**TRƯỜNG ĐẠI HỌC VINH**

**VIỆN KỸ THUẬT VÀ CÔNG NGHỆ**

**Logo

Description automatically generated**

**BÁO CÁO HỌC PHẦN**

**THỊ GIÁC MÁY TÍNH**

**ĐỒ ÁN TRÍCH XUẤT THÔNG TIN THẺ SINH VIÊN**

|  |  |
| --- | --- |
| GVHD: | Th.S. Nguyễn Thị Minh Tâm |
| Nhóm SVTH: | **Nhóm: 03** |
|  | Nguyễn Tiến Huy MSV: 205748020110203 |
|  | Võ Thị Thu Hương MSV: 205748020110207 |
|  | Đinh Xuân Hoàng MSV: 205748020110106 |
|  | Hồ Hoài Lâm MSV: 205748020110204 |

**Nghệ An – 202****3**

MỤC LỤC

[I. GIỚI THIỆU BÀI TOÁN 2](#_Toc151071103)

[1.1. Giới thiệu bài toán 2](#_Toc151071104)

[1.2. Mục đích đề tài 2](#_Toc151071105)

[II. CÁC THUẬT TOÁN HIỆN NAY 2](#_Toc151071106)

[2.1. Các thuật toán (phương pháp) hiện có trong bài toán 2](#_Toc151071107)

[2.1.1. Chuyển đổi màu và lọc Bilateral 2](#_Toc151071108)

[2.1.2. Phát hiện cạnh bằng Canny 2](#_Toc151071114)

[2.1.3. Tìm Contours và lọc contour 2](#_Toc151071118)

[2.1.5. Hiển thị và cập nhật giao diện 3](#_Toc151071130)

[2.1.6. Xử lý và nhận diện thông tin 3](#_Toc151071135)

[III. TRIỂN KHAI 5](#_Toc151071140)

[1. Quy trình thực hiện 5](#_Toc151071141)

[2. **Giao diện chương trình** 5](#_Toc151071148)

[IV. KẾT LUẬN 5](#_Toc151071150)

[4.1. Kết quả đạt được 5](#_Toc151071151)

[4.2. Những tồn tại và hạn chế 5](#_Toc151071152)

[4.3. Hướng phát triển 5](#_Toc151071153)

[V. PHỤ LỤC 5](#_Toc151071154)

[1. Các bước chạy dự án 5](#_Toc151071155)

[**2.** **Video trình bày về sản phẩm** 5](#_Toc151071290)

# I. GIỚI THIỆU BÀI TOÁN

## 1.1. Giới thiệu bài toán

Trong thời đại 4.0 ngày nay, việc quản lý và trích xuất thông tin của sinh viên trở nên ngày càng phức tạp và quan trọng. Phần mềm lưu trữ thông tin truyền thống dưới dạng văn bản không chỉ đối mặt với hạn chế về dung lượng mà còn gặp khó khăn trong việc quản lý số lượng sinh viên. Điều này tạo ra nhu cầu cấp thiết về một giải pháp thuận tiện và hiệu quả cho việc trích xuất thông tin sinh viên từ thẻ sinh viên.

Bài toán mà chúng em giải quyết là làm thế nào để xây dựng một phần mềm có khả năng nhận dạng và trích xuất thông tin từ thẻ sinh viên một cách nhanh chóng và chính xác. Chúng em chọn ứng dụng ý tưởng này vào lĩnh vực sinh viên, tận dụng tiềm năng của việc nhận diện thông tin từ thẻ sinh viên giống như trong căn cước công dân.

## 1.2. Mục đích đề tài

Mục đích chính của đề tài là xây dựng một phần mềm trích xuất thông tin cho sinh viên, giúp giải quyết nững khó khăn trong quản lý và trích xuất thông tin sinh viên. Phần mềm này có các mục tiêu cụ thể như sau:

- Hiệu quả và nhanh chóng: Tạo ra một giải pháp hiệu quả và nhanh chóng, giúp nhân viên quản lý thông tin sinh viên một cách thuận tiện.

- Nhận diện tự động: Thực hiện nhận diện tự động khi người dùng đưa thẻ sinh viên vào phần mềm.

- Trích xuất thông tin chính xác: Trích xuất các thông tin quan trọng như họ tên, mã sinh viên, ngành học, khoá học từ thẻ sinh viên với độ chính xác cao.

- Ứng dụng rộng rãi: Cung cấp một giải pháp có ứng dụng rộng rãi trong nhiều lĩnh vực như tuyển dụng, quản lý nhân sự, và quản lý học sinh sinh viên.

Mục tiêu của chúng em là không chỉ đáp ứng nhu cầu chính của người dùng mà còn tạo ra một phần mềm linh hoạt, dễ mở rộng để đáp ứng các yêu cầu và thách thức trong tương lai. Đồng thời, chúng em cam kết giảm thiểu sự cố lỗi gây ra do người sử dụng, tạo ra một trải nghiệm người dùng tối ưu.

II. CÁC THUẬT TOÁN HIỆN NAY

## **2.1. Các thuật toán (phương pháp) hiện có trong bài toán**

### 2.1.1. Chuyển đổi màu và lọc Bilateral

Ưu điểm:

- Giảm chiều sâu của dữ liệu, giảm độ phức tạp tính toán.

- Lọc Bilateral giúp giảm nhiễu và giữ lại các đặc trưng quan trọng.

Nhược điểm:

- Có thể mất mát một số thông tin màu sắc.

### 2.1.2. Phát hiện cạnh bằng Canny

Sử dụng thuật toán Canny để phát hiện cạnh trong ảnh, tạo ra ảnh nhị phân với các đường biên rõ nét.

Ưu điểm:

- Có khả năng phát hiện cạnh tốt trong ảnh, giúp xác định rõ biên của các đối tượng.

### 2.1.3. Tìm Contours và lọc contour

Tìm contours trong ảnh sử dụng thuật toán findContours của OpenCV. Lọc ra các contours thẻ sinh viên bằng cách xác định các đa giác có 4 đỉnh, giả định rằng thẻ sinh viên là hình chữ nhật.

Ưu điểm:

- Tìm contours giúp xác định thông tin trên thẻ sinh viên.

- Lọc contour giúp loại bỏ các đối tượng không mong muốn, tập trung vào việc xác định và trích xuất thông tin chính từ thẻ sinh viên.

Nhược điểm:

- Cần đặt ra giả định về hình dạng của thẻ sinh viên. Điều này có thể là một hạn chế nếu có sự biến đổi về kích thước hoặc hình dạng của thẻ.

### 2.1.4. Sử dụng Tesseract OCR

Sử dụng Tesseract OCR để đọc văn bản từ ảnh chứa biển số xe đã được xác định. Cấu hình OCR được điều chỉnh để phù hợp với đặc điểm của biển số xe.

Ưu điểm:

- Mạnh mẽ trong việc đọc văn bản từ hình ảnh.

- Có thể đọc được nhiều kiểu font và kích thước văn bản.

Nhược điểm:

- Nhạy cảm với chất lượng ảnh, độ nghiêng và nhiễu.

### 2.1.5. Hiển thị và cập nhật giao diện

Ưu điểm:

- Sử dụng Tkinter để tạo giao diện người dùng đơn giản và hiệu quả.

Nhược điểm:

- Không có nhược điểm đáng kể được mô tả trong phạm vi mã nguồn cung cấp.

### 2.1.6. Xử lý và nhận diện thông tin

Ưu điểm:

- Sử dụng Pillow và pytesseract để xử lý và nhận diện thông tin từ hỉnh ảnh.

Nhược điểm:

- Tuỳ thuộc vào chất lượng hình ảnh và điều kiện ánh sáng, có thể xảy ra sai sót trong quá trình nhận diện.

# **III. TRIỂN KHAI**

1. Quy trình thực hiện

A diagram of a computer

Description automatically generated with medium confidenceSơ đồ dưới đây tóm tắt các bước truy xuất thông tin thẻ sinh viên:

Đầu tiên từ video sẽ sử dụng bộ lọc nhiễu và chuyển ảnh xám, sau đó sử dụng thuật toán phát hiện cạnh canny để trích xuất những chi tiết cạnh của thẻ sinh viên. Trong quá trình xử lý máy tính có thể nhầm lẫn thẻ sinh viên với những chi tiết nhiễu, việc lọc lần cuối bằng các tỉ lệ cao/rộng hay diện tích của biển số sẽ giúp xác định được đúng thẻ sinh viên. Cuối cùng, ta sẽ xác định vị trí của biển số trong ảnh bằng cách vẽ Contour bao quanh.. Sau khi tìm được vị trí thẻ sinh viên, lần lượt cắt các thông tin có trên thẻ bao gồm: họ tên, ngành, khoa, khóa học, mã sinh viên. Sau đó lần lượt chuyển xám, nhị phân và sử dụng thư viện pytesseract để truy xuất các thông tin trên thẻ. Cuối cùng hiển thị giao diện bằng thư viện tkinter.

1. Lọc nhiễu, chuyển ảnh xám

Ảnh xám (Gray Scale) đơn giản là một hình ảnh trong đó các màu là các sắc thái của màu xám với 256 cấp độ xám biến thiên từ màu đen đến màu trắng, nằm trong giải giá trị từ 0 đến 255, nghĩa là cần 8 bits hay 1 byte để biểu diễn mỗi điểm ảnh này. Lý do cần phải phân biệt giữa ảnh xám và các ảnh khác nằm ở việc ảnh xám cung cấp ít thông tin hơn cho mỗi pixel. Với ảnh thông thường thì mỗi pixel thường được cung cấp 3 trường thông tin trong khi với ảnh xám chỉ có 1 trường thông tin, việc giảm khối lượng thông tin giúp tăng tốc độ xử lý, đơn giản hóa giải thuật nhưng vẫn đảm bảo các tác vụ cần thiết

1. Phát hiện cạnh canny

Trong hình ảnh, thường tồn tại các thành phần như: vùng trơn, góc/cạnh và nhiễu. Cạnh trong ảnh mang đặc trưng quan trọng, thường là thuộc đối tượng trong ảnh. Do đó, để phát hiện cạnh trong ảnh, có nhiều giải thuật khác nhau như toán tử Sobel, toán tử Prewitt, Zero crossing .... Giải thuật Canny hơn hẳn các phương pháp khác do ít bị tác động của nhiễu và cho khả năng phát hiện các biên yếu. Phương pháp này đi theo 4 bước chính:

* Giảm nhiễu (Noise reduction)
* Tính toán Gradient (Gradient calculation)
* Loại bỏ những điểm không phải là cực đại (Non-maximum suppression)
* Lọc ngưỡng (Double threshold)

1. Tìm vị trí thẻ sinh viên bằng Contour

Phát hiện contour là một phương pháp quan trọng trong xử lý ảnh để xác định và biểu diễn các đường biên của đối tượng trong ảnh. Quá trình phát hiện contour:Sử dụng hàm như **cv2.findContours()** trong OpenCV để tìm các contour trong ảnh. Hàm này trả về danh sách các contour và một biến hierarchy (cấu trúc của contour). **Vẽ contour:** Sau khi có danh sách contour, bạn có thể vẽ chúng lên ảnh gốc bằng cách sử dụng hàm như **cv2.drawContours()**

1. Trích xuất thông tin

Sau khi tìm được vị trí của thẻ, tiến hành cắt các vị trí thông tin cần thiết và sử dụng thư viện pytesseract để trích xuất thông tin

Thư viện pytesseract là một thư viện Python giúp bạn nhận dạng văn bản trong hình ảnh. Thư viện này sử dụng Tesseract OCR, một công cụ nhận dạng ký tự quang học mã nguồn mở và miễn phí.

Các khái niệm cơ bản

* Hình ảnh là một tập hợp các điểm ảnh, mỗi điểm ảnh đại diện cho một giá trị màu sắc.
* Văn bản là một tập hợp các ký tự, mỗi ký tự đại diện cho một chữ cái hoặc một ký hiệu.
* Nhận dạng văn bản là quá trình xác định các ký tự trong một hình ảnh và chuyển chúng sang dạng văn bản.

Cách thức hoạt động

Tesseract OCR hoạt động theo các bước sau:

* Tách rời hình ảnh thành các ký tự. Bước này sử dụng các thuật toán phân mảnh hình ảnh để tách rời hình ảnh thành các ký tự hoặc cụm ký tự.
* Phân loại các ký tự. Bước này sử dụng các thuật toán học máy để phân loại các ký tự thành các chữ cái hoặc ký hiệu.
* Tạo ra văn bản. Bước này sử dụng các thuật toán ghép nối để tạo ra văn bản từ các ký tự đã phân loại.

1. Thiết kế giao diện

Sau khi hoàn thành phần chức năng chính của dự án, ta tiến hành lập trình và hiển thị lên giao diện tkinter

Tkinter , viết tắt của "Tk interface", là tiêu chuẩn của Python để xây dựng GUI và được bao gồm trong Thư viện tiêu chuẩn Python . Nó là một ràng buộc với bộ công cụ Tk GUI , một thư viện mã nguồn mở miễn phí của các tiện ích GUI có thể được sử dụng để xây dựng giao diện đồ họa bằng nhiều ngôn ngữ lập trình.

1. Giao diện chương trình

A screenshot of a cell phone

Description automatically generated

# **IV. KẾT LUẬN**

## **4.1. Kết quả đạt được**

Chúng em đã thành công trong việc xây dựng một hệ thống thông tin giúp trích xuất thông tin từ thẻ sinh viên một cách hiệu quả. Qua quá trình triển khai, chúng em đã thực hiện một chu trình rõ ràng, từ phân tích và thiết kế đến xây dựng hệ thống, đặt vai trò và vị trí mỗi yếu tố trong quá trình này.

Tiến trình phân tích và thiết kế hệ thống của chúng tôi đã đảm bảo tính toàn vẹn và hiệu quả của quá trình trích xuất thông tin. Chúng em đã không chỉ trình bày kết quả mà còn thảo luận về vai trò và vị trí của hệ thống trong môi trường thông tin phức tạp ngày nay.

Chúng em cũng đã phát triển kỹ năng giao tiếp, phối hợp và giải quyết vấn đề kỹ thuật trong quá trình làm việc nhóm. Điều này là quan trọng để đảm bảo sự hiệu quả của dự án và sự hài lòng của người sử dụng cuối cùng.

## 4.2. Những tồn tại và hạn chế

Mặc dù đã đạt được những thành công, nhưng chúng em nhận ra rằng phạm vi của chúng em còn hạn chế và chưa bao quát được toàn bộ các khía cạnh của đề tài. Điều này mở ra cơ hội để nghiên cứu và mở rộng hệ thống trong tương lai.

Chúng em cũng thừa nhận rằng còn thiếu kinh nghiệm trong việc phân tích nghiệp vụ. Điều này là một bài học quý báu, và chúng tôi sẽ áp dụng kiến thức này để cải thiện quy trình làm việc của chúng tôi và đảm bảo tính toàn vẹn của thông tin trích xuất.

## 4.3. Hướng phát triển

Chúng em cam kết tiếp tục phát triển và cải thiện hệ thống trích xuất thông tin từ thẻ sinh viên. Cùng với sự phát triển của công nghệ, chúng tôi sẽ liên tục nghiên cứu và tích hợp những tiến bộ mới vào hệ thống của mình.

Hướng phát triển chủ yếu tập trung vào việc mở rộng phạm vi ứng dụng và tối ưu hóa thuật toán nhận diện. Chúng em sẽ liên tục thu thập thông tin và đánh giá hiệu suất để đảm bảo rằng hệ thống của chúng tôi có thể hoạt động chính xác và hiệu quả trong mọi điều kiện.

Chúng em hy vọng rằng hệ thống trích xuất thông tin này sẽ đóng góp tích cực vào việc quản lý thông tin sinh viên và có ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau.

# **V. PHỤ LỤC**

## 1. Các bước chạy dự án

- Cài đặt các thư viện: Opencv, Tkinter, Pillow, Pytesseract

- Cài đặt ngôn ngữ tiếng việt cho thư viện Pytesseract

- Thực hiện chạy phần mã nguồn sau:

import cv2

from tkinter import \*

from tkinter import messagebox

from PIL import ImageTk, Image

import pytesseract

pytesseract.pytesseract.tesseract\_cmd = (r'd:\TGMT\tesseract.exe') #Link đến địa chỉ cài đặt tesseract

cap = cv2.VideoCapture(3)

root = Tk()

root.geometry('1300x770')

root.resizable(width=False, height=False)

root.title("TRÍCH XUẤT THÔNG TIN TỪ THẺ SINH VIÊN")

root.configure(bg='white')

tentruong = Label(root, text="TRƯỜNG ĐẠI HỌC VINH", bg='white',font=('Time 25 bold'), fg='blue')

tentruong.pack(side=TOP)

a = Label(root, text=" ", bg='white',font=('Time 5 bold'))

a.pack(side=TOP)

khoa = Label(root, text="HỌC PHẦN THỊ GIÁC MÁY TÍNH", bg='white',font=('Time 22 bold'))

khoa.pack(side=TOP)

c = Label(root, text=" ", bg='white',font=('Time 10 bold'))

c.pack(side=TOP)

doan = Label(root, text="ĐỒ ÁN", bg='white', fg='red',font=('Time 25 bold'))

doan.pack(side=TOP)

detai = Label(root, text="TRÍCH XUẤT THÔNG TIN TỪ THẺ SINH VIÊN", bg='white', fg='red', font=('Time 25 bold'))

detai.pack(side=TOP)

thoigian = Label(root, text="Vui lòng kiểm tra thông tin", bg='white', fg='blue', font=('Time 25 bold'))

thoigian.place(x= 740, y= 250)

gd\_hoten = Label(root, text="Họ và tên:", bg='white', fg='black', font=('Time 18 bold'))

gd\_hoten.place(x= 745, y= 300)

gd\_ngaysinh = Label(root, text="Ngành:", bg='white', fg='black', font=('Time 18 bold'))

gd\_ngaysinh.place(x= 745, y= 335)

gd\_nganh = Label(root, text="Khoa:", bg='white', fg='black', font=('Time 18 bold'))

gd\_nganh.place(x= 745, y= 370)

gd\_khoa = Label(root, text="Khóa:", bg='white', fg='black', font=('Time 18 bold'))

gd\_khoa.place(x= 745, y= 405)

gd\_mssv = Label(root, text="MSSV:", bg='white', fg='black', font=('Time 18 bold'))

gd\_mssv.place(x= 745, y= 440)

logo = cv2.imread('d:\Dowload\logodhv.jpg')

logo = cv2.resize(logo, (200, 200))

logo = cv2.cvtColor(logo, cv2.COLOR\_BGR2RGB)

img = Image.fromarray(logo)

img = ImageTk.PhotoImage(image=img)

imglabel = Label(root, image=img)

imglabel.place(x=0, y=0)

logo1 = cv2.imread('d:\Dowload\logodhv.jpg')

logo1 = cv2.resize(logo1, (195, 195))

logo1 = cv2.cvtColor(logo1, cv2.COLOR\_BGR2RGB)

img1 = Image.fromarray(logo1)

img1 = ImageTk.PhotoImage(image=img1)

img1label = Label(root, image=img1)

img1label.place(x=1095, y=0)

canvas = Canvas(root, width= 320, height= 240, bg= "white")

canvas.place(x= 50, y= 250)

cut\_tsv = Canvas(root, width= 320, height= 240, bg= "white")

# cut\_tsv = Canvas(root, width= 300, height= 350, bg= "white")

cut\_tsv.place(x= 400, y= 250)

global roi\_resized

def display():

global image1, image2

\_, frame = cap.read()

frame = cv2.resize(frame, (480, 350))

frame = cv2.cvtColor(frame, cv2.COLOR\_BGR2RGB)

image1 = ImageTk.PhotoImage(image=Image.fromarray(frame))

canvas.create\_image(0, 0, image=image1, anchor=NW)

blurred = cv2.GaussianBlur(frame, (5, 5), 0)

img\_gray = cv2.cvtColor(blurred, cv2.COLOR\_BGR2GRAY)

edges = cv2.Canny(img\_gray, 75, 150)

contours, \_ = cv2.findContours(edges, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_SIMPLE)

for contour in contours:

x, y, w, h = cv2.boundingRect(contour)

if w > 100 and h > 100:

global roi\_resized

roi = frame[y:y + h, x:x + w]

roi\_resized = cv2.resize(roi, (320, 240)) # Resize vùng cắt

image2 = ImageTk.PhotoImage(image=Image.fromarray(roi\_resized))

cut\_tsv.create\_image(0, 0, image=image2, anchor=NW)

cv2.imwrite('sv.jpg',roi\_resized)

cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

root.after(1, display)

display()

def txtt():

global roi\_resized, textTEN, textNGANH, textKHOA, textKHOAS, textMSV

HOTEN = roi\_resized[92:118, 90:214]

NGANH = roi\_resized[115:138,112:238]

KHOA = roi\_resized[135:158,80:238]

KHOAS = roi\_resized[155:176,120:238]

MSV = roi\_resized[190:215,238:315]

HOTEN = cv2.resize(HOTEN, None, fx=2.5, fy=2.5, interpolation=cv2.INTER\_CUBIC)

NGANH = cv2.resize(NGANH, None, fx=2.5, fy=2.5, interpolation=cv2.INTER\_CUBIC)

KHOA = cv2.resize(KHOA, None, fx=2.5, fy=2.5, interpolation=cv2.INTER\_CUBIC)

KHOAS = cv2.resize(KHOAS, None, fx=2.5, fy=2.5, interpolation=cv2.INTER\_CUBIC)

MSV = cv2.resize(MSV, None, fx=3, fy=3, interpolation=cv2.INTER\_CUBIC)

grayTEN = cv2.cvtColor(HOTEN, cv2.COLOR\_BGR2GRAY)

grayNGANH = cv2.cvtColor(NGANH, cv2.COLOR\_BGR2GRAY)

grayKHOA = cv2.cvtColor(KHOA, cv2.COLOR\_BGR2GRAY)

grayKHOAS = cv2.cvtColor(KHOAS, cv2.COLOR\_BGR2GRAY)

grayMSV = cv2.cvtColor(MSV, cv2.COLOR\_BGR2GRAY)

ret, threshTEN = cv2.threshold(grayTEN, 170, 255, cv2.THRESH\_BINARY)

ret, threshNGANH = cv2.threshold(grayNGANH, 170, 255, cv2.THRESH\_BINARY)

ret, threshKHOA = cv2.threshold(grayKHOA, 170, 255, cv2.THRESH\_BINARY)

ret, threshKHOAS = cv2.threshold(grayKHOAS, 170, 255, cv2.THRESH\_BINARY)

threshMSV = cv2.adaptiveThreshold(grayMSV,255,cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C, cv2.THRESH\_BINARY,9,2)

textTEN = pytesseract.image\_to\_string((threshTEN), lang='vie')

textNGANH = pytesseract.image\_to\_string((threshNGANH), lang='vie')

textKHOA = pytesseract.image\_to\_string((threshKHOA), lang='vie')

textKHOAS = pytesseract.image\_to\_string((threshKHOAS), lang='vie')

textMSV = pytesseract.image\_to\_string((threshMSV), lang='vie')

print(textTEN)

print(textNGANH)

print(textKHOA)

print(textKHOAS)

print(textMSV)

ht\_TEN = Label(root, text=textTEN, height=2, bg='white', font=('Time 18 bold'))

ht\_TEN.place(x=915, y=300)

ht\_NGANH = Label(root, text=textNGANH, height=2, bg='white', font=('Time 18 bold'))

ht\_NGANH.place(x=930, y=335)

ht\_KHOA = Label(root, text=textKHOA, height=2, bg='white', font=('Time 18 bold'),justify='left')

ht\_KHOA.place(x=915, y=370)

ht\_KHOAS = Label(root, text=textKHOAS, height=2, bg='white', font=('Time 18 bold'))

ht\_KHOAS.place(x=855, y=405)

ht\_MSV = Label(root, text=textMSV, bg='white', fg='black', font=('Time 18 bold'))

ht\_MSV.place(x= 855, y= 440)

#Nút nhấn trích xuất dữ liệu

btn\_txtt = Button(root, text='Trích xuất thông tin', bg='green', fg='white', font=('Time 15 bold'))

btn\_txtt.config(command=txtt)

btn\_txtt.place(x=400, y=560)

#Kết thúc chương trình

root.mainloop()

1. Video trình bày về sản phẩm