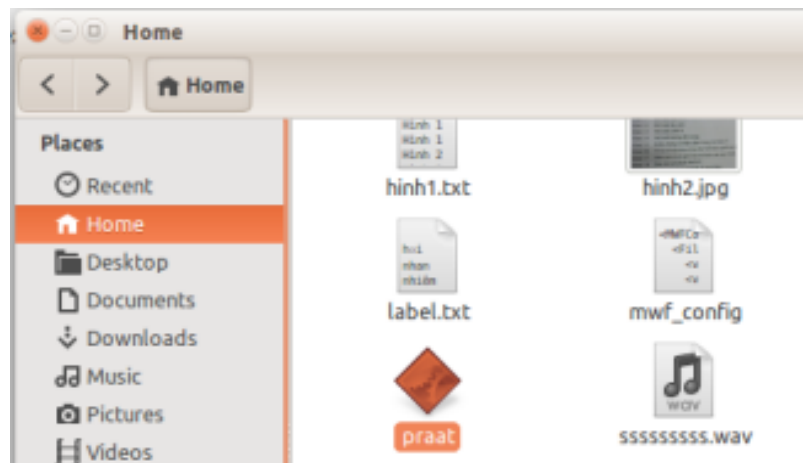
- Click chọn [praat6008_linux64.tar.gz](#) (nếu hệ điều hành 64bit) hoặc Click chọn [praat6008_linux32.tar.gz](#) (nếu hệ điều hành 32bit) để tải phần mềm về máy tính



- Sau khi tải về chúng ta được 1 file nén [praat6008_linux64.tar.gz](#)

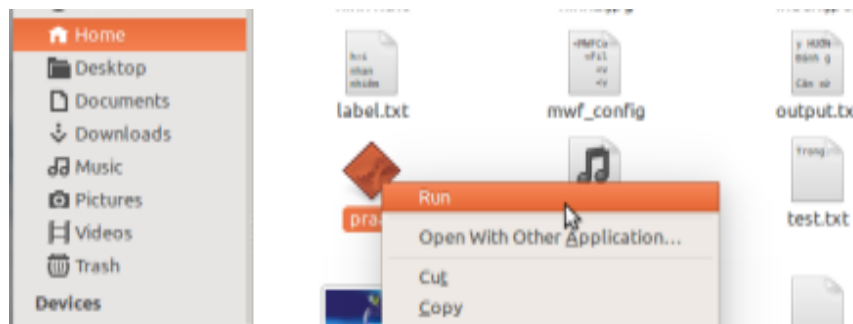


- Tiếp theo ta giải nén file [praat6008_linux64.tar.gz](#) vào thư Mục Người dùng (Home)

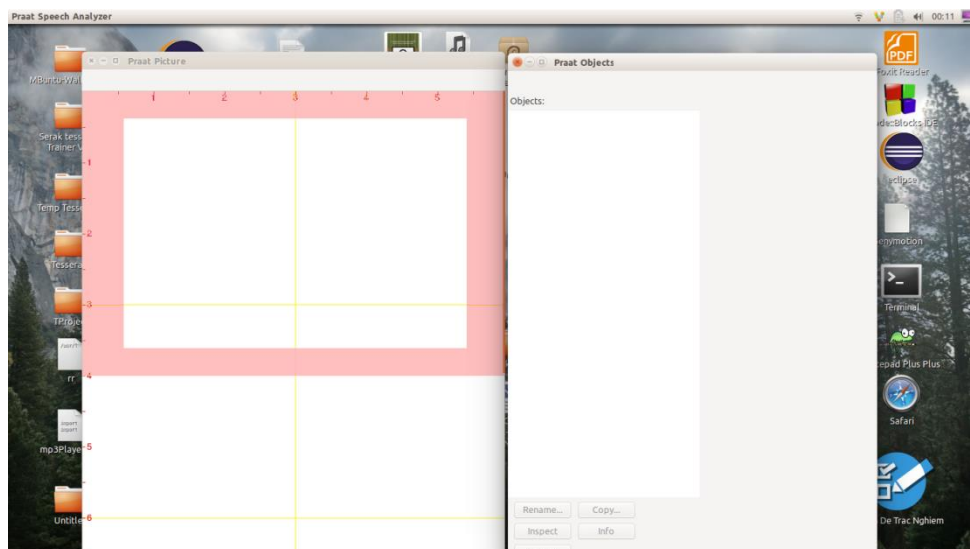


3.2.5. Sử dụng Praat:

- Click chuột phải và chọn run



Giao diện chính của Praat



4. Arduino:

Gới thiệu Arduino:

Arduino là một nền tảng phần cứng mã nguồn mở được phát triển để xây dựng các thiết bị điện tử và các thiết bị tương tác với nhiều loại cảm biến, điều khiển vật lý...

Cài đặt Arduino IDE:

Để cài đặt Arduino IDE trên hệ điều hành Ubuntu, mở Terminal (Ctrl+Alt+T) và thực thi lệnh sau:

```
$ sudo apt-get install arduino
```

Hoặc:

```
$ sudo apt-get install arduino-core
```

Cách sử dụng Arduino:

5. Boost:

Giới thiệu Boost:

Boost là một thư viện hỗ trợ lập trình C++ mã nguồn mở.

Cài đặt Boost:

Để cài đặt thư viện Boost trên hệ điều hành Ubuntu, sử dụng lệnh sau:

```
$ sudo apt-get install libboost-all-dev
```

Sử dụng Boost:

```
Boost.System
```

```
Boost.Thread
```

```
#include <boost/thread.hpp>
```

```
Boost.Asio
```

```
#include <boost/asio.hpp>
```

6. fswebcam:

Giới thiệu fswebcam:

Fswebcam là một gói ứng dụng mã nguồn mở hỗ trợ chụp ảnh bằng webcam (usb camera) trên hệ điều hành Linux.

Cài đặt fswebcam:

```
$ sudo apt-get install fswebcam
```

Sử dụng fswebcam:

Xem hướng dẫn:

```
$ fswebcam -h
```

Lệnh chụp ảnh:

```
$ fswebcam --input 0 --device v4l2:/dev/video1 --  
jpeg 100 --resolution 3264x2448 --no-banner --skip 15 -  
-save /home/haiau/image.jpg --frame 1
```

