**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỒ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**A picture containing text

Description automatically generated**

**ĐỒ ÁN HỌC PHẦN**

TÊN HỌC PHẦN:   
**KỸ THUẬT LẬP TRÌNH ỨNG DỤNG ĐA PHƯƠNG TIỆN   
(Multimedia Application Programming Techniques)**

MÃ SỐ LỚP HP: **MUL14146\_01**

Tên đề tài: **LẬP TRÌNH PHÁT TRIỂN HỆ THỐNG QUẢN LÝ MULTIMEDIA LIÊN QUAN ĐẾN CÁC QUỐC GIA DÙNG CV2 VỚI PYTHON**

Họ tên sinh viên: **NGUYỄN LÊ HOÀNG THANH**

**Mã số sinh viên:N17DCPT058 [13]**

**Lớp: D17CQPU01-N**

**Ngày nộp: ……./……./2021**

**Ký tên:**

**TP.HCM, ngày 10 tháng 11 năm 2021**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỒ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI ĐỒ ÁN HỌC PHẦN**

**Giảng viên giảng dạy: VÕ XUÂN THỂ**

TÊN HỌC PHẦN: **KỸ THUẬT LẬP TRÌNH ỨNG DỤNG ĐA PHƯƠNG TIỆN   
 (Multimedia Application Programming Techniques)**

MÃ SỐ LỚP HP: **MUL14146\_01**

Tên đề tài: **LẬP TRÌNH PHÁT TRIỂN HỆ THỐNG QUẢN LÝ MULTIMEDIA LIÊN QUAN ĐẾN CÁC QUỐC GIA DÙNG CV2 VỚI PYTHON**

Họ tên sinh viên: **NGUYỄN LÊ HOÀNG THANH**

**Mã số sinh viên: N17DCPT058 [13]**

**Lớp: D17CQPU01-N**

**Tên sản phẩm đề tài: 13.NguyenLeHoangThanh.DAHP.MP\_Country**

**Công cụ sử dụng (phiên bản): Spyder 4.1.4 (Anaconda3)  
Thư viên (phiên bản): cv2**

**Ngôn ngữ lập trình sử dụng: Python 3.8.3 64-bit | Qt 5.9.7 | PyQt5 5.9.2 | Windows 10**

**Tập dữ liệu thực nghiệm:**

. Âm thanh: 13NguyenLeHoangThanh.mp3,...

. Hình ảnh: quoc-huy.jpg,viet-nam.jpg,....

. Video Clips: NguyenLeHoangThanh\_13\_Country.mp4,vid.mp4,...

**Nhận xét của giảng viên:**

**………………………………………………………..**

**Điểm đánh giá:………….(…………………)**

**Ngày……./……../2021**

**Giảng viên: Ký tên**

**TP.HCM, ngày 10 tháng 11 năm 2021**

# LỜI CẢM ƠN

Em xin chân thành cảm ơn thầy Võ Xuân Thể - giảng viên bộ môn “KỸ THUẬT LẬP TRÌNH ỨNG DỤNG ĐA PHƯƠNG TIỆN” thuộc khoa Công nghệ thông tin 2 đã chỉ dạy và hỗ trợ cho em để có những kiến thức, kỹ năng cơ bản cần thiết giúp em hoàn thành đề tài.

Tuy nhiên, do kiến thức chuyên ngành của em còn hạn chế nên vẫn có nhiều thiếu sót khi tìm hiểu và trình bày về đề tài. Rất mong nhận được sự góp ý của các thầy/cô để đề tài của em có thể được hoàn chỉnh hơn.

Em xin chân thành cảm ơn.

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
|  | |
| GUI | Graphical User Interface: Giao diện đồ họa người dùng |
| App | Application :ứng dụng, là phần mềm ứng dụng |
| HD | High Definition : Độ nét cao |
| SD | Standard Definition : Độ nét tiêu chuẩn. |
| Mic | Microphone: Ống thu thanh |
| WinApp | Windows Application: Ứng dụng trên hệ điều hành Windows |
|  |  |
| PYTHON | Ngôn ngữ lập trình Python |

# DANH MỤC CÁC THUẬT NGỮ

|  |  |
| --- | --- |
| Giao diện người dùng  (User-Interface) | Là hệ thống các màn hình giao tiếp cho phép người sử dụng tương tác với các thành phần phần mềm, điều khiển phần mềm hoạt động theo yêu cầu của người dùng - tương ứng các chức năng hiện có của phần mềm. |
| Ảo ảnh quang học | là ảo giác thị giác, liên quan đến việc lừa dối thị giác. Do sự sắp xếp của hình ảnh, hiệu ứng của màu sắc, tác động của nguồn ánh sáng hoặc biến khác, một loạt các hiệu ứng hình ảnh gây hiểu lầm có thể được nhìn thấy. |

# 

# MỤC LỤC

[LỜI CẢM ƠN 3](#_Toc89201865)

[DANH MỤC CÁC TỪ VIẾT TẮT 4](#_Toc89201866)

[DANH MỤC CÁC THUẬT NGỮ 4](#_Toc89201867)

[MỤC LỤC 5](#_Toc89201868)

[PHIẾU GIAO NHIỆM VỤ ĐỀ TÀI ĐỒ ÁN HỌC PHẦN 7](#_Toc89201869)

[Chương 1: GIỚI THIỆU VỀ ĐỀ TÀI 9](#_Toc89201870)

[**1.1.** Tổng quan về đề tài 9](#_Toc89201871)

[**1.2.** Nội dung chuyên môn chính của đề tài 9](#_Toc89201872)

[**1.3.** Bố cục của báo cáo 9](#_Toc89201873)

[Chương 2: CƠ SỞ LÝ THUYẾT VỀ KỸ THUẬT LẬP TRÌNH ỨNG DỤNG ĐA PHƯƠNG TIỆN 11](#_Toc89201874)

[**2.1.** TỔNG QUAN VỀ KỸ THUẬT MULTIMEDIA (kỹ thuật đa phương tiện) 11](#_Toc89201875)

[2.1.1. KỸ THUẬT ÂM THANH (GIỌNG NÓI) 11](#_Toc89201876)

[2.1.2. KỸ THUẬT HÌNH ẢNH (IMAGE) 17](#_Toc89201877)

[2.1.3. KỸ THUẬT VIDEO CLIPS 21](#_Toc89201878)

[**2.2.** CÔNG CỤ VÀ CÁC HỆ THỐNG THƯ VIỆN HỖ TRỢ LẬP TRÌNH MULTIMEDIA 23](#_Toc89201879)

[2.2.1. Phần mềm công cụ sử dụng chính trong học phần 23](#_Toc89201880)

[2.2.2. Các thư viện lập trình chính được sử dụng trong học phần 23](#_Toc89201881)

[2.2.3. Tập các Multimedia mẫu được sử dụng trong đề tài của Học phần 23](#_Toc89201882)

[Chương 3: GIỚI THIỆU CÁC THƯ VIỆN MULTIMEDIA SỬ DỤNG 24](#_Toc89201883)

[3.1. Nạp các thư viện 24](#_Toc89201884)

[3.2. Giới thiệu các thư viện được dùng 24](#_Toc89201885)

[Chương 4: GIỚI THIỆU VÀ CÀI ĐẶT GIAO DIỆN GUI SẢN PHẨM 25](#_Toc89201886)

[**4.1.** Mô tả sản phẩm 25](#_Toc89201887)

[**4.2.** Mô tả sản phẩm 25](#_Toc89201888)

[**4.3.** Giới thiệu giao diện và các chức năng 25](#_Toc89201889)

[**4.4.** Thiết kế và Lập trình Winform màn hình chính 26](#_Toc89201890)

[4.4.1. Thiết kế (Design) 26](#_Toc89201891)

[4.4.2. Lập trình (Codes) 26](#_Toc89201892)

[Chương 5: LẬP TRÌNH XỬ LÝ ÂM THANH VÀ GIỌNG NÓI 29](#_Toc89201893)

[5.1. Giới thiệu 29](#_Toc89201894)

[5.2. Thiết kế giao diện và xử lý cơ bản 29](#_Toc89201895)

[5.3. Lập trình xử lý âm thanh 31](#_Toc89201896)

[5.4. Lập trình Xử lý giọng nói 31](#_Toc89201897)

[5.5. Ứng dụng của sản phẩm 32](#_Toc89201898)

[Chương 6: LẬP TRÌNH XỬ LÝ và NHẬN DIỆN ẢNH 33](#_Toc89201899)

[6.1. Giới thiệu 33](#_Toc89201900)

[6.2. Thiết kế giao diện và xử lý cơ bản 33](#_Toc89201901)

[6.3. Lập trình xử lý ảnh và khung ảnh 36](#_Toc89201902)

[6.4. Lập trình xử lý nhận diện ảnh (tĩnh) 39](#_Toc89201903)

[6.5. Ứng dụng của sản phẩm 41](#_Toc89201904)

[Chương 7: LẬP TRÌNH XỬ LÝ VIDEO VÀ NHẬN DIỆN LIÊN QUAN 44](#_Toc89201905)

[7.1. Giới thiệu 44](#_Toc89201906)

[7.2. Thiết kế giao diện và xử lý Video cơ bản 44](#_Toc89201907)

[7.3. Lồng ghép Audio và Video kết Images 49](#_Toc89201908)

[**7.4.** **Ứng dụng của sản phẩm** 49](#_Toc89201909)

[Chương 8: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI 51](#_Toc89201910)

[**8.1.** **Kết luận** 51](#_Toc89201911)

[8.1.1. Những kết quả đạt được 51](#_Toc89201912)

[8.1.2. Hạn chế 51](#_Toc89201913)

[**8.2.** **Hướng phát triển** 51](#_Toc89201914)

[8.2.1. Hướng khắc phục các hạn chế 51](#_Toc89201915)

[8.2.2. Hướng mở rộng đề tài 51](#_Toc89201916)

[DANH MỤC TÀI LIỆU THAM KHẢO 52](#_Toc89201917)

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỒ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

# PHIẾU GIAO NHIỆM VỤ ĐỀ TÀI ĐỒ ÁN HỌC PHẦN

|  |  |
| --- | --- |
| Giảng viên giảng dạy: **VÕ XUÂN THỂ** | |
| Tên học phần: **KỸ THUẬT LẬP TRÌNH ỨNG DỤNG ĐA PHƯƠNG TIỆN** Mã số lớp HP: **MUL14146\_01** | |
| Tên đề tài: **LẬP TRÌNH PHÁT TRIỂN HỆ THỐNG QUẢN LÝ MULTIMEDIA LIÊN QUAN ĐẾN CÁC QUỐC GIA DÙNG CV2 VỚI PYTHON.** | |
| Sinh viên thực hiện: **Nguyễn Lê Hoàng Thanh, N17DCPT058** | |
| Thời gian thực hiện: **29/09/2021**  đến **10/11/2021** | |
| **Yêu cầu của đề tài**  **Lý thuyết:** Vận dụng kiến thức KỸ THUẬT LẬP TRÌNH ỨNG DỤNG ĐA PHƯƠNG TIỆN vào việc phát triển một hệ thống ứng dụng quản lý đa phương tiện dùng trong một lĩnh vực chuyên môn thực tiễn, minh họa với ngôn ngữ lập trình PYTHON dùng công cụ Spyder (Anaconda3) với một số thư viện hỗ trợ lập trình ứng dụng đa phương tiện, như: cv2, speech\_recognition , gTTS … tham khảo tập multimedia mẫu https://www.flaticon.com/ và files mẫu: NguyenLeHoangThanh\_13\_Country.mp4, ….  **Thực hành:** LẬP TRÌNH PHÁT TRIỂN HỆ THỐNG QUẢN LÝ MULTIMEDIA LIÊN QUAN ĐẾN CÁC QUỐC GIA DÙNG CV2 VỚI PYTHON; gồm:  Nạp Media:  ++Lựa chọn thao tác xử lí bằng giọng nói hoặc bằng thao tác (ít nhất 2 ngôn ngữ :in-out)  ++Nạp từ file video, image ,audio, //mic, camera : video giới thiệu quốc gia (img:quốc kỳ , quốc huy, audio:quốc ca, video: quảng bá, vid quốc ca)  ++việc dùng giọng nói có thể chọn 1 số ngôn ngữ khác nhau  Nguồn dữ liệu:  Các xử lí bao gồm:  +Ra lệnh bằng giọng nói  +trích khung ảnh từ video:quốc kỳ, quốc huy rõ nét nhất  +xử lí liên quan đến ảnh trích ra : trắng đen, mờ, cắt 1 phần khung ảnh từ bối cảnh chung(chuột hoặc tọa độ),  Giao diện đồ họa hoặc text  Kết xuất:  +giọng nói tên quốc gia, trích bài quốc ca+quốc kỳ, quốc huy  + Thư viện chính được dùng: cv2 #thư viện Computer Vision version 2,  + Thiết kế và phát triển giao diện GUI dùng thư viện: Tkinter.  + Xử lý trợ lý âm thanh (Voice Assistant)  speech\_recognition: Chuyển giọng nói thành văn bản.  gtts: Chuyển văn bản thành giọng nói.  playsound: Phát file audio.  Ffpyplayer: Trình phát media.  + Xử lý ảnh: cv2, PIL .  + Xử lý Video: cv2, ….  + Các thư viện hỗ trợ khác:  . os # Thư viện os của MS. Windows = Lập thư mục & lưu các files Media  . numpy #Thư viện Mummeric Python: Lập ma trận giữ chỗ đề ghép các Images và Videos  .datetime#Thư viện hiển thị ngày giờ  .googletrans#Thư viện dịch thuật của Google | |
| **GIẢNG VIÊN** | **Ngày 29 tháng 09 năm 2021**  **SV Thực hiện** |

# GIỚI THIỆU VỀ ĐỀ TÀI

## Tổng quan về đề tài

Tìm hiểu về kỹ thuật đa phương tiện (bao gồm các kỹ thuật về âm thanh, giọng nói, hình ảnh, video clip,..) đồng thời dựa trên công cụ lập trình Spyder (Anaconda3) với ngôn ngữ lập PYTHON, sử dụng các thư viện hỗ trợ kỹ thuật lập trình ứng dụng đa phương tiện (Multimedia Program), như: cv2, gtts, PlayAudio, speech\_recognition, PIL,...để phát triển ứng dụng (App) dạng Winforms (tức là WinApp) cho phép quản lý và xử lý các tập tin multimedia sử dụng trong LẬP TRÌNH PHÁT TRIỂN HỆ THỐNG QUẢN LÝ MULTIMEDIA LIÊN QUAN ĐẾN CÁC QUỐC GIA DÙNG CV2 VỚI PYTHON

Tập dữ liệu thực nghiệm chính đước tham khảo (thu thập) từ <https://www.flaticon.com/>(icon), <https://vi.wikipedia.org/> (Quốc ca, quốc huy, quốc kỳ) và một số files multimedia mẫu:

.Videos: vid.mp4, NguyenLeHoangThanh\_13\_Country.mp4…

. Images: QK.jpg, QH.png,...

. Audio: 13NguyenLeHoangThanh.mp3,QC.mp3….

## Nội dung chuyên môn chính của đề tài

+ Thư viện: opencv với ngôn ngữ Python

+ Giải thuật gì: Xử lý media dùng thư viện với ngôn ngữ lập trình PYTHON

+ Sản phẩm đề tài: Tên sản phẩm: **13.NguyenLeHoangThanh.DAHP.MP\_Country**

ỨNG DỤNG QUẢN LÝ & XỬ LÝ MULTIMEDIA: LẬP TRÌNH BỞI NGUYỄN LÊ HOÀNG THANH - HỌC VIÊN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG TP.HCM

+ Nhập Media từ 2 nguồn: Files có sẵn + trực tiếp từ MICrophne, CAMera.

## Bố cục của báo cáo

Chương 1: Giới thiệu tổng quan về đề tài lập trình Multimedia quản lý và xử lý các đối tượng đa phương tiện liên quan đến LẬP TRÌNH PHÁT TRIỂN HỆ THỐNG QUẢN LÝ MULTIMEDIA LIÊN QUAN ĐẾN CÁC QUỐC GIA DÙNG CV2 VỚI PYTHON

Chương 2: Cơ sở lý thuyết của học phần: gồm những lý thuyết cơ bản về

+ Kỹ thuật Multimedia liên quan đến đề tài.

+ Kỹ thuật lập trình Winforms dùng thư viện Tkinter và xử lý media dùng thư viện cv2 và gtts, trên Spyder (Anaconda3) với ngôn ngữ Python

Chương 3: Giới thiệu về thư viện và thủ tục nạp các thư viện vào App

Chương 4: Thiết kế và lập trình giao diện GUI

Chương 5: Thiết kế và lập trình sản phẩm của đề tài: quản lý và xử lý âm thanh LẬP TRÌNH PHÁT TRIỂN HỆ THỐNG QUẢN LÝ MULTIMEDIA LIÊN QUAN ĐẾN CÁC QUỐC GIA DÙNG CV2 VỚI PYTHON

Chương 6: Thiết kế và lập trình sản phẩm của đề tài: quản lý và xử lý hình ảnh LẬP TRÌNH PHÁT TRIỂN HỆ THỐNG QUẢN LÝ MULTIMEDIA LIÊN QUAN ĐẾN CÁC QUỐC GIA DÙNG CV2 VỚI PYTHON

Chương 7: Thiết kế và lập trình sản phẩm của đề tài: quản lý và xử lý Video Clips LẬP TRÌNH PHÁT TRIỂN HỆ THỐNG QUẢN LÝ MULTIMEDIA LIÊN QUAN ĐẾN CÁC QUỐC GIA DÙNG CV2 VỚI PYTHON

Chương 8: Kết luận về kết quả đạt được và những tồn tại, trên cơ cở đó đề xuất các giải pháp khắc phục tồn và hướng mở rộng đề tài.

# CƠ SỞ LÝ THUYẾT VỀ KỸ THUẬT LẬP TRÌNH ỨNG DỤNG ĐA PHƯƠNG TIỆN

## TỔNG QUAN VỀ KỸ THUẬT MULTIMEDIA (kỹ thuật đa phương tiện)

### KỸ THUẬT ÂM THANH (GIỌNG NÓI)

#### Âm thanh

+ Là một hiện tượng vật lý được tạo ra bởi sự rung động của các phân tử, nguyên tử hay các hạt làm nên vật chất và sự rung động đó lan truyền qua một môi trường trung gian dưới dạng sóng đến tai người nghe. Sóng âm đập vào và làm rung động màng nhĩ, kích thích não bộ khiến chúng ta nghe được âm thanh.

+ Các vật có thể phát ra âm thanh được gọi là nguồn âm. Trong quá trình phát ra âm thanh, các nguồn âm đều dao động.

#### Các đặc tính cơ bản của âm thanh

+ Cao độ của âm thanh ≈ Tần số âm thanh (sound frequency [Hz]):   
 Âm trầm (bass), Âm trung (mid), Âm cao (treble)

+ Cường độ âm thanh (sound intensity) [deciBel = dB = 1/10 B || Bel = B]

+ Trường độ của âm thanh = Băng thông (Độ trễ || Bandwidth) [Hz]

+ Âm sắc (Timbre) (theo đối tượng phát ra âm).

NGOÀI RA, CÒN CÓ:

+ Thanh áp (áp suất âm thanh, ký hiệu p [miliBar = mB, Bar])

+ Công suất âm thanh P = psv [Watt] { s : thiết diện, v: vận tốc âm thanh, p: thanh áp}

+ Độ xuất hiện đột ngột (transients)

+ Độ rung (vibrato)

+ Độ thay đổi (envelope modulation).

. . .

CHÚ Ý:

+ Tone (tông) của âm thanh [“Giọng” của âm thanh] = trường độ + cường độ + cao độ + âm sắc

VD: Tông Nam, Tông Nữ trong Karaoke

#### Các đặc tính cơ bản của file lưu trữ âm thanh (Digital)

+ Sample (giá trị biên độ tần số lấy mẫu)

+ Sample Rate (Số lần lấy mẫu/s) [T]=> Sample Frequency (tần số lấy mẫu) [f]; VD: 44100 Hz

+ BitDepth (số lượng bit số trong mẫu được lấy, còn gọi là độ “nét”: resolution); VD: 16 bits, 24 bits

+ Channel (số kênh tín hiệu số được tách ra); VD 2 kênh

Đây là cơ sở xác định âm thanh Mono (âm thanh “phẳng” = đơn kênh = 1 channel) hay   
 Stereo (âm thanh “nổi” = đa kênh); có nhiều dạng

2D = 2 channel = kênh “trái” và kênh “phải” = 2 chiều

3D (3 Dimensions = 3 chiều) = 3 kênh, . . .

Ghi âm lời nói nên Mono, nhưng Âm nhạc thì nên Stereo, 2D, 3D, Surround,…

+ Bit-Rate (số lượng bit số của âm thanh/s) [kbps (Kilobits per second)] ;  
 VD: Một phút nhạc 128 kbps có dung lượng khoảng 1 MB và   
 bản nhạc 320 kbps thì chắc chắn sẽ hay hơn bản nhạc 128 kbps.

#### Các dạng đuôi (phần mở rộng || Extensions) file âm thanh

Trên Internet công cụ online (offline=Installed) cho phép chuyển kiểu file audio . => .

.mp3 (MPEG-1 audio Player 3 || Motion Pictures Expert Group 1 Layer 3)

= nén có mất thông tin Kich thức file rất bé, nhưng âm thanh không còn tốt như ban đầu (do nén âm)

.wma (Windows Media Audio) = File âm thanh đa phương tiện của hệ điều hành Windows

.wav (Waveform Audio File Format)

= File âm thanh nguyên bản (dạng sóng)= trung thực, kích thước lớn

.mid (Musical Instrument Digital) || midi (Musical Instrument Digital Interface) file

= File âm thanh kỹ thuật số của các nhạc cụ (lưu thông tin của file âm thanh)

.m4a MPEG-4 Compressed Audio File

.flac (Free Lossless Audio Codec) = File âm thanh mã hóa không mất mát

.alac / m4a file (Apple Lossless Audio Code) = Files âm thanh mã hóa không mất mát của Apple

.aac = .ac (Advanced Audio Coding) = File âm thanh mã hóa cải tiến

.ogg (file âm thanh sử dụng phương pháp nén Ogg Vorbis) = files âm thanh nén theo giải thuật OGG

.aif Audio Interchange File Format Sound File

.aiff Audio Interchange File Format

**NGOÀI RA, CÒN NHIỀU EXTENSION KHÁC, NHƯ**

.mod: amiga MODules = File âm thanh điều chế tín hiệu theo hệ điều hành AmigaOS

.it: Impulse Tracker Music Module File

.s3d: Scream Tracker 3

.xm: Fast Tracker 2

.ftm FamiTracker Module File

.amr: Adaptive Multi-rate Compressed Audio File

.m3u Tập tin Playlist các bài hát MP3

.ptm PolyTracker Module

.voc Creative Labs Sound File

.xspf XML Shareable Playlist Format

.dss Digital Speech Standard File

.aax Audible Enhanced Audio File

.abc ABC Music Notation File

.cda CD Audio Track File

.acd ACID Project File

.aaf Advanced Authoring Format Multimedia File

.ac3 Audio Codec 3 File

.act Adpcm Compressed Audio File

.ape Monkey's Audio Lossless Audio File

.aud Westwood Studios Audio File

.aup Audacity Project File

.dvf Sony DV Voice File

.flp FruityLoops Project File

.gsm Global System for Mobile Audio File

.iff Autodesk Maya Image Interchange File

.kar Karaoke MIDI File

.m4p Apple Itunes Music Store Audio File

.mod Amiga Music Module File

.mpc Musepack Audio File

.msv Mystars! Saved View File

.mus Finale Notation File

.nra Nero Audio-cd Compilation File

.odm Overdrive Media Console Media Control File

.pcm Pulse Code Modulation File

.pk Audition Peak File

.pls Audio Playlist File

.ram Real Audio Metafile

.rbs Rebirth Song File

.sds MIDI Sample Dump Standard File

.seq PowerTracks Pro Audio Project File

.ses Adobe Audition Session File

.shn Shorten Compressed Audio File

.spx Ogg Vorbis Speex File

.vlc VLC Playlist File

.wv Wavpack Lossless Compressed Audio File

.dcf Drm Content Format Delivery File

.gp3 Guitar Pro 3 Project File

.mtd Musicnotes Digital Music Sheet File

.nwc Noteworthy Composer Song File

.pc Personal Composer Musical Notation File

.qcp PureVoice Audio File

.sf2 Creative Labs Soundfont 2.0 Bank File

.sng MIDI Song File

.vpl Karaoke Player Playlist File

.wrf Webex Recording File

.zpl Zune Playlist File

.aa Audible Audio Book File

.acm Interplay Audio File

.aifc Compressed Audio Interchange File

.apl Monkey's Audio Track Info File

.cpr Cubase Project File

.dmf Delusion Digital Music File

.imf Id Music File

.kmp Korg Trinity Audio File

.m4b MPEG-4 Audio Layer Audio Book File

.m4r Apple Iphone Ringtone Format

.mbr Zune Smooth Streaming File

.mxl Compressed Musicxml Music Notation File

.nst NoiseTracker Module File

.oma Sony Openmg Music Format File

.ptx Pro Tools 10 Session File

.ra Realaudio Audio File

.rmi RMID MIDI File

.rng Nokia Composer Ringtone File

.smf Standard MIDI File

.smp SmartMusic Performance File

.tak Tom's Lossless Audio Kompressor File

.tta True Audio File

.vox Dialogic Voice Audio File

.wax Windows Media Audio Redirector File

.wpk Nero Wave Editor File

.wrk Cakewalk Music Project File

.wve WaveEditor Project File

.all Cubasis Project File

.als Ableton Live Set File

.amz Amazon Mp3 Downloader File

.bun Cakewalk Bundle File

.emp eMusic Music Download File

.emx Emusic Download File

.fev FMOD Audio Events File

.gig Gigasampler/gigastudio Audio File

.gpk WaveLab Audio Peak File

.imp Audition Impulse File

.lvp Avaya Voice Player Compressed Voice Audio File

.mtp Minitab Portable Worksheet File

.omf Open Media Framework File

.ove Overture Music Score File

.pts Pro Tools Session File

.rfl Reason ReFill Sound Bank File

.sdat Nintendo Ds Sound Data File

.sfl Sound Forge Data File

.sm Stepmania Song File

.svd Roland Patch File

.syn SimSynth Document File

.vdj Virtualdj Audio Sample File

.vrf Ventrilo Audio Recording File

.wow Grave Composer Module File

. . .

#### Các dạng nén file âm thanh thông dụng dựa vào BitRate

+ CBR (Constant BitRate) = Dùng giá trị BitRate cố đinh (chọn), đơn giản, ít hiệu quả

+ ABR (Average BitRate) = Dùng giá trị bình quân của BitRate, phức tạp hơn, hiệu quả

+ VBR (Variable BitRate) = Dùng các giá trị BitRate khác biệt cho từng đoạn nén, rất phức tạp, hiệu quả rất cao

#### Một số dạng xử lý chất lượng âm thanh

như:

+ Xử lý tạp âm (Noise Processing) và lọc nhiễu (Noise Filter) dùng “bộ lọc”

+ Cộng hưởng âm= Cộng hưởng qua tần số của âm (cùng tần số f sẽ cộng hưởng âm)

= Khuếch đại âm thanh

+ Trích lọc âm và tách âm

+ Hòa âm (Kết hợp các dạng âm thanh) – phối khí (phối hợp các nhạc cụ)

+ Âm thanh Hi-Fi (Chất lượng cao) và Hi-End

+ Hợp âm (chord) trong âm nhạc: Đồ, Rê, Mi, . . .= kết hợp nhiều âm thanh (âm thanh phát ra từ nhiều dây đàn)

. . .

#### Một số kỹ thuật thu thập âm thanh (Thu âm)

Chương trình phần mềm, có thể:

+ Ghi âm

+ Nạp files âm thanh

#### Các xử lý cơ bản với đặc tính âm thanh kỹ thuật số

+ Xử lý đặc tính âm thanh:

. Cao độ (Bass, treble),

. tăng/giảm dB, (khuếch đại âm thanh)

. chuyển âm sắc,

. chuyển tông/Tone..

### KỸ THUẬT HÌNH ẢNH (IMAGE)

#### Hình ảnh

Ảnh thứ ghi lại hay thể hiện/tái tạo được cảm nhận thị giác, tương tự với cảm nhận thị giác từ vật thể có thật, do đó mô tả được những vật thể đó.

Ảnh có thể biểu diễn dưới dạng tín hiệu tương tự hoặc tín hiệu số.

Ảnh số là đối tượng biểu diễn hình ảnh trong máy tính để xử lý .Ảnh số là một ma trận 2 chiều, và việc xử lý chúng là những thao tác trên ma trận này sao cho ra kết quả hợp lý.

Pixel là đơn vị nhỏ nhất của hình ảnh hoặc đồ họa kỹ thuật số có thể được hiển thị và biểu diễn trên thiết bị hiển thị kỹ thuật số. Một pixel là đơn vị logic cơ bản trong đồ họa kỹ thuật số. Các điểm ảnh được kết hợp để tạo thành một hình ảnh, video, văn bản hoàn chỉnh hoặc bất kỳ thứ nào có thể nhìn thấy trên màn hình máy tính.

Độ phân giải (Resolution) là mật độ điểm ảnh của hình ảnh được hiển thị. Nó thường được trích dẫn là Chiều rộng x Chiều cao tính trên đơn vị pixels, các chỉ số này càng lớn thì màn hình hiển thị càng chi tiết.

Aspect Ratio (Tỷ lệ khung hình ) là mối quan hệ giữa chiều rộng và chiều cao của hình ảnh. Để biểu diễn tỉ lệ thông thường người sử dụng hai số dương viết liền với nhau với dấu hai chấm ở giữa, chẳng hạn như 16:9. Với một hình ảnh có tỉ lệ x:y thì x được xem là chiều rộng, y là chiều cao và hai biến này có quan hệ tỉ lệ với nhau.

#### Tiến trình xử lý ảnh

. Tiến trình xử lỷ ảnh:

Thu nhận ảnh →tiền xử lí→phân đoạn ảnh→tách các đặc tính→nhận dạng và giải thích

**Thu nhận ảnh**: Đây là công đoạn đầu tiên mang tính quyết định đối với quá trình xử lý ảnh. Ảnh đầu vào sẽ được thu nhận qua các thiết bị như camera, sensor, máy scanner,v.v… và sau đó các tín hiệu này sẽ được số hóa. Việc lựa chọn các thiết bị thu nhận ảnh sẽ phụ thuộc vào đặc tính của các đối tượng cần xử lý. Các thông số quan trọng ở bước này là độ phân giải, chất lượng màu, dung lượng bộ nhớ và tốc độ thu nhận ảnh của các thiết bị.

**Tiền xử lý**: Ở bước này, ảnh sẽ được cải thiện về độ tương phản, khử nhiễu, khử bóng, khử độ lệch,v.v… với mục đích làm cho chất lượng ảnh trở lên tốt hơn nữa, chuẩn bị cho các bước xử lý phức tạp hơn về sau trong quá trình xử lý ảnh. Quá trình này thường được thực hiện bởi các bộ lọc.

**Phân đoạn ảnh**: phân đoạn ảnh là bước then chốt trong xử lý ảnh. Giai đoạn này

phân tích ảnh thành những thành phần có cùng tính chất nào đó dựa theo biên hay các

vùng liên thông. Tiêu chuẩn để xác định các vùng liên thông có thể là cùng màu, cùng

mức xám v.v… Mục đích của phân đoạn ảnh là để có một miêu tả tổng hợp về nhiều

phần tử khác nhau cấu tạo lên ảnh thô. Vì lượng thông tin chứa trong ảnh rất lớn,

trong khi đa số các ứng dụng chúng ta chỉ cần trích một vài đặc trưng nào đó, do vậy

cần có một quá trình để giảm lượng thông tin khổng lồ đó. Quá trình này bao gồm

phân vùng ảnh và trích chọn đặc tính chủ yếu. **Tách các đặc tính**: Kết quả của bước phân đoạn ảnh thường được cho dưới dạng

dữ liệu điểm ảnh thô, trong đó hàm chứa biên của một vùng ảnh, hoặc tập hợp tất cả

các điểm ảnh thuộc về chính vùng ảnh đó. Trong cả hai trường hợp, sự chuyển đổi dữ

liệu thô này thành một dạng thích hợp hơn cho việc xử lý trong máy tính là rất cần

thiết. Để chuyển đổi chúng, câu hỏi đầu tiên cần phải trả lời là nên biểu diễn một vùng

ảnh dưới dạng biên hay dưới dạng một vùng hoàn chỉnh gồm tất cả những điểm ảnh

thuộc về nó. Biểu diễn dạng biên cho một vùng phù hợp với những ứng dụng chỉ quan

tâm chủ yếu đến các đặc trưng hình dạng bên ngoài của đối tượng, ví dụ như các góc

cạnh và điểm uốn trên biên chẳng hạn. Biểu diễn dạng vùng lại thích hợp cho những

ứng dụng khai thác các tính chất bên trong của đối tượng, ví dụ như vân ảnh hoặc cấu

trúc xương của nó. Sự chọn lựa cách biểu diễn thích hợp cho một vùng ảnh chỉ mới là một phần trong việc chuyển đổi dữ liệu ảnh thô sang một dạng thích hợp hơn cho các xử lý về sau. Chúng ta còn phải đưa ra một phương pháp mô tả dữ liệu đã được chuyển đổi đó sao cho những tính chất cần quan tâm đến sẽ được làm nổi bật lên, thuận tiện cho việc xử lý chúng.

**Nhận dạng và giải thích**: Đây là bước cuối cùng trong quá trình xử lý ảnh. Nhận dạng ảnh có thể được nhìn nhận một cách đơn giản là việc gán nhãn cho các đối tượng trong ảnh. Ví dụ đối với nhận dạng chữ viết, các đối tượng trong ảnh cần nhận dạng là các mẫu chữ, ta cần tách riêng các mẫu chữ đó ra và tìm cách gán đúng các ký tự của bảng chữ cái tương ứng cho các mẫu chữ thu được trong ảnh. Giải thích là công đoạn gán nghĩa cho một tập các đối tượng đã được nhận biết.

#### Histogram

Trong lĩnh vực xử lí ảnh, histogram là biểu đồ tần xuất được dùng để thống kê số lần xuất hiện các mức sáng trong ảnh.

Histogram cung cấp cho những thông cơ bản, như độ sáng và độ tương phản (contrast)

của ảnh. Độ tương phản đặc trưng cho sự thay đổi độ sáng của đối tượng so với nền. Có thể nói, độ tương phản là độ nổi của điểm ảnh hay vùng ảnh so với nền.

Cân bằng histogram (histogram equalization) là sự điều chỉnh histogram về trạng thái cân bằng, làm cho phân bố (distribution) giá trị pixel không bị co cụm tại một khoảng hẹp mà được "kéo dãn" ra. Cân bằng histogram là một phương pháp tiền/hậu xử lí ảnh rất mạnh.

#### Hệ màu (không gian màu) RGB

Không gian màu RGB là mô hình bổ sung trong đó ánh sáng đỏ, xanh lá cây và xanh lam được tổ hợp với nhau theo nhiều phương thức khác nhau để tạo thành các màu khác. Từ viết tắt RGB trong tiếng Anh có nghĩa là đỏ (red), xanh lục (green) và xanh lam (blue), là ba màu gốc trong các mô hình ánh sáng bổ sung.

#### Một số thao tác xử lý ảnh thông dụng

Xoay ảnh

Co dãn

Cắt ảnh theo khung (frame)

Chuyển xám (3D →2D)

Làm mờ

#### Nhận diện ảnh bằng trích xuất đặc trưng (SIFT)

SIFT (Scale-invariant feature transform) là một feature descriptor được sử dụng trong computer vision và xử lý hình ảnh được dùng để nhận dạng đối tượng, matching image, hay áp dụng cho các bài toán phân loại...

Với đầu vào là một hình ảnh >>> SIFT >>> các keypoint. Mỗi đối tượng trong hình ảnh sẽ cho ra rất nhiều các keypoint khác nhau, để ta phân biệt được các keypoint này với nhau sẽ thông qua một vector 128 chiều hay còn gọi là descriptor. Các descriptor này sẽ được dùng để nhận dạng đối tượng trong ảnh, hay dùng cho các bài toán classication.

Sau khi áp dụng biến đổi SIFT, ứng với mỗi keypoint ta sẽ thu được: toạ độ keypoint, scale và orientation của keypoint, descriptor. Các mũi tên trong hình dưới vẽ nhờ vào scale và orientation. Đặc điểm :

Các keypoint sẽ ít bị phụ thuộc bởi cường độ sáng, nhiễu, góc xoay của ảnh do các descriptor được tạo ra từ gradients do đó nó đã bất biến với các thay đổi về độ sáng (ví dụ: thêm 10 vào tất cả các pixel hình ảnh sẽ mang lại cùng một mô tả chính xác).

Nhanh và hiệu quả, tốc độ xử lý gần như với thời gian thực (realtime)

Có thể xử lý khi xoay ảnh

#### Ảnh màu (color = 3 D) và Gray (Ảnh trắng đen = 2D)

Ảnh màu là ma trận pixel.Mỗi pixel được biểu diễn bằng 3 màu red, green, blue. Mỗi màu có giá trị từ 0-255

Ảnh xám là ma trận pixel.Mỗi pixel được biểu diễn trị nguyên trong khoảng từ [0,255] .

### KỸ THUẬT VIDEO CLIPS

#### Video Clips

Video hoặc video clip là một chuỗi các tín hiệu điện tử được sử dụng để tạo ra nguồn ảnh tĩnh ổn định, mô phỏng chuyển động. Video có thể sử dụng đồ họa, âm thanh, hình ảnh hoặc văn bản và được sử dụng cho mục đích giải trí, giáo dục hoặc các mục đích khác.

Video phải có ít nhất 24 khung hình trên giây để video trở nên mượt mà.

#### Video Analog và Video kỹ thuật số (Digital)

Tín hiệu Analog hay tín hiệu tương tự là tín hiệu được truyền liên tục (không ngắt quãng) thay đổi theo thời gian. Nghĩa là sau một chu kỳ, tín hiệu sẽ được lặp lại và chỉ khác nhau về cường độ. Về mặt lý thuyết, loại tín hiệu này có thể giữ nguyên hình dạng đến vô hạn, nhưng khi truyền thực tế, nó sẽ bị ảnh hưởng bởi sự nhiễu sóng, dẫn đến méo tín hiệu. Một số tín hiệu Analog mà chúng ta gặp hàng ngày như âm thanh, ánh sáng, tốc độ gó, tần số,...

Tín hiệu Digital là tín hiệu số (tín hiệu rời rạc), được sử dụng để biểu diễn dữ liệu dưới dạng chuỗi các giá trị rời rạc ở bất kỳ thời điểm nào. Tín hiệu kỹ thuật số chỉ có thể nhận một trong hai mức giá trị cố định là cao và thấp. Do đó, trong khoa học máy tính, mức giá trị này được biểu thị bằng mã nhị phân là 1 (cao) và 0 (thấp), còn trong công nghiệp thì nó là On/Off. Loại tín hiệu này đại diện cho một số thực trong phạm vi giá trị không đổi.

Video tương tự biểu thị hình ảnh video dưới dạng tín hiệu tương tự.

Video kỹ thuật số biểu thị hình ảnh video dưới dạng dữ liệu kỹ thuật số được mã hóa.

#### Video nén

.MP4 : MPEG-4 Part 14

.WMV :Windows Media Video

.AVI :Audio Video Interleave

.3GP :3GPP Multimedia File (3rd Generation Partnership Project)

.WebM :Web Movie

.FLV : Flash Video

#### Frames và Frame rate

Frame hay khung hình là một trong nhiều hình ảnh tĩnh tạo nên video. Mỗi khung hình sẽ được chiếu trên màn hình trong một thời gian ngắn (ngày nay thường là 1/24, 1/25 hoặc 1/30 giây) và sau đó được thay thế ngay lập tức bằng khung hình tiếp theo. Kết hợp với khả năng duy trì thị lực của võng mạc tạo ra ảo ảnh quang học hình ảnh chuyển động.

Tốc độ khung hình hay tốc độ thay đổi khung hình (frame rate) là một trong những đại lượng cơ bản của kỹ thuật điện ảnh, đó là số khung hình xuất hiện trước mắt khán giả trong một đơn vị thời gian(1 giây) với tốc độ quay ổn định, được tính bằng số khung hình trên một giây (frame per second hay fps)

Về bản chất, một khung hình là một hình ảnh nên nó có các đặc trưng của hình ảnh như: độ phân giải, tỉ lệ khung hình,....

PHÂN LOẠI ĐỘ PHÂN GIẢI VIDEO:

Có hai loại độ phân giải video: định dạng chuẩn (SD) và độ phân giải cao (HD).

Độ phân giải SD là viết tắt của Standard Denfinition có nghĩa là hình ảnh có độ nét tiêu chuẩn. Chuẩn SD có độ phân giải phổ biến nhất là 640 x 360, 640 x 480 cho video và 720 x 480, 720 x 576 cho đĩa DVD. SD là một định dạng cũ, có tỉ lệ khung hình và số lượng điểm ảnh khá thấp, nên chất lượng hình ảnh chỉ đạt mức trung bình. Và tất nhiên dung lượng của những video này cũng rất nhẹ.

video HD thường có độ phân giải 1280 x 720 (720p) hoặc 1920 x 1080 (1080p, còn được gọi là Full HD). Có thể thấy video HD có mật độ điểm ảnh nhiều và sở hữu một độ sắc nét và chi tiết cao. Tuy nhiên, khuyết điểm của các video này là dung lượng lớn và cần cấu hình cao để xử lý video.

Độ phân giải và tỷ lệ khung hình khuyên dùng:

Tỷ lệ khung hình 16:9 mã hóa ở các độ phân giải sau:

2160p: 3840x2160

1440p: 2560x1440

1080p: 1920x1080

720p: 1280x720

480p: 854x480

360p: 640x360

240p: 426x240

## CÔNG CỤ VÀ CÁC HỆ THỐNG THƯ VIỆN HỖ TRỢ LẬP TRÌNH MULTIMEDIA

### Phần mềm công cụ sử dụng chính trong học phần

\* Công cụ chính: Spyder (Anaconda3)

\* Ngôn ngữ lập chính: Python

\* Thư viện: cv2

NGOÀI RA: có thể dùng Python trên Microsoft Visual Studio = .NET  
 (2005, [2007]2008, 2010, [2012]2013, [2014]2015, 2017, 2019)

*Tên khác : .NET == DOT NET (thiết lập Environment Python)*

### Các thư viện lập trình chính được sử dụng trong học phần

#### Thiết kế và Lập trình giao diện: GUI

Thư viện Tkinter

#### Xử lý âm thanh và giọng nói

Thư viện speech\_recogntition, gTTS, ffpyplayer, playsound

#### Xử lý hình ảnh

Thư viện PIL,cv2

#### Xử lý Video

Thư viện cv2

### Tập các Multimedia mẫu được sử dụng trong đề tài của Học phần

. Âm thanh: <https://www.wikipedia.org/>

. Hình ảnh: <https://www.wikipedia.org/> , <https://www.flaticon.com/> ,...và sưu tầm trên internet.

. Video Clips: <https://www.youtube.com/>

Các tập multimedia mẫu chỉ dùng cho mục đích nghiên cứu.

# GIỚI THIỆU CÁC THƯ VIỆN MULTIMEDIA SỬ DỤNG

## Nạp các thư viện

1. **import** cv2 #xử lí computer vision
2. **import** os #thao tác với tài nguyên máy tính
3. **import** datetime #lấy thời gian
4. **import** speech\_recognition as sr  #Thư viện nhận dạng tiếng nói
5. **from** PIL **import** Image, ImageTk #Thư viện hình ảnh Python
6. **import** tkinter as tk#thư viện giao diện
7. **from** tkinter **import** filedialog as fd
8. **from** tkinter **import** ttk
9. **from** googletrans **import** Translator # google dịch
10. **import** googletrans
11. **import** playsound #Thư viện phát âm thanh
12. **from** gtts **import** gTTS #Thư viện Chuyển văn bản thành giọng nói
13. **import** numpy as np #Thư viện tính toán ma trận
14. **from** ffpyplayer.player **import** MediaPlayer #Mở audio của video

## Giới thiệu các thư viện được dùng

* GUI

Tkinter là thư viện GUI tiêu chuẩn cho Python.

* Audio và Speech

speech\_recogntition là thư viện nhận dạng tiếng nói.

Playsound là thư viện phát audio

Ffpyplayer là thư viện phát media

gTTS là thư viện chuyển văn bản thành giọng nói

* Images Proccesing

PIL là thư viện hình ảnh của python

Cv2 là thư viện xử lí hình ảnh,video,...

* Video Clip Proccesing

Cv2 là thư viện xử lí hình ảnh,video,...

# GIỚI THIỆU VÀ CÀI ĐẶT GIAO DIỆN GUI SẢN PHẨM

## Mô tả sản phẩm

Tên sản phẩm: **13.NguyenLeHoangThanh.DAHP.MP\_Country**

**ỨNG DỤNG QUẢN LÝ & XỬ LÝ MULTIMEDIA: LẬP TRÌNH BỞI NGUYỄN LÊ HOÀNG THANH - HỌC VIÊN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG TP.HCM**

**LẬP TRÌNH PHÁT TRIỂN HỆ THỐNG QUẢN LÝ MULTIMEDIA LIÊN QUAN ĐẾN CÁC QUỐC GIA DÙNG CV2 VỚI PYTHON**

Thực hiện bởi: Nguyễn Lê Hoàng Thanh [13] Mssv: N17DCPT058, Lớp:D17CQPU01-N

Ngôn ngữ lập trình: **PYTHON**

Công cụ: Spyder 4.1.4(Anaconda3)

Thư viện chính: **Tkinter**

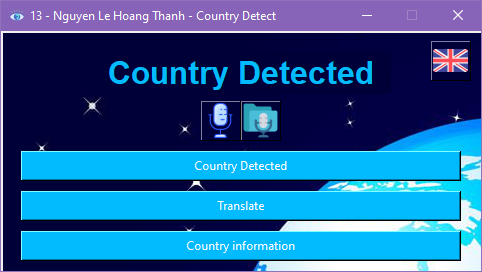
Tập dự liệu được tham khảo (thu thập) từ: https://www.flaticon.com/

## Mô tả sản phẩm

Sản phẩm giúp nhận diện và xử lí đơn giản hình ảnh, video, âm thanh, văn bản của các quốc gia cũng như giới thiệu các quốc gia.

## Giới thiệu giao diện và các chức năng





Chương trình có 3 chức năng chính:

Nhận diện quốc gia (quốc kỳ, quốc huy)

Nhận diện ngôn ngữ và dịch

Thông tin giới thiệu quốc gia(video giới thiệu, quốc ca, quốc kỳ, quốc huy)

## Thiết kế và Lập trình Winform màn hình chính

### Thiết kế (Design)

Màn hình chính



### Lập trình (Codes)

1. **class** App:#Cửa sổ chính
2. **def** \_\_init\_\_(self, root):
3. self.main\_lang=main\_lang#ngôn ngữ mặc định
4. root.title(title[self.main\_lang])#title
5. root.geometry('480x240')#kích thước
6. root.iconbitmap(res\_ico)#icon
7. root.resizable(False, False)#chặn resize cửa sổ
8. bg=tk.Label(root,image=background)#ảnh nền
9. bg.place(x=0,y=0)
10. #tên phần mềm
11. self.lbl = tk.Label(root,
12. text=tBtn\_detect[self.main\_lang],font=("Arial Bold", 24),
13. fg=color\_1,bg=color\_2,anchor="center")
14. self.lbl.place(x=90,y=20,width=300)
15. #nút input file âm thanh
16. btn\_aip=tk.Button(root,
17. image = mic\_file,borderwidth=1,
18. activebackground=color\_1,bg=color\_2,
19. command=self.btn\_auiput)
20. btn\_aip.place(x=240,y=70,width=40,height=40 )
21. #nút input bằng microphone
22. btn\_mic=tk.Button(root,
23. image = mic\_photo,borderwidth=1,
24. activebackground=color\_1,bg=color\_2,
25. command=self.btn\_mic\_clicked )
26. btn\_mic.place(x=200,y=70,width=40,height=40 )
27. #nút thay đổi ngôn ngữ
28. self.btn\_lag=tk.Button(root,
29. image = vn\_lan,borderwidth=1,
30. activebackground=color\_1,bg=color\_2,
31. command=self.btn\_lang )
32. self.btn\_lag.place(x=430,y=10,width=40,height=40 )
33. #nút chức năng nhận diện quốc kỳ, quốc huy
34. self.btn1 = tk.Button(root,
35. text=tBtn\_detect[self.main\_lang],
36. activebackground=color\_2,
37. activeforeground=color\_1,
38. bg=color\_1,fg="#fff",
39. command=self.btn1\_clicked)
40. self.btn1.place(x=20,y=120,width=440,height=30)
41. #nút chức năng dịch ngôn ngữ
42. self.btn2 = tk.Button(root,
43. text=tBtn\_trans[self.main\_lang],
44. activebackground=color\_2,
45. activeforeground=color\_1,
46. bg=color\_1,fg="#fff",
47. command=self.btn2\_clicked)
48. self.btn2.place(x=20,y=160,width=440,height=30)
49. #nút chức năng xem thông tin quốc gia
50. self.btn3 = tk.Button(root,
51. text=tBtn\_info[self.main\_lang],
52. activebackground=color\_2,
53. activeforeground=color\_1,
54. bg=color\_1,
55. fg="#fff",
56. command=self.btn3\_clicked)
57. self.btn3.place(x=20,y=200,width=440,height=30)
58. **def** btn\_lang(self):#thay đổi ngôn ngữ phần mềm
59. #đổi ngôn ngữ
60. **if** self.main\_lang==0:  #tiếng anh
61. self.main\_lang=1
62. self.btn\_lag.configure(image=en\_lan)
63. **elif** self.main\_lang==1:  #tiếng việt
64. self.main\_lang=0
65. self.btn\_lag.configure(image=vn\_lan)
66. m\_lang(self.main\_lang) #lưu ngôn ngữ đã chọn và truyền cho các chức năng khác
67. #đổi ngôn ngữ hiển thị của các nút chức năng
68. root.title(title[self.main\_lang])
69. self.btn1.configure(text=tBtn\_detect[self.main\_lang])
70. self.btn2.configure(text=tBtn\_trans[self.main\_lang])
71. self.btn3.configure(text=tBtn\_info[self.main\_lang])
72. self.lbl.configure(text=tBtn\_detect[self.main\_lang])
73. **pass**
74. **def** btn\_auiput(self):#input file âm thanh
75. nguon=select\_file()
76. **print**(nguon)
77. dulieu=talk(Data[self.main\_lang][0],sr.AudioFile(nguon)).lower()
78. #so sánh kết quả
79. **if** dulieu==tBtn\_detect[self.main\_lang]:
80. window = NhanDienQK()
81. window.grab\_set()
82. **if** dulieu==tBtn\_trans[self.main\_lang]:
83. window = NhanDienNgonNgu()
84. window.grab\_set()
85. **if** dulieu==tBtn\_info[self.main\_lang]:
86. window = ThongTinQuocGia()
87. window.grab\_set()
88. **if** dulieu=="thay đổi ngôn ngữ" **or** dulieu=="change language":
89. self.btn\_lang()
90. **pass**
91. **def** btn\_mic\_clicked(self):#input bằng microphone
92. dulieu=talk(Data[self.main\_lang][0]).lower()
93. #so sánh kết quả
94. **if** dulieu==tBtn\_detect[self.main\_lang].lower():
95. window = NhanDienQK()
96. window.grab\_set()
97. **if** dulieu==tBtn\_trans[self.main\_lang].lower():
98. window = NhanDienNgonNgu()
99. window.grab\_set()
100. **if** dulieu==tBtn\_info[self.main\_lang].lower():
101. window = ThongTinQuocGia()
102. window.grab\_set()
103. **if** dulieu=="thay đổi ngôn ngữ" **or** dulieu=="change language":
104. self.btn\_lang()
105. **pass**
106. **def** btn1\_clicked(self):#chức năng nhận diện quốc kỳ, quốc huy
107. window = NhanDienQK()
108. window.grab\_set()
109. **pass**
110. **def** btn2\_clicked(self):#chức năng dịch ngôn ngữ
111. window=NhanDienNgonNgu()
112. window.grab\_set()
113. **pass**
114. **def** btn3\_clicked(self):#chức năng xem thông tin
115. window=ThongTinQuocGia()
116. window.grab\_set()
117. **pass**

# LẬP TRÌNH XỬ LÝ ÂM THANH VÀ GIỌNG NÓI

## Giới thiệu

. Các thư viện

speech\_recognition: Thư viện chuyển âm thành thành văn bản

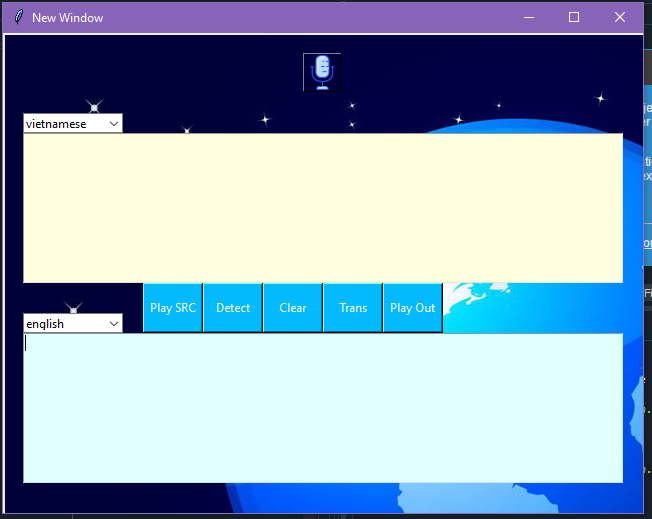
gTTS:Thư viện chuyển văn bản thành âm thanh

playsound: Thư viện phát âm thanh

googletrans:Thư viện dịch thuật

ffpyplayer: thư viện phát media

## Thiết kế giao diện và xử lý cơ bản



Code xử lí giao diện

1. **class** NhanDienNgonNgu(tk.Toplevel):
2. **def** \_\_init\_\_(self, master = None):
3. super().\_\_init\_\_(master = master)
4. self.main\_lang=m\_lang.variable#ngôn ngữ
5. self.title(title[self.main\_lang])#title
6. self.geometry("{}x{}".format(height,widht))#kích thước
7. self.iconbitmap(res\_ico)#icon
8. self.resizable(False, False)#chặn resize cửa sổ
9. bg=tk.Label(self,image=background)#ảnh nền
10. bg.place(x=0,y=0)
11. #microphone
12. btn\_mic=tk.Button(self,
13. image = mic\_photo,
14. borderwidth=1,
15. activebackground=color\_1,
16. bg=color\_2,
17. command=self.ghiAm,
18. )
19. btn\_mic.pack(pady=20)
20. #COmbobox ngôn ngữ nguồn và ngôn ngữ đích
21. Combo\_src = ttk.Combobox(self, values = d1)
22. Combo\_src.current(101)
23. Combo\_src.place(x=20,y=80,width=100,height=20)
24. self.Combo\_src=Combo\_src
25. Combo = ttk.Combobox(self, values = d1)
26. Combo.current(21)
27. Combo.place(x=20,y=280,width=100,height=20)
28. self.Combo=Combo
29. #text box
30. gInput = tk.Text(self,
31. bg = "light yellow")
32. gInput.place(x=20,y=100,width=600,height=150)
33. self.gInput=gInput
34. #button
35. #Đọc nguồn
36. btn\_p1=tk.Button(self,
37. text="Play SRC",
38. activebackground=color\_2,
39. activeforeground=color\_1,
40. bg=color\_1,
41. fg="#fff",
42. command=self.play1,
43. )
44. btn\_p1.place(x=140,y=250,width=60,height=50)
45. #Nhận diện ngôn ngữ nguồn
46. btn\_det=tk.Button(self,
47. text="Detect",
48. activebackground=color\_2,
49. activeforeground=color\_1,
50. bg=color\_1,
51. fg="#fff",
52. command=self.det,
53. )
54. btn\_det.place(x=200,y=250,width=60,height=50)
55. #Xóa dữ liệu nhập
56. btn\_cle=tk.Button(self,
57. text="Clear",
58. activebackground=color\_2,
59. activeforeground=color\_1,
60. bg=color\_1,
61. fg="#fff",
62. command=self.cle,
63. )
64. btn\_cle.place(x=260,y=250,width=60,height=50)
65. #Dịch
66. btn\_tras=tk.Button(self,
67. text="Trans",
68. activebackground=color\_2,
69. activeforeground=color\_1,
70. bg=color\_1,
71. fg="#fff",
72. command=self.trans,
73. )
74. btn\_tras.place(x=320,y=250,width=60,height=50)
75. #Đọc kết quả
76. btn\_p2=tk.Button(self,
77. text="Play Out",
78. activebackground=color\_2,
79. activeforeground=color\_1,
80. bg=color\_1,
81. fg="#fff",
82. command=self.play2,
83. )
84. btn\_p2.place(x=380,y=250,width=60,height=50)
85. #text hiển thị kết quả
86. gOutput = tk.Text(self,
87. bg = "light cyan")
88. gOutput.place(x=20,y=300,width=600,height=150)
89. self.gOutput=gOutput

## Lập trình xử lý âm thanh

1. **def** play1(self):#Đọc nguồn
2. text=self.gInput.get(1.0,tk.END)
3. lang=d2[self.Combo\_src.current()]
4. speak(text,lang)
5. **def** play2(self):#Đọc kết quả
6. text=self.gOutput.get(1.0,tk.END)
7. lang=d2[self.Combo.current()]
8. speak(text,lang)
9. **def** det(self):#Nhận diện ngôn ngữ
10. text=self.gInput.get(1.0,tk.END)
11. translator = Translator()
12. lang=translator.detect(text).lang
13. **for** i **in** range(len(d2)):
14. **if** d2[i] == lang:
15. **print**(i)
16. self.Combo\_src.current(i)
17. **def** cle(self):#Xóa dữ liệu nhập
18. self.gInput.delete(1.0,tk.END)
19. self.gOutput.delete(1.0,tk.END)
20. **def** trans(self):#Dịch
21. self.gOutput.delete(1.0,tk.END)
22. text=self.gInput.get(1.0,tk.END)
23. i=self.Combo\_src.current()
24. o=self.Combo.current()
25. translator = Translator()
26. translated = translator.translate(text,dest=d2[o],src=d2[i])
27. self.gOutput.insert(tk.END,translated.text)
28. **pass**
29. **def** ghiAm(self):#Microphone
30. lang=d2[self.Combo\_src.current()]#Lấy ngôn ngữ input đang chọn
31. dulieu=talk(lang)
32. dulieu=dulieu+" "
33. self.gInput.insert(tk.END,dulieu)
34. **pass**

## Lập trình Xử lý giọng nói

1. #Chuyển văn bản thành giọng nói
2. **def** speak(text,lang):
3. tts = gTTS(text=text, lang=lang) #chuyển văn bản thành giọng nói
4. audiofile = '13NLHT\_Audio/13NguyenLeHoangThanh.mp3'
5. **if** os.path.exists(audiofile):
6. os.remove(audiofile)
7. **else**:
8. **print**(audiofile)
9. tts.save(audiofile)#lưu file âm thanh
10. **try**:
11. playsound.playsound(audiofile)#phát file âm thanh
12. **except**:
13. **pass**
15. #Nghe người dùng nói
16. **def** talk(lang="vi",sour=sr.Microphone()):#Nhận diện giọng nói nhập dữ liệu
17. query=''
18. r = sr.Recognizer()
19. with sour as source:
20. **print**("Điều chỉnh tiếng ồn  ")
21. r.adjust\_for\_ambient\_noise(source, duration=1)
22. **print**("Nói...")
23. **try**:
24. audio\_data = r.listen(source)
25. **except**:
26. audio\_data = r.record(source)
27. **print**("Kết quả...")
28. **try**:
29. query = r.recognize\_google(audio\_data,language=lang)
30. **except**:
31. **print**("Lỗi")
32. **print**(str(query))
33. **return** str(query)

## Ứng dụng của sản phẩm

13.NguyenLeHoangThanh.B2Ex1SpeechProcessing

1. **import** speech\_recognition as sr #thư viện nhận diện giọng nói
2. **from** gtts **import** gTTS #Thư viện Chuyển văn bản thành giọng nói
3. **import** playsound #Thư viện phát âm thanh
5. r = sr.Recognizer()
6. with sr.Microphone() as source: #sử dụng micrô mặc định làm nguồn(source ):
7. **print**("Điều chỉnh tiếng ồn ")
8. r.adjust\_for\_ambient\_noise(source, duration=1)#lắng nghe trong 1 giây để hiệu chỉnh độ ồn
9. **print**("Nói bằng tiếng Việt, 5s sau sẽ in ra Text")
10. audio\_data = r.record(source, duration=5) #nghe trong 5s
11. **print**("Kết quả nhận diện...")
12. **try**:
13. text = r.recognize\_google(audio\_data,language="vi") #Nhận diện âm thanh tiếng Việt
14. **except**:
15. text = "bạn nói gì mình không hiểu!"
16. **print**("Bạn đã nói là: {}".format(text))
18. **def** speak(text):
19. tts = gTTS(text=text, lang='vi') #chuyển văn bản thành giọng nói(tiếng Việt )
20. tenfile = '13NguyenLeHoangThanh.mp3'
21. tts.save(tenfile)#lưu file âm thanh
22. playsound.playsound(tenfile)#phát file âm thanh
24. speak("Nguyễn Lê Hoàng Thanh ")

# LẬP TRÌNH XỬ LÝ và NHẬN DIỆN ẢNH

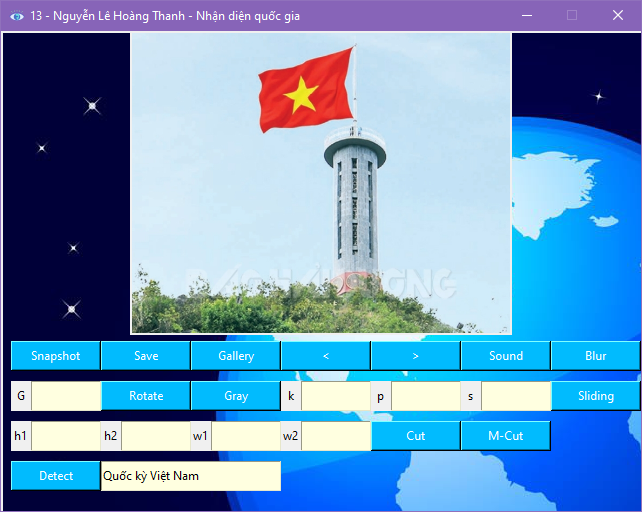
## Giới thiệu

. Các thư viện

Opencv: Thư viện Computer Vision xử lý hình ảnh

PIL: Python Imaging Library hỗ trợ thêm việc mở, thao tác và lưu nhiều định dạng tệp hình ảnh khác nhau.

## Thiết kế giao diện và xử lý cơ bản



1. **class** NhanDienQuocKy(tk.Toplevel):#cửa sổ chức năng nhận diện
2. **def** \_\_init\_\_(self, master = None):
3. super().\_\_init\_\_(master = master)
4. self.main\_lang=m\_lang.variable#ngôn ngữ
5. self.title(title[self.main\_lang])#title
6. self.geometry("{}x{}".format(height,widht))#kích thước
7. self.iconbitmap(res\_ico)#icon
8. self.resizable(False, False)#chặn resize cửa sổ
9. bg=tk.Label(self,image=background)#nền
10. bg.place(x=0,y=0)
11. self.lmain = tk.Label(self)#vị trí phát video,ảnh
12. self.lmain.pack()
13. self.frame=None
15. self.img\_link=func.variable
16. #Chụp ảnh
17. btn\_snapshot  = tk.Button(self,
18. text="Snapshot",
19. activebackground=color\_2,
20. activeforeground=color\_1,
21. bg=color\_1,
22. fg="#fff",
23. command=self.snapshot)
24. btn\_snapshot.place(x=10,y=310,width=90,height=30)
25. #Lưu
26. btn\_save  = tk.Button(self,text="Save",bg=color\_1,fg="#fff",
27. activebackground=color\_2,
28. activeforeground=color\_1,
29. command=self.save)
30. btn\_save.place(x=100,y=310,width=90,height=30)
31. #Mở ảnh đã lưu
32. btn\_openfolder  = tk.Button(self,text="Gallery",bg=color\_1,fg="#fff",
33. activebackground=color\_2,
34. activeforeground=color\_1,
35. command=self.ofolder)
36. btn\_openfolder.place(x=190,y=310,width=90,height=30)
37. #tiến, lùi ảnh trong folder
38. btn\_tien  = tk.Button(self,text=">",bg=color\_1,fg="#fff",
39. activebackground=color\_2,
40. activeforeground=color\_1,
41. command=self.tien)
42. btn\_tien.place(x=370,y=310,width=90,height=30)
43. btn\_lui  = tk.Button(self,text="<",bg=color\_1,fg="#fff",
44. activebackground=color\_2,
45. activeforeground=color\_1,
46. command=self.lui)
47. btn\_lui.place(x=280,y=310,width=90,height=30)
48. #Bật tắt âm thanh video
49. btn\_amthanh  = tk.Button(self,text="Sound",bg=color\_1,fg="#fff",
50. activebackground=color\_2,
51. activeforeground=color\_1,
52. command=self.sound
53. )
54. btn\_amthanh.place(x=460,y=310,width=90,height=30)
55. #Xoay ảnh
56. self.rote\_data=0
57. lb\_xoay = tk.Label( self, text="G")
59. lb\_xoay.place(x=10,y=350,width=20,height=30)
60. self.tb\_xoay=tk.Entry(self,
61. bg = "light yellow",)
62. self.tb\_xoay.place(x=30,y=350,width=70,height=30)
63. btn\_xoay  = tk.Button(self,text="Rotate",bg=color\_1,fg="#fff",
64. activebackground=color\_2,
65. activeforeground=color\_1,
66. command=self.rote)
67. btn\_xoay.place(x=100,y=350,width=90,height=30)
68. #Ảnh xám
69. self.gray=True
70. btn\_xam  = tk.Button(self,text="Gray",bg=color\_1,fg="#fff",
71. activebackground=color\_2,
72. activeforeground=color\_1,
73. command=self.toGray)
74. btn\_xam.place(x=190,y=350,width=90,height=30)
75. #sliding window
76. self.kernel=[[1]]
77. self.padding=0
78. self.stride=1
79. self.checkSlideWin=True
80. lb\_k = tk.Label( self, text="k")
81. lb\_k.place(x=280,y=350,width=20,height=30)
82. self.tb\_k=tk.Entry(self,
83. bg = "light yellow",)
84. self.tb\_k.place(x=300,y=350,width=70,height=30)
85. lb\_p = tk.Label( self, text="p")
86. lb\_p.place(x=370,y=350,width=20,height=30)
87. self.tb\_p=tk.Entry(self,
88. bg = "light yellow",)
89. self.tb\_p.place(x=390,y=350,width=70,height=30)
90. lb\_s = tk.Label( self, text="s")
91. lb\_s.place(x=460,y=350,width=20,height=30)
92. self.tb\_s=tk.Entry(self,
93. bg = "light yellow",)
94. self.tb\_s.place(x=480,y=350,width=70,height=30)
95. btn\_sw  = tk.Button(self,text="Sliding",bg=color\_1,fg="#fff",
96. activebackground=color\_2,
97. activeforeground=color\_1,
98. command=self.sWin)
99. btn\_sw.place(x=550,y=350,width=90,height=30)
100. #blur
101. self.checkBlur=True
102. btn\_blur  = tk.Button(self,text="Blur",bg=color\_1,fg="#fff",
103. activebackground=color\_2,
104. activeforeground=color\_1,
105. command=self.blurimg)
106. btn\_blur.place(x=550,y=310,width=90,height=30)
107. #cắt ảnh
108. self.h1=0
109. self.h2=0
110. self.w1=0
111. self.w2=0
112. self.checkcut=True
113. lb\_h1 = tk.Label( self, text="h1")
114. lb\_h1.place(x=10,y=390,width=20,height=30)
115. self.tb\_h1=tk.Entry(self,
116. bg = "light yellow",)
117. self.tb\_h1.place(x=30,y=390,width=70,height=30)
118. lb\_h2 = tk.Label( self, text="h2")
119. lb\_h2.place(x=100,y=390,width=20,height=30)
120. self.tb\_h2=tk.Entry(self,
121. bg = "light yellow",)
122. self.tb\_h2.place(x=120,y=390,width=70,height=30)
123. lb\_w1 = tk.Label( self, text="w1")
124. lb\_w1.place(x=190,y=390,width=20,height=30)
125. self.tb\_w1=tk.Entry(self,
126. bg = "light yellow",)
127. self.tb\_w1.place(x=210,y=390,width=70,height=30)
128. lb\_w2 = tk.Label( self, text="w2")
129. lb\_w2.place(x=280,y=390,width=20,height=30)
130. self.tb\_w2=tk.Entry(self,
131. bg = "light yellow",)
132. self.tb\_w2.place(x=300,y=390,width=70,height=30)
133. btn\_cut  = tk.Button(self,text="Cut",bg=color\_1,fg="#fff",
134. activebackground=color\_2,
135. activeforeground=color\_1,
136. command=self.cut)
137. btn\_cut.place(x=370,y=390,width=90,height=30)
138. #cắt ảnh bằng chuột
139. self.cropping=False
140. btn\_cut  = tk.Button(self,text="M-Cut",bg=color\_1,fg="#fff",
141. activebackground=color\_2,
142. activeforeground=color\_1,
143. command=self.cutm)
144. btn\_cut.place(x=460,y=390,width=90,height=30)
145. ####Nhận diện ảnh
146. self.lb = tk.Entry(self,
147. bg = "light yellow",)
148. self.lb.place(x=100,y=430,width=180,height=30)
149. btn\_cut  = tk.Button(self,text="Detect",bg=color\_1,fg="#fff",
150. activebackground=color\_2,
151. activeforeground=color\_1,
152. command=self.detect\_img)
153. btn\_cut.place(x=10,y=430,width=90,height=30)
154. ###
155. #mở
156. self.open\_f()
157. #đóng
158. self.protocol("WM\_DELETE\_WINDOW", self.on\_closing)

## Lập trình xử lý ảnh và khung ảnh

1. **def** resetLB(self):#reset lại các thuộc tính label phát video
2. self.lmain.destroy()
3. self.lmain = tk.Label(self)
4. self.lmain.pack()
5. **def** cutm(self):#Cắt ảnh bằng chuột
6. **def** mouse\_crop(event, x, y, flags, param):#Cắt ảnh
7. self.w1, self.h1, self.w2, self.h2#tọa độ các điểm cắt
8. **if** event == cv2.EVENT\_LBUTTONDOWN:#Khi nhấn chuột trái thì lấy tọa độ w1,h1
9. self.w1, self.h1, self.w2, self.h2 = x, y, x, y
10. self.cropping = True
11. **elif** event == cv2.EVENT\_MOUSEMOVE:#Khi di chuyển chuột lấy tọa độ w2, h2
12. **if** self.cropping == True:
13. self.w2, self.h2 = x, y
14. **elif** event == cv2.EVENT\_LBUTTONUP:#Thả chuột ra lưu tọa độ w2,h2 và cho phép cắt
15. self.w2, self.h2 = x, y
16. self.cropping = False
17. self.checkcut=False
18. **print**(self.h1, self.w1, self.h2, self.w2)
20. **pass**
22. img=cv2.cvtColor( self.img\_out.copy(),cv2.COLOR\_BGR2RGBA)
23. cv2.imshow("a",img)#hiện cửa sổ ảnh
24. cv2.setMouseCallback("a", mouse\_crop)# event bấm chuột
25. **while** True:#Vẽ khung cắt và cắt ảnh
26. i = img.copy()
27. **if** **not** self.cropping:
28. cv2.imshow("a", img)
29. **elif** self.cropping:
30. cv2.rectangle(i, (self.w1, self.h1), (self.w2, self.h2), (255, 0, 0), 2)#Khung ảnh cắt
31. cv2.imshow("a", i)
32. **if** self.checkcut==False:
33. self.open\_f()
34. **break**
35. cv2.waitKey(1)
36. cv2.destroyAllWindows()
37. self.checkcut=True
38. **pass**
39. **def** open\_f(self):
40. self.resetLB()
41. #Mo anh hoac video
42. **try**:
43. self.open\_img()
44. **except**:
45. self.open\_vid()
46. **pass**
47. **def** save(self):#Lưu file đã xử lí
48. ts = datetime.datetime.now()
49. filename="13NLHT\_Save/{}.jpg".format(ts.strftime("%Y-%m-%d\_%H-%M-%S"))
50. img=cv2.cvtColor( self.img\_out.copy(),cv2.COLOR\_BGR2RGBA)
51. cv2.imwrite(os.path.join(filename),img)
52. **pass**
53. **def** ofolder(self):#Mở file dữ liệu lưu ở thư mục 13NLHT\_Img
54. files= []
55. self.classNames  = []
56. self.temp=0
57. myList = os.listdir(data\_adress)#lấy danh sách tệp tin
58. #Lấy ảnh và tên ảnh
59. **for** cl **in** myList:
60. file= cv2.imread(f'{data\_adress}/{cl}',0) #đọc file ở đường dẫn
61. files.append(file)#thêm vào files
62. self.classNames.append(os.path.basename(cl))#thêm tên vào className
63. self.classNames.reverse()#đảo mảng
64. self.img\_link=data\_adress+"/"+self.classNames[0]#Ảnh lưu gần nhất
65. self.cap.release()
66. cv2.destroyAllWindows()
67. **try**:
68. self.player.close\_player()
69. **except**:
70. **pass**
71. self.open\_f()
72. **pass**
73. **def** detect\_img(self):#Nhận diện hình ảnh
74. desList = findDes(images) #trích đặc trưng dữ liệu
75. id= findID(self.img\_out,desList)#trích đặc trưng và nhận diện ảnh
76. fn=""
77. **if** id==-1:
78. **if** self.main\_lang==0:
79. fn="Không xác định được!"
80. **else**:
81. fn="Error: Unknown!"
82. **else**:
83. **print**(id)
84. x=divmod(id, 2)#vì mảng = quốc kỳ+quốc huy+...
85. **print**(Data[x[0]][self.main\_lang+1])
86. **if** x[1]==0:
87. **if** self.main\_lang==0:
88. y="Quốc kỳ"
89. **else**:
90. y="Ensign:"
91. **elif** x[1]==1:
92. **if** self.main\_lang==0:
93. y="Quốc huy"
94. **else**:
95. y="Emblem: "
96. **pass**
97. fn=y+" "+Data[x[0]][self.main\_lang+1]
98. self.lb.delete(0,tk.END)
99. self.lb.insert(tk.END, fn)
100. **def** tien(self):#lấy ảnh tiếp theo
101. **if** self.temp == len(self.classNames):
102. **return**
103. **else**:
104. self.rote\_data=0
105. self.gray=True
106. self.checkSlideWin=True
107. self.checkcut=True
108. self.checkBlur=True
109. self.temp+=1
110. self.img\_link=data\_adress+"/"+self.classNames[self.temp]
111. self.open\_f()
112. **pass**
113. **def** lui(self):#lấy ảnh phía trước
114. **if** self.temp == 0:
115. **return**
116. **else**:
117. self.rote\_data=0
118. self.gray=True
119. self.checkSlideWin=True
120. self.checkcut=True
121. self.checkBlur=True
122. self.temp-=1
123. self.img\_link=data\_adress+"/"+self.classNames[self.temp]
124. self.open\_f()
125. **pass**
126. **def** cut(self):#cắt bằng tọa độ
127. self.checkcut=False
128. self.h1=int(self.tb\_h1.get())
129. self.h2=int(self.tb\_h2.get())
130. self.w1=int(self.tb\_w1.get())
131. self.w2=int(self.tb\_w2.get())
132. self.open\_f()
133. **pass**
134. **def** rote(self):#xoay
135. i=self.tb\_xoay.get()
136. **if** i=='':
137. **return**
138. self.rote\_data=int(i)
139. self.open\_f()
140. **pass**
141. **def** toGray(self):#chuyển xám
142. **if** self.gray:
143. self.gray=False
144. **elif** **not** self.gray:
145. self.gray=True
146. self.open\_f()
147. **pass**
148. **def** blurimg(self):#làm mờ
149. **if** self.checkBlur:
150. self.checkBlur=False
151. **else**:
152. self.checkBlur=True
153. self.open\_f()
154. **pass**
155. **def** sWin(self):#sliding window
156. **if** self.checkSlideWin:
157. self.checkSlideWin=False
158. **elif** **not** self.checkSlideWin:
159. self.kernel=[[float(self.tb\_k.get())]]
160. self.padding=int(self.tb\_p.get())
161. self.stride=int(self.tb\_s.get())
162. self.checkSlideWin=True
163. self.open\_f()
164. **pass**
165. **def** open\_img(self):#Mở ảnh và xử lý
166. im=cv2.imread(self.img\_link)
167. **if** **not** self.checkSlideWin:#sliding window
168. im=apply\_sliding\_window\_on\_3\_channels(im, self.kernel,self.padding,self.stride)
169. im=(im \* 255).round().astype(np.uint8)#chuyển im về kiểu thích hợp để xử lí tiếp
170. cv2image = cv2.cvtColor(im, cv2.COLOR\_BGR2RGBA)
171. cv2image=resize(cv2image)#resize
172. cv2image=rotate(cv2image, self.rote\_data)#xoay
173. **if** self.checkcut==False:#cắt
174. cv2image=cutimg(cv2image,self.h1,self.h2,self.w1,self.w2)
175. **if** self.gray==False: #xám
176. cv2image=rgb\_to\_gray(cv2image )
177. **if** **not** self.checkBlur:#mờ
178. cv2image=cv2.blur(cv2image,(5,5))
179. img = Image.fromarray(cv2image)
180. imgtk = ImageTk.PhotoImage(image=img)
181. self.img\_out=cv2image#Truyền ảnh kết quả sang các chức năng khác
182. #Cập nhật ảnh
183. self.lmain.imgtk = imgtk
184. self.lmain.configure(image=imgtk)
185. **pass**
186. **def** on\_closing(self):#Xử lí khi thoát chức năng
187. **try**:
188. self.cap.release()
189. cv2.destroyAllWindows()
190. self.player.close\_player()
191. **except**:
192. **pass**
193. self.destroy()
194. **pass**

## Lập trình xử lý nhận diện ảnh (tĩnh)

1. #Nhận diện ảnh với hàm sift của opencv
2. sift = cv2.SIFT\_create()
3. #Trích xuất đặc trưng của dữ liệu mẫu
4. **def** findDes(images):
5. desList=[]
6. **for** img **in** images:
7. img=resize(img)
8. img=cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)
9. kp, des = sift.detectAndCompute(img, None)#trích xuất đặc trưng
10. desList.append(des)#lưu đặc trưng
11. **return** desList
12. #Trích xuất đặc trưng của ảnh cần xử lí và so sánh, tìm ảnh mẫu giống nhất
13. **def** findID(img, desList,thres=15 ):
14. img=resize(img)
15. img=cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)
16. kp2, des2 = sift.detectAndCompute(img, None)#trích xuất đặc trưng
17. bf = cv2.BFMatcher()
19. matchList=[]
20. finalVal =-1
21. **try**:
22. #kiểm tra kết quả phù hợp bằng cách kiểm tra của Lowe
23. **for** des **in** desList:
24. matches = bf.knnMatch(des,des2,k=2)#tìm 2 bộ giống nhất của điểm đặc trưng thuộc des trong des2
25. good=[]
26. **for** m,n **in** matches:
27. **if** m.distance <0.75 \*n.distance:#0.75 là tỉ lệ theo lowe ,distance là khoảng cách giữa các điểm
28. good.append([m])#nếu khoảng cách m,n quá gần thì chúng không đủ khác biệt nên loại bỏ và ngược lại
29. matchList.append(len(good))#số điểm đặc trưng phù hợp điều kiện
30. **except**:
31. **pass**
32. #print(max(matchList))
33. # kiểm tra số điểm phù hợp có đủ nhiều để xác định không
34. **if** len(matchList)!=0:
35. **if** max(matchList) > thres:
36. finalVal= matchList.index(max(matchList))
37. **return**  finalVal
38. #Lưu giá trị biến
39. **def** func(var):
40. func.variable = var
42. **def** fc(var):
43. fc.variable = var
45. **def** m\_lang(var):
46. m\_lang.variable = var
47. m\_lang(main\_lang)#Hàm lưu biến ngôn ngữ mặc định
48. # thay đổi kích thước ảnh theo tỉ lệ với chiều cao (mặc định 300 pixel )
49. **def** resize(img,size\_h=300):
50. (h, w, d) = img.shape# kích thước ảnh màu
51. s=size\_h/h
52. w=int(w\*s)
53. h=int(h\*s)
54. dim=(w,h)#kích thước mới
55. resized = cv2.resize(img, dim)#thay đổi kích thước
56. **return** resized
57. #Chọn file
58. **def** select\_file():
59. filetypes = (
60. ('All files', '\*.\*'),
61. ('video files', '\*.mp4;\*.mov'),
62. ('image files', '\*.jpg;\*.png')
63. )
65. filename = fd.askopenfilename(
66. title='Open a file',
67. initialdir='/',
68. filetypes=filetypes)
69. **return** filename#trả về đường dẫn
70. # xoay ảnh (img: hình ảnh, g : góc quay)
71. **def** rotate(img,g):
72. (h, w,d) = img.shape # kích thước ảnh
73. center = (w // 2, h // 2)#vị trị giữa
74. M = cv2.getRotationMatrix2D(center, g, 1.0) #ma trận biến đổi theo góc g
75. rotated = cv2.warpAffine(img, M, (w, h))#xoay ảnh
76. **return** rotated #trả về kết quả
77. #cắt ảnh theo tọa độ
78. **def** cutimg(img,h1,h2,w1,w2):
79. p= img[h1:h2,w1:w2]#cắt ảnh
80. **return** p
81. # chuyển ảnh xám theo The weighted method
82. **def** rgb\_to\_gray(img):
83. grayImage = np.zeros(img.shape) #tạo ảnh rỗng
84. #lấy ma trận màu
85. R = np.array(img[:, :, 0])
86. G = np.array(img[:, :, 1])
87. B = np.array(img[:, :, 2])
88. #Lấy giá trị màu xám theo The weighted method
89. R = (R \*.299)
90. G = (G \*.587)
91. B = (B \*.114)
92. Avg = (R+G+B)
93. grayImage = img.copy()
94. #chuyển thành ảnh xám
95. **for** i **in** range(3):
96. grayImage[:,:,i] = Avg
97. **return** grayImage
98. #Sliding window
99. **def** apply\_sliding\_window(img, kernel, padding=0, stride=1):
100. h, w = img.shape[:2]
101. img\_p = np.zeros([h+2\*padding, w+2\*padding])
102. img\_p[padding:padding+h, padding:padding+w] = img
103. kernel = np.array(kernel) # lập cửa sổ trượt
104. **assert** len(kernel.shape) == 2 **and** kernel.shape[0] == kernel.shape[1]
105. **assert** kernel.shape[0] % 2 != 0
107. k\_size = kernel.shape[0]
108. k\_half = int(k\_size/2)
110. y\_pos = [v **for** idx, v **in** enumerate(list(range(k\_half, h-k\_half))) **if** idx % stride == 0]
111. x\_pos = [v **for** idx, v **in** enumerate(list(range(k\_half, w-k\_half))) **if** idx % stride == 0]
113. new\_img = np.zeros([len(y\_pos), len(x\_pos)])
114. **for** new\_y, y **in** enumerate(y\_pos):
115. **for** new\_x, x **in** enumerate(x\_pos):
116. **if** k\_half == 0:
117. pixel\_val = img\_p[y, x] \* kernel
118. **else**:
119. pixel\_val = np.sum(img\_p[y-k\_half:y-k\_half+k\_size, x-k\_half:x-k\_half+k\_size] \* kernel)
120. new\_img[new\_y, new\_x] = pixel\_val
121. **return** new\_img
122. **def** apply\_sliding\_window\_on\_3\_channels(img, kernel, padding=0, stride=1):   #làm mờ ảnh
123. layer\_blue = apply\_sliding\_window(img[:,:,0], kernel, padding, stride)
124. layer\_green = apply\_sliding\_window(img[:,:,1], kernel, padding, stride)
125. layer\_red = apply\_sliding\_window(img[:,:,2], kernel, padding, stride)
126. new\_img = np.zeros(list(layer\_blue.shape) + [3])
127. new\_img[:,:,0], new\_img[:,:,1], new\_img[:,:,2] = layer\_blue, layer\_green, layer\_red
128. **return** new\_img

## Ứng dụng của sản phẩm

13.NguyenLeHoangThanh.B3Ex2\_ImagesPro\_opencv

1. **import** cv2 # opencv xử lý hình ảnh
2. **import** os #tương tác với hệ điều hành
4. **def** menu():#menu đầu tiên
5. **print**("Nhập lựa chọn : ")
6. **print**("1.Nạp ảnh màu ")
7. **print**("2.Nạp ảnh xám ")
8. ch  = input("Chọn ct :  ")
9. **return** ch
10. **def** iput():#nhập đường dẫn
11. path=input("Nhập đường dẫn file ảnh: ")
12. #kiểm tra file có tồn tại hay không
13. **if** os.path.exists(path):
14. **print**("exists.")
15. **else**:
16. **print**("not exists.")
17. iput()
18. **return** path
19. **def** submenu1():#Menu nạp ảnh màu
20. **print**("""
21. Chọn chức năng:
22. 1.1.Chuyển ảnh xám
23. 1.2.Lấy kích thước ảnh
24. 1.3.Cắt ảnh theo thông tin về kích thước ảnh được cắt do NSD nhập vào
25. 1.4.Xoay ảnh theo yêu cầu của NSD (NSD nhập yêu câu)
26. 1.5.Lấy giá trị màu của điểm ảnh tại vị trí NSD chọn với hệ màu RGB
27. 1.6.Thay đổi kích thước ảnh theo yêu cầu của NSD
28. """)
29. ch  = input("Chọn chức năng :  ")
30. **return** ch
31. **def** submenu2():#Menu nạp ảnh xám
32. **print**("""
33. Chọn chức năng :
34. 2.1.Lấy kích thước ảnh
35. 2.2.Cắt ảnh theo thông tin về kích thước ảnh được cắt do NSD nhập vào
36. 2.3.Xoay ảnh theo yêu cầu của NSD (NSD nhập yêu câu)
37. 2.4.Lấy giá trị màu của điểm ảnh tại vị trí NSD chọn(màu xám)
38. """)
39. ch  = input("Chọn chức năng :  ")
40. **return** ch
41. **def** imgsize(img,mau):#kích thước ảnh
42. **if** mau==1:# kích thước ảnh màu
43. (h, w, d) = img.shape
44. **print**("width={}, height={}, depth={}".format(w, h, d))
45. **if** mau==0:#kích thước ảnh xám
46. (h, w) = img.shape
47. **print**("width={}, height={}".format(w, h))
48. **def** cutimg(img):#cắt ảnh
49. h1=int(input("Cắt ảnh từ chiều cao:"))
50. h2=int(input(" đến "))
51. w1=int(input("Cắt ảnh từ chiều rộng :"))
52. w2=int(input(" đến "))
53. p= img[h1:h2,w1:w2]
54. cv2.imshow('Cat',p)
55. **def** rotate(img,mau):#xoay ảnh
56. **if** mau == 1:# lấy kích thước ảnh màu
57. (h, w, d) = img.shape
58. **if** mau == 0:# lấy kích thước ảnh xám
59. (h, w) = img.shape
60. center = (w // 2, h // 2)# lấy điểm giữa ảnh
61. g=float(input("Góc xoay theo độ(>0 quay ngược kim đồng hồ ):"))
62. M = cv2.getRotationMatrix2D(center, g, 1.0) # ma trận biến đổi theo góc g
63. rotated = cv2.warpAffine(img, M, (w, h))#xoay ảnh theo ma trận
64. cv2.imshow('Xoay',rotated)
65. **def** getcolor(img,mau):# lấy màu tại 1 điểm
66. x=int(input("Giá trị tọa độ x của điểm: "))
67. y=int(input("Giá trị tọa độ y của điểm: "))
68. **if** mau==1:# lấy màu ảnh màu
69. (B, G, R) = img[x,y]
70. **print**("Blue={},Green={},Red={}".format(B,G,R))
71. **if** mau==0:# lấy màu ảnh xám
72. **print**("Gray:",img[x,y])
73. **def** resize(img):# thay đổi kích thước ảnh
74. (h, w, d) = img.shape# kích thước ảnh màu
75. imgsize(img, 1)
76. s=int(input("Tỉ lệ ảnh: "))
77. w=int(w\*s/100)
78. h=int(h\*s/100)
79. dim=(w,h)#kích thước mới
80. resized = cv2.resize(img, dim)#thay đổi kích thước
81. imgsize(resized,1)
82. cv2.imshow('TiLe',resized)
83. **while** True:
84. ch=menu()
85. **if** ch=="1":
86. path=iput()
87. img = cv2.imread(path)
88. ch1=submenu1()
89. **if** ch1=="1":
90. imgGray = cv2.imread(path,cv2.IMREAD\_GRAYSCALE)# đọc ảnh xám
91. cv2.imshow('Xam',imgGray)
92. cv2.imshow('Goc',img)
93. cv2.waitKey(0)
94. cv2.destroyAllWindows()
95. **if** ch1=="2":
96. imgsize(img,1)
97. cv2.waitKey(0)
98. cv2.destroyAllWindows()
99. **if** ch1=="3":
100. cutimg(img)
101. cv2.waitKey(0)
102. cv2.destroyAllWindows()
103. **if** ch1=="4":
104. rotate(img,1)
105. cv2.waitKey(0)
106. cv2.destroyAllWindows()
107. **if** ch1=="5":
108. getcolor(img, 1)
109. **if** ch1=="6":
110. resize(img)
111. cv2.waitKey(0)
112. cv2.destroyAllWindows()
113. **if** ch=="2":
114. path=iput()
115. img= cv2.imread(path,cv2.IMREAD\_GRAYSCALE)# đọc ảnh xám
116. ch2=submenu2()
117. **if** ch2=="1":
118. imgsize(img,0)
119. **if** ch2=="2":
120. cutimg(img)
121. cv2.waitKey(0)
122. cv2.destroyAllWindows()
123. **if** ch2=="3":
124. rotate(img,0)
125. cv2.waitKey(0)
126. cv2.destroyAllWindows()
127. **if** ch2=="4":
128. getcolor(img, 0)
129. cv2.waitKey(0)
130. cv2.destroyAllWindows()

# LẬP TRÌNH XỬ LÝ VIDEO VÀ NHẬN DIỆN LIÊN QUAN

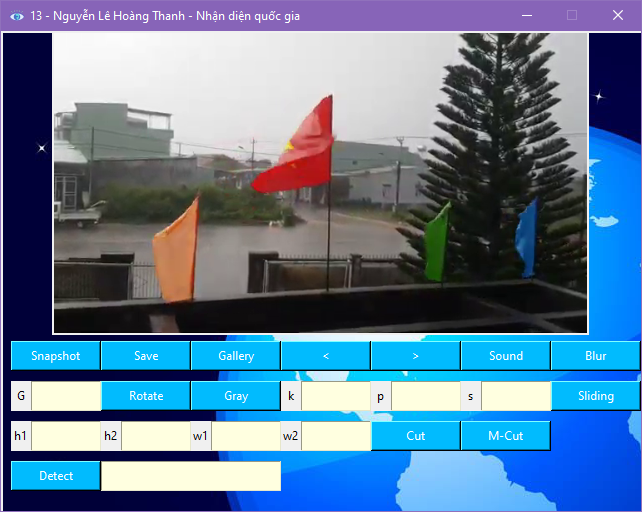
## Giới thiệu

. Các thư viện

Opencv :Thư viện Computer Vision xử lý hình ảnh

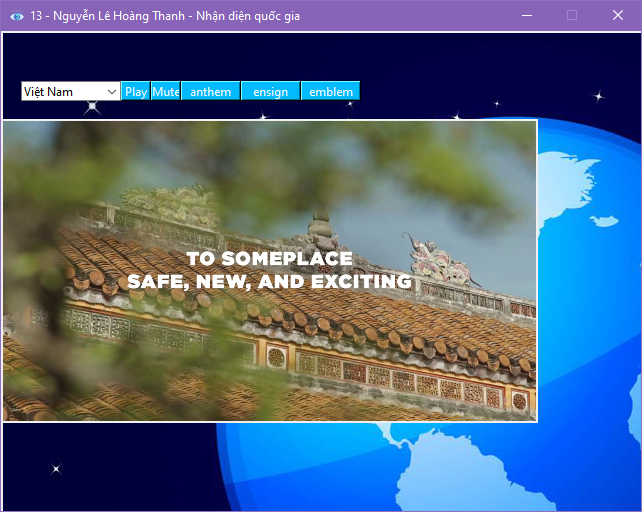
Ffpyplayer: Thư viện phát media

## Thiết kế giao diện và xử lý Video cơ bản



1. **def** open\_vid(self):#Mở video và xử lý
2. self.cap = cv2.VideoCapture(self.img\_link)
3. **try**:
4. self.player = MediaPlayer(self.img\_link)
5. **except**:
6. **pass**
7. **def** show\_frame():
8. ret, self.frame = self.cap.read()
9. **try**: #Âm thanh video
10. audio\_frame, val = self.player.get\_frame()
11. **if** val != 'eof' **and** audio\_frame **is** **not** None:
12. img, t = audio\_frame
13. **except**:
14. **pass**
15. self.frame
16. **if** self.img\_link==0:
17. self.frame = cv2.flip(self.frame, 1)#đảo hình webcam
18. **if** **not** ret:
19. **return**
20. im=self.frame
21. **if** **not** self.checkSlideWin:#sliding window
22. im=apply\_sliding\_window\_on\_3\_channels(im, self.kernel,self.padding,self.stride)
23. im=(im \* 255).round().astype(np.uint8)#chuyển im về kiểu thích hợp để xử lí tiếp
24. cv2image = cv2.cvtColor(im, cv2.COLOR\_BGR2RGBA)
25. cv2image=resize(cv2image)#resize
26. cv2image=rotate(cv2image, self.rote\_data)#xoay
27. **if** self.checkcut==False:#cắt
28. cv2image=cutimg(cv2image,self.h1,self.h2,self.w1,self.w2)
29. **if** self.gray==False: #xám
30. cv2image=rgb\_to\_gray(cv2image )
31. **if** **not** self.checkBlur:#mờ
32. cv2image=cv2.blur(cv2image,(5,5))
33. img = Image.fromarray(cv2image)
34. imgtk = ImageTk.PhotoImage(image=img)
35. self.img\_out=cv2image#truyền frame kết quả ra ngoài
36. #cập nhật
37. self.lmain.imgtk = imgtk
38. self.lmain.configure(image=imgtk)
39. self.after\_id=self.lmain.after(25, show\_frame)
40. show\_frame()#mở frame tiếp theo
41. **pass**
42. **def** sound(self):#bật tắt âm thanh
43. playerVolume = self.player.get\_volume()#Âm lượng hiện tại
44. **print**('Volume = ', playerVolume)
45. **if** playerVolume == 0.0:
46. self.player.set\_volume(1.0)#Bật âm thanh
47. **print**('unmuted')
48. **else**:
49. self.player.set\_volume(0.0)#Tắt âm thanh
50. **print**('muted')
51. **pass**
52. **def** snapshot(self):#Chụp ảnh frame đang phát
53. ts = datetime.datetime.now()
54. filename = "{}.jpg".format(ts.strftime("%Y-%m-%d\_%H-%M-%S"))
55. **try**:#Lưu vào 13NLHT\_Img
56. cv2.imwrite(os.path.join(fc.variable,filename), self.frame.copy())
57. **except**:
58. cv2.imwrite(os.path.join(data\_adress,filename), self.frame.copy())

Và



1. **class** ThongTinQuocGia(tk.Toplevel):
2. **def** \_\_init\_\_(self, master = None):
3. super().\_\_init\_\_(master = master)
4. self.main\_lang=m\_lang.variable#ngôn ngữ
5. self.title(title[self.main\_lang])#title
6. self.geometry("{}x{}".format(height,widht))#kích thước
7. self.iconbitmap(res\_ico)#icon
8. self.resizable(False, False)#chặn resize cửa sổ
9. bg=tk.Label(self,image=background)#nền
10. bg.place(x=0,y=0)
11. #Label phát video
12. self.lm = tk.Label(self)
13. self.lm.pack(side = tk.LEFT)
14. self.frame=None
15. #Lấy danh sách quốc gia theo ngôn ngữ chương trình
16. self.qg\_ten=[]
17. **for**  i **in** Data:
18. self.qg\_ten.append(i[self.main\_lang+1])
19. Combo\_src = ttk.Combobox(self, values = self.qg\_ten)
20. Combo\_src.current(self.qg\_ten.index(Data[self.main\_lang][self.main\_lang+1]))#Quốc gia mặc định
21. Combo\_src.place(x=20,y=50,width=100,height=20)
22. self.quocgia=Data[self.qg\_ten.index(self.qg\_ten[Combo\_src.current()])][0]#Lấy mã quốc gia
23. #print(self.quocgia)
24. #Nut play video
25. btn\_p=tk.Button(self,
26. text="Play",
27. activebackground=color\_2,
28. activeforeground=color\_1,
29. bg=color\_1,
30. fg="#fff",
31. command=self.play )
32. btn\_p.place(x=120,y=50,width=30,height=20)
33. #nut mute
34. btn\_m=tk.Button(self,
35. text="Mute",
36. activebackground=color\_2,
37. activeforeground=color\_1,
38. bg=color\_1,
39. fg="#fff",
40. command=self.mute )
41. btn\_m.place(x=150,y=50,width=30,height=20)
42. #Quoc ca
43. btn\_qc=tk.Button(self,
44. text="anthem",
45. activebackground=color\_2,
46. activeforeground=color\_1,
47. bg=color\_1,
48. fg="#fff",
49. command=self.quocca )
50. btn\_qc.place(x=180,y=50,width=60,height=20)
51. #Quoc ky
52. btn\_qc=tk.Button(self,
53. text="ensign",
54. activebackground=color\_2,
55. activeforeground=color\_1,
56. bg=color\_1,
57. fg="#fff",
58. command=self.quocky )
59. btn\_qc.place(x=240,y=50,width=60,height=20)
60. #Quoc huy
61. btn\_qc=tk.Button(self,
62. text="emblem",
63. activebackground=color\_2,
64. activeforeground=color\_1,
65. bg=color\_1,
66. fg="#fff",
67. command=self.quochuy )
68. btn\_qc.place(x=300,y=50,width=60,height=20)
69. self.Combo\_src=Combo\_src
70. self.open\_vid()
71. #dong cua so
72. self.protocol("WM\_DELETE\_WINDOW", self.on\_closing)
73. **def** quocky(self):#Mở ảnh quốc kỳ
74. quocky=Data[self.qg\_ten.index(self.qg\_ten[self.Combo\_src.current()])][0]#Lấy mã quốc gia đang chọn
75. v="13NLHT\_Res/QuocGia/{}/QK.jpg".format(quocky)
76. self.player.toggle\_pause()#Tạm dừng audio
77. #Mở ảnh
78. image = cv2.imread(v)
79. image = resize(image)
80. cv2.imshow("ensign",image)
81. cv2.waitKey(0)
82. self.player.toggle\_pause()#Mở lại audio
83. **pass**
84. **def** quochuy(self):#Mở ảnh quốc huy
85. quocky=Data[self.qg\_ten.index(self.qg\_ten[self.Combo\_src.current()])][0]#Lấy mã quốc gia đang chọn
86. v="13NLHT\_Res/QuocGia/{}/QH.png".format(quocky)
87. self.player.toggle\_pause()#Tạm dừng audio
88. #Mở ảnh quốc huy
89. image = cv2.imread(v)
90. image = resize(image)
91. cv2.imshow("emblem",image)
92. cv2.waitKey(0)
93. self.player.toggle\_pause()#Mở lại audio
94. **pass**
95. **def** quocca(self):#Mở quốc ca
96. self.cap.release()
97. quocca=Data[self.qg\_ten.index(self.qg\_ten[self.Combo\_src.current()])][0]#Lấy mã quốc gia
98. v="13NLHT\_Res/QuocGia/{}/QC.mp3".format(quocca)
99. self.player = MediaPlayer(v)
100. self.player.start()
101. **pass**
102. **def** mute(self):#Bật tắt âm lượng
103. playerVolume = self.player.get\_volume()#Âm lượng hiện tại
104. **print**('Volume = ', playerVolume)
105. **if** playerVolume == 0.0:
106. self.player.set\_volume(1.0)#Bật
107. **print**('unmuted')
108. **else**:
109. self.player.set\_volume(0.0)#Tắt
110. **print**('muted')
111. **pass**
112. **def** play(self):#Mở video giới thiệu
113. self.quocgia=Data[self.qg\_ten.index(self.qg\_ten[self.Combo\_src.current()])][0]
114. #Reset Label
115. self.lm.destroy()
116. self.lm = tk.Label(self)
117. self.lm.pack(side = tk.LEFT)
118. #mo video
119. self.open\_vid()
120. **def** open\_vid(self):#Mở video với audio
121. v="13NLHT\_Res/QuocGia/{}/VID.mp4".format(self.quocgia)
122. **print**(v)
123. self.cap = cv2.VideoCapture(v)
124. self.player = MediaPlayer(v)
125. **def** show\_frame():
126. ret, self.frame = self.cap.read()
127. #audio
128. audio\_frame, val = self.player.get\_frame()
129. **if** val != 'eof' **and** audio\_frame **is** **not** None:
130. img, t = audio\_frame
131. #video frame
132. self.frame
133. cv2im = cv2.cvtColor(self.frame, cv2.COLOR\_BGR2RGBA)
134. cv2image=resize(cv2im)
135. img = Image.fromarray(cv2image)
136. imgtk = ImageTk.PhotoImage(image=img)
137. self.lm.imgtk = imgtk
138. self.lm.configure(image=imgtk)
139. self.after\_id=self.lm.after(25, show\_frame)
140. show\_frame()
141. **pass**
142. **def** on\_closing(self):#tắt các chức năng khi thoát
143. **try**:
144. self.cap.release()
145. cv2.destroyAllWindows()
146. self.player.close\_player()
147. **except**:
148. **pass**
149. self.destroy()
150. **pass**

## Lồng ghép Audio và Video kết Images

1. **def** open\_vid(self):#Mở video với audio
2. v="13NLHT\_Res/QuocGia/{}/VID.mp4".format(self.quocgia)
3. **print**(v)
4. self.cap = cv2.VideoCapture(v)
5. self.player = MediaPlayer(v)
6. **def** show\_frame():
7. ret, self.frame = self.cap.read()
8. #audio
9. audio\_frame, val = self.player.get\_frame()
10. **if** val != 'eof' **and** audio\_frame **is** **not** None:
11. img, t = audio\_frame
12. #video frame
13. self.frame
14. cv2im = cv2.cvtColor(self.frame, cv2.COLOR\_BGR2RGBA)
15. cv2image=resize(cv2im)
16. img = Image.fromarray(cv2image)
17. imgtk = ImageTk.PhotoImage(image=img)
18. self.lm.imgtk = imgtk
19. self.lm.configure(image=imgtk)
20. self.after\_id=self.lm.after(25, show\_frame)
21. show\_frame()
22. **pass**

## **Ứng dụng của sản phẩm**

13.NguyenLeHoangThanh.B5Ex4\_FramesColorGrayFromVideo

1. **import** cv2
2. **import** os
3. **import** numpy as np
4. #B2: KHAI BÁO CÁC HẰNG
6. INPUT\_VIDEO = '13NguyenLeHoangThanh/HongKong.mp4'
7. OUTPUT\_DIR = '13NguyenLeHoangThanh'
9. #B3: TẠO THƯ MỤC
11. os.makedirs(OUTPUT\_DIR, exist\_ok=True)
12. #B4: THỦ TỤC GHÉP Color Frame với Gray Frame (khác biệt)
14. **def** nlht\_image(img, diff\_im):
15. new\_img = np.zeros([img.shape[0], img.shape[1]\*2, img.shape[2]])#ma trận ảnh trống bằng img
16. new\_img[:, :img.shape[1], :] = img#img vào bên trái
17. #đặt diff\_im vào bên phải
18. new\_img[:, img.shape[1]:, 0] = diff\_im
19. new\_img[:, img.shape[1]:, 1] = diff\_im
20. new\_img[:, img.shape[1]:, 2] = diff\_im
21. cv2.imshow('diff', new\_img)
22. **return** new\_img
23. **def** main(video\_path):
24. cap = cv2.VideoCapture(video\_path)
25. last\_gray = None
26. idx = -1
27. **while**(True):
28. ret, frame = cap.read()
29. idx += 1
30. **if** **not** ret:
31. **print**('Dừng đọc Video vì đã hết (%s)' % video\_path)
32. **break**
33. gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)#chuyển xám
34. **if** last\_gray **is** None:
35. last\_gray = gray
36. **continue**
37. diff = cv2.absdiff(gray, last\_gray)#frame hình khác biệt giữa hình xám trước và sau
38. cv2.imwrite(os.path.join(OUTPUT\_DIR, 'NLHT\_%06d.jpg' %idx), nlht\_image(frame, diff))#ghi ra file
39. last\_gray = gray
40. **print**('Lưu hình thứ : @ %d...' % idx)
41. **pass**
42. **pass**
43. cap.release()
44. cv2.destroyAllWindows()
45. **print**('Chạy chương trình với video clip %s' % INPUT\_VIDEO)
46. main(video\_path=INPUT\_VIDEO)

13.NguyenLeHoangThanh.B4Ex3\_VideoFramesProcessing

1. **import** cv2 # opencv xử lý hình ảnh
3. cap = cv2.VideoCapture("13NguyenLeHoangThanh/HongKong.mp4") #mở video
4. count = 0  #biến đếm khung hình
5. **while** cap.isOpened(): #Kiểm tra mở được video hay không
6. ret,frame = cap.read() # đọc khung hình, chuẩn bị khung tiếp theo
7. cv2.imshow('Khung Hinh', frame)#hiển thị hình ảnh
8. cv2.imwrite("Khung%d.jpg" %count, frame)#lưu hình ảnh vào tệp
9. count = count + 1
10. **if** cv2.waitKey(10) & 0xFF == ord('q'):# chờ phím q được nhấn thì dừng
11. **break**
12. cap.release()
13. cv2.destroyAllWindows()

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

## **Kết luận**

### Những kết quả đạt được

Đề tài đã:

+ Tìm hiểu về kỹ thuật multimedia

+ Thiết kế và lập trình được sản phẩm dạng WinApp minh họa của đề tài dùng PYTHON trên công cụ Spyder (Anaconda3) với thư viện cv2, bao gồm:

. Màn hình chính có chọn chức năng bằng click hoặc âm thanh, thay đổi ngôn ngữ.

. Màn hình chức năng nhận diện quốc gia: Chọn nguồn dữ liệu(webcam, image, video,...), nhận diện hình ảnh quốc kỳ, quốc huy, xử lí ảnh và video đơn giản

. Màn hình chức năng nhận diện ngôn ngữ: Nghe và đọc văn bản, dịch văn bản,...

. Màn hình chức năng thông tin quốc gia: Phát video giới thiệu, quốc ca, hiển thị quộc kỳ, quốc huy

### Hạn chế

Do thời gian hạn chế, nên App sản phẩm của đề tài còn một số hạn chế sau:

+ Phân chức năng: chưa nhận lệnh bằng file audio

+ Độ chích xác nhận diện còn cần được cải thiện

+ Một số xử lí còn chậm trên webcam

## **Hướng phát triển**

### Hướng khắc phục các hạn chế

Sẽ tiếp tục thực hiện :

+ Tăng độ xác khi nhận diện

+ Cải thiện thuật toán để tăng tốc độ xử lí

+ Thêm nhập lệnh bằng file âm thanh

### Hướng mở rộng đề tài

Sẽ nghiên cứu và phát triển app sản phẩm trên Python dùng các thư viện mở.

# DANH MỤC TÀI LIỆU THAM KHẢO

1. Võ Xuân Thể (2021), *Bài giảng học phần KỸ THUẬT LẬP TRÌNH ỨNG DỤNG ĐA PHƯƠNG TIỆN*, Học viện Công nghệ Bưu chính – Viễn thông, cơ sở TP.HCM (lưu hành nội bộ).
2. David G. Lowe(2004), Báo cáo Distinctive Image Features from Scale-Invariant Keypoints, <https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf> (30/11/2021)
3. Life2coding.com, HOW TO CROP OPENCV IMAGE USING MOUSE CLICK AND MOVEMENT IN PYTHON, <https://www.life2coding.com/crop-image-using-mouse-click-movement-python/> (30/11/2021)
4. OpenCV-Python Tutorials, <https://opencv24-python-tutorials.readthedocs.io/> (30/11/2021)