

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

**KHOA CÔNG NGHỆ THÔNG TIN**

# BÁO CÁO GIỮA KÌ

##### HỌC PHẦN : PHÁT TRIỂN HỆ THỐNG THÔNG MINH

##### GIẢNG VIÊN : Nguyễn Ngọc Duy

##### ĐỀ TÀI :XÂY DỰNG HỆ THỐNG ĐỌC THÔNG TIN CCCD TỪ ẢNH VÀ XỦ LÝ

NHÓM SINH VIÊN THỰC HIỆN : **Nhóm 12**

**Liêu Minh Kha – N20DCCN110**

**Nguyễn Văn Thịnh – N20DCCN150**

**Lê Phi Hùng – N20DCCN105**

***TP.HCM-2023***

# Mục lục

[Chương I. Xây dựng dữ liệu và đánh giá lựa chọn thuật toán 4](#_bookmark0)

1. [Xây dựng dữ liệu 4](#_bookmark1)
   1. [Tổng quan về hệ thống 4](#_bookmark2)
      1. [Mô tả hệ thống 4](#_bookmark3)
      2. [Yêu cầu Dữ liệu 4](#_bookmark4)
      3. [Định hướng phương pháp xây dựng hệ thống 4](#_bookmark5)
   2. [Thu thập dữ liệu 5](#_bookmark6)
      1. [Thu thập dữ liệu 5](#_bookmark7)
         1. [Tập mẫu 5](#_bookmark8)
      2. [Phân loại mẫu: 6](#_bookmark9)
   3. [Gán nhãn dữ liệu 7](#_bookmark10)
      1. [Gán nhãn cho vùng ảnh có chứa thông tin trên CCCD và gán nhãn 7](#_bookmark11)
      2. [Gán nhãn cho ảnh 8](#_bookmark12)
2. [Lựa chọn và đánh giá các thuật toán 9](#_bookmark13)
   1. [Lựa chọn các thuật toán 9](#_bookmark14)
      1. [Thuật toán phát hiện văn bản (Text Detection) 9](#_bookmark15)
      2. [Thuật toán nhận diện kí tự (Text Recognition) 9](#_bookmark16)
   2. [Đánh giá thuật toán 10](#_bookmark17)
      1. [Các thuật toán phát hiện văn bản 10](#_bookmark18)
      2. [Các thuật toán nhận diện kí tự 11](#_bookmark19)

[Chương II. Thiết kế hệ thống 12](#_bookmark20)

1. [Thiết kế giao diện bên ứng dụng của Client](#_bookmark21) 12
2. [Thiết kế kiến trúc + chi tiết phía Server 13](#_bookmark24)
3. Mô phỏng quá trình [train model 13](#_bookmark29)
4. [Thiết kế hoạt động tương tác giữa Client và Server](#_bookmark30) 14

## Chương I. Xây dựng dữ liệu và đánh giá lựa chọn thuật toán

### Xây dựng dữ liệu

#### Tổng quan về hệ thống

#### Mô tả hệ thống

* Người dùng tải lên ảnh thẻ CCCD mẫu mới nhất(CCCD gắn chip) đưa vào hệ

thống

* Hệ thống xuất ra thông tin như trên thẻ dưới dạng file excel hoặc cho bạn chọn các loại mẫu từ thông dụng và thông tin sẽ được điền vào đó gửi lại cho bạn

##### Các công việc cần thực hiện:

* + Thu thập/xây dựng mẫu
  + Tìm hiểu các thuật toán/ đánh giá và lựa chọn thuật toán phù hợp cho bài toán
  + Xây dựng hệ thống nhận dạng trên server
  + Xây dựng hệ thống client (web browser client) kết nối vào server để chạy

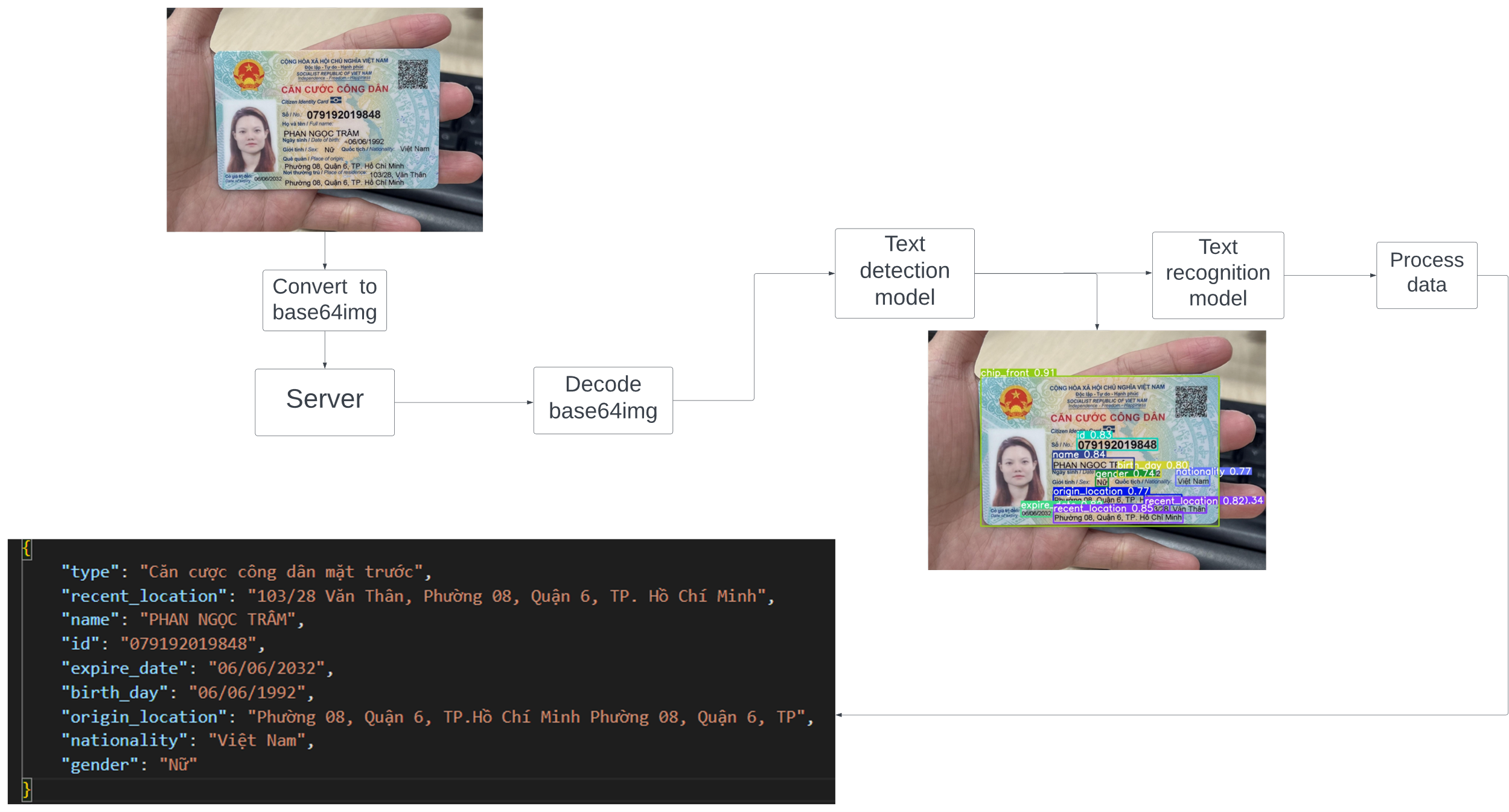
#### Yêu cầu Dữ liệu

Hệ thống trích xuất thông tin cá nhân nên yêu cầu độ chính xác cao và gần như tuyệt đối không được sai sót. Vì vậy nên có một số yêu cầu cho dữ liệu đầu vào:

* Ảnh CCCD mặt trước và mặt sau
* Ảnh chụp rõ nét đầy đủ cả thẻ không bị cắt, mờ, có vật đè lên thông tin để hệ thống có thể đọc được tất cả các chi tiết và thông tin trên ảnh
* Ảnh chụp trên nền có màu khác hoàn toàn với màu nền của CCCD

#### Định hướng phương pháp xây dựng hệ thống

Với dữ liệu trên, mô hình xây dựng hệ thống được đề ra như sau:



1. *Chuyển đổi ảnh thành base64*: Chuyển đổi ảnh thành format base64 để gửi đến server thông qua resfulAPI
2. *Decode base64 thành Img*: Sau khi server nhận được request từ client tiến hành decode base64 thành ảnh để xử lí
3. *Information bounding box location (Text Detection)*: Dùng model đã train để phát hiển những phần text trên ảnh
4. *Text Recognition*: Nhận dạng văn bản
5. *Process Data*: Format lại data trước khi gửi lại cho client

#### Thu thập dữ liệu

#### Thu thập dữ liệu

Một phần vì thời gian, cũng như dữ liệu thật là có hạn, nên quá trình gán nhãn dữ liệu mất rất nhiều thời gian cũng như là bị hạn chế của dữ liệu thật, bên cạnh đó có nhưng bước tiền xử lý, áp dụng xử lý ảnh để bỏ nhiễu cũng như làm giàu dữ liệu, bằng cách xoay ảnh với các góc khác nhau như 00, 900, 1800.

#### Tập mẫu

1068 mẫu là ảnh chứng minh nhân dân, trong đó chia thành 3 tập training set, test set và validation set

* + - * Tập huấn luyện (764 mẫu)
      * Tập kiểm tra (107 mẫu)
      * Tập validation (197 mẫu)

Ảnh được lấy từ các nguồn khác nhau, là ảnh chụp trong môi trường thật với điều kiện khác nhau hoặc là ảnh xuất từ máy tính được chỉnh màu.

#### Phân loại mẫu:

* + - * Mẫu chuẩn (Ảnh CCCD có màu, rõ nét, ngay ngắn không có nền thừa) : **568 mẫu**
      * Mẫu không chuẩn:

+ Ảnh xoay các góc: 233 mẫu

+ Ảnh có nền thừa: 185 mẫu

+ Ảnh mờ: 82 mẫu



***Hình 1****. Minh họa dữ liệu mẫu ảnh chuẩn*



***Hình 2****. Minh họa dữ liệu mẫu ảnh xoay 90 độ, 180 độ và ảnh dính nền*

#### Gán nhãn dữ liệu

#### Gán nhãn cho vùng ảnh có chứa thông tin trên CCCD và gán nhãn

* + - * Xác định nhãn: 12 nhãn (id, name, birthday,gender,nationality, origin\_location, recent\_location, expire\_date,12so\_font, 12so\_back, issue\_date, issue\_place)
      * Mỗi mẫu gồm 12 nhãn
      * Cài đặt tool LabelImg để gán nhãn cho từng mẫu (xác định hộp cho từng vùng chứa thông tin trong ảnh, nhãn sinh ra ứng với mỗi ảnh là một file xml có tên giống với tên ảnh gốc) Nguồn: <https://github.com/heartexlabs/labelImg>



***Hình 3****. Tool LabelImg gán nhãn cho từng mẫu*

#### Format lại nhãn sang yolo v8 format

Sau khi đánh nhãn xong thì vẫn chưa thể train vì chưa đúng format của model cần vì vậy cần phải format lại nhãn theo model, vì model tụi em dùng là YOLOv8 nên tụi em sẽ dùng RobotFlow để format nhãn lại cho đúng với format của model

(Nguồn: <https://roboflow.com/convert/pascal-voc-xml-to-yolov8-pytorch-txt>)

### Lựa chọn các thuật toán

#### Lựa chọn các thuật toán

#### Thuật toán phát hiện văn bản (Text Detection)

**\* Thuật toán được nhóm em lựa chọn là Yolo version 8**

Thuật toán YOLO (You Only Look Once) là một trong những thuật toán phát hiện văn bản quan trọng trong lĩnh vực thị giác máy tính và trí tuệ nhân tạo. Nó được thiết kế để phát hiện và xác định vị trí của các đối tượng trong hình ảnh một cách nhanh chóng và chính xác. Dưới đây là một số thông tin cơ bản về YOLO:

-Tổng quan về YOLO: YOLO là một mô hình đa lớp dựa trên mạng neural convolutional (CNN). Tên gọi "You Only Look Once" ám chỉ rằng YOLO chỉ cần xem một lần vào hình ảnh để thực hiện phát hiện và phân loại đối tượng. Nó giúp tăng tốc quá trình phát hiện đối tượng so với các phương pháp truyền thống.

-Phương thức hoạt động: YOLO chia hình ảnh thành một lưới ô lưới (grid) và dự đoán các hộp giới hạn (bounding boxes) cho đối tượng trong mỗi ô lưới. Mỗi hộp giới hạn được dự đoán kèm theo xác suất phân loại của đối tượng trong hộp đó. Dự đoán này được thực hiện trong một lớp đầu ra duy nhất của mạng, làm giảm thiểu việc tính toán so với các phương pháp tạo ra nhiều lớp đầu ra.

-Ưu điểm: Tốc độ cao: YOLO nhanh chóng và hiệu quả trong việc phát hiện đối tượng trong thời gian thực. Độ chính xác: YOLO có khả năng xác định nhiều đối tượng khác nhau trong cùng một hình ảnh. Tích hợp: Có nhiều phiên bản của YOLO có thể tích hợp vào các ứng dụng và dự án khác nhau.

-Ứng dụng: YOLO được sử dụng rộng rãi trong các ứng dụng như phát hiện vật thể trong hình ảnh và video, xe tự hành, giám sát an ninh, và nhiều ứng dụng khác.

Nguồn: https://github.com/ultralytics/ultralytics

#### Thuật toán nhận diện kí tự (Text Recognition)

**\* Thuật đoạn được nhóm em lựa chọn là VietOCR:**

VietOCR là một ứng dụng và thư viện mã nguồn mở được sử dụng để nhận dạng ký tự tiếng Việt trong hình ảnh hoặc tệp PDF. Dưới đây là một số thông tin về thuật toán và ứng dụng VietOCR:

-OCR (Optical Character Recognition): VietOCR sử dụng OCR, hay còn gọi là công nghệ nhận dạng ký tự quang học, để chuyển đổi các hình ảnh hoặc tệp PDF chứa văn bản tiếng Việt thành văn bản có thể xem và chỉnh sửa được trên máy tính.

-Công nghệ OCR: VietOCR có thể sử dụng nhiều công nghệ OCR khác nhau, bao gồm Tesseract OCR, một thư viện mã nguồn mở phổ biến cho OCR, và VietOCR.NET, một phiên bản được tùy chỉnh cho tiếng Việt.

-Đặc điểm chính: Hỗ trợ đa ngôn ngữ: VietOCR không chỉ hỗ trợ tiếng Việt mà còn có khả năng nhận dạng văn bản trong nhiều ngôn ngữ khác nhau. Giao diện người dùng đơn giản: Ứng dụng cung cấp một giao diện người dùng thân thiện và dễ sử dụng. Kết hợp với giao diện dòng lệnh: Ngoài giao diện đồ họa, VietOCR cũng cung cấp khả năng sử dụng từ dòng lệnh để tích hợp vào các quy trình tự động hóa.

-Ứng dụng: VietOCR được sử dụng rộng rãi trong việc chuyển đổi các tài liệu tiếng Việt từ hình ảnh hoặc PDF sang văn bản có thể chỉnh sửa được. Điều này hữu ích cho việc số hóa tài liệu, quản lý văn bản, và nhiều ứng dụng khác trong lĩnh vực công nghệ thông tin và văn phòng.

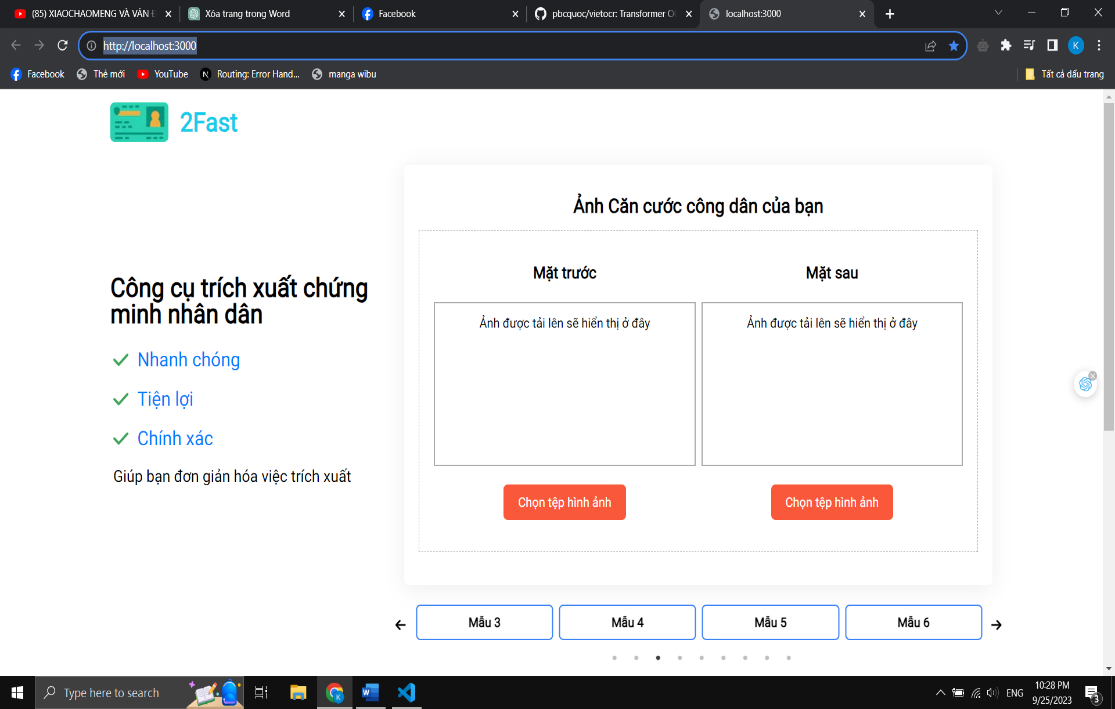
Nguồn: https://github.com/pbcquoc/vietocr

## 

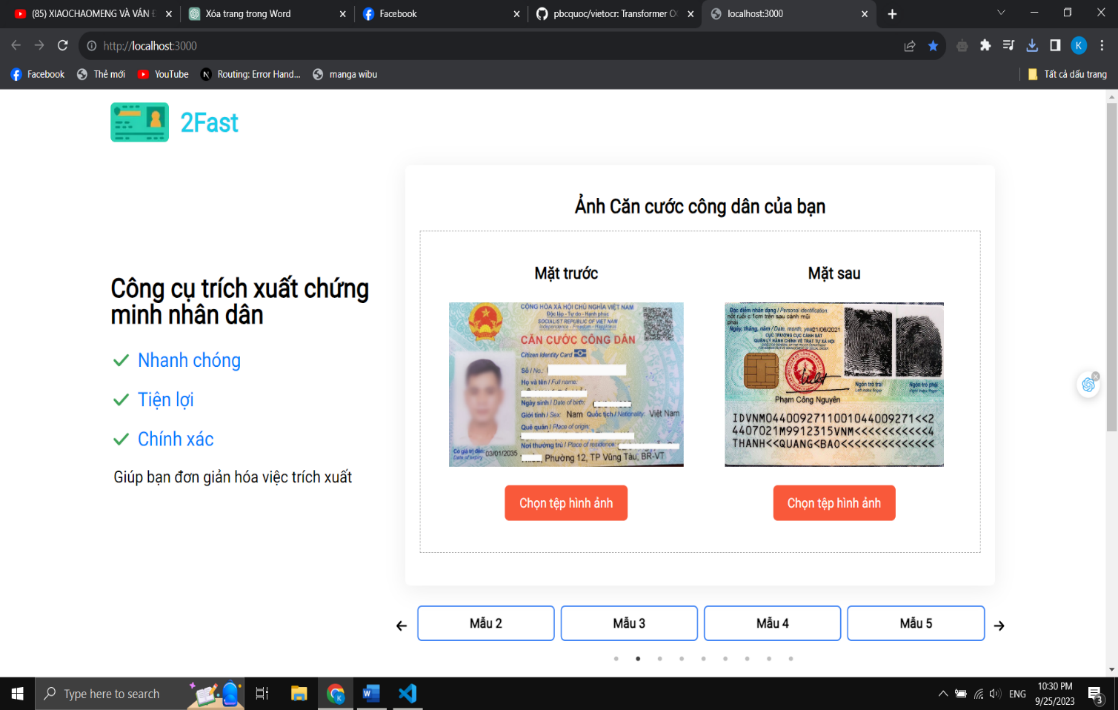
## Chương II. Thiết kế hệ thống

### Thiết kế giao diện cho bên ứng dụng của Client.

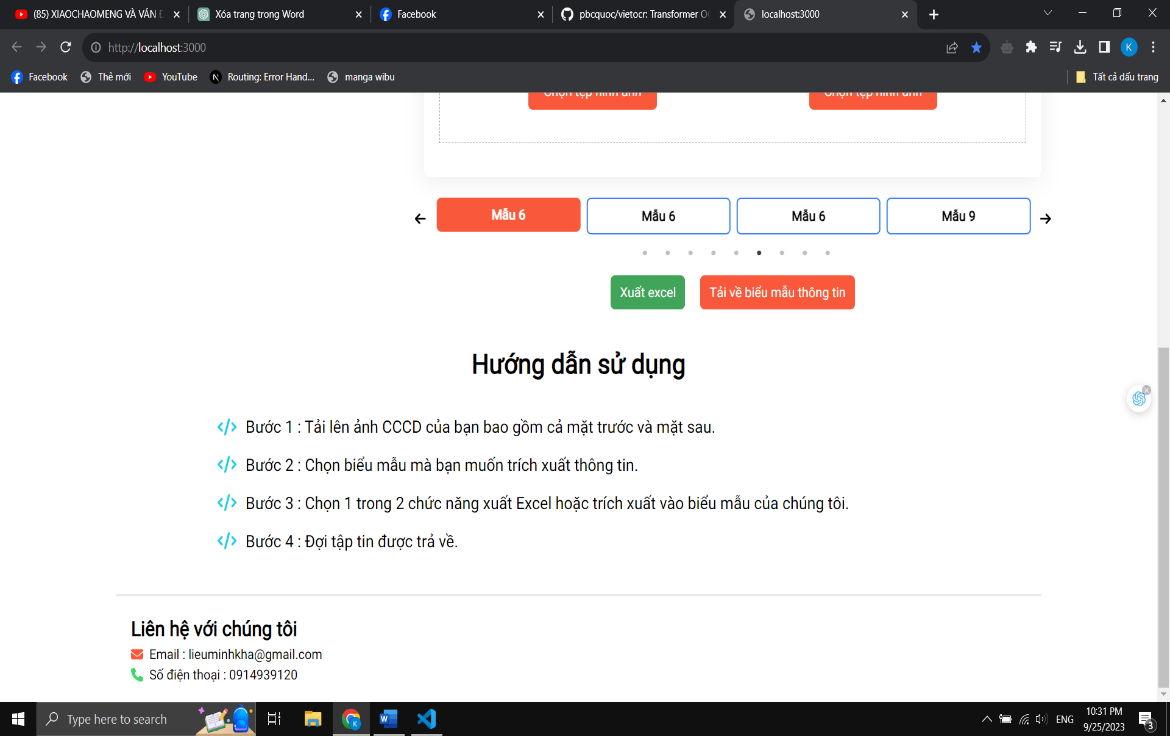
Giao diện khi chưa tải ảnh lên:



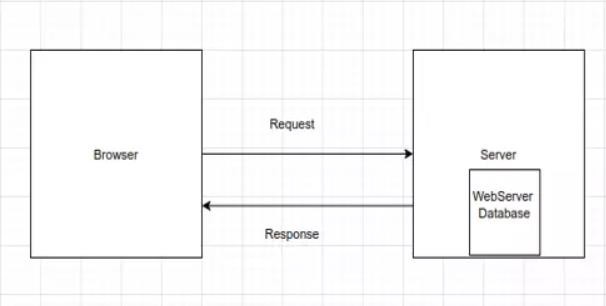
Giao diện khi tải ảnh lên thành công



Giao diện chức năng chính



### Thiết kế kiến trúc + chi tiết phía Server



*Kiến trúc bao quát của Server*

Sau khi nhận ảnh từ client về, Server sẽ thực hiện các chức năng cơ bản sau:

* + - * Decode base64 ảnh từ dữ liệu database Khi nhận dữ liệu từ cơ sở dữ liệu, chúng tôi thực hiện giải mã base64 để chuyển đổi dữ liệu thành hình ảnh gốc.
      * Phát hiện box chứa kí tự bằng mô hình YOLOv8 Hình ảnh sau khi được giải mã được đưa vào mô hình YOLOv8, một mô hình dựa trên YOLO (You Only Look Once), đã được huấn luyện trước đó để phát hiện và định vị các box chứa kí tự.
      * Nhận diện kí tự bằng mô hình VietOCR Các box chứa kí tự được đưa vào mô hình VietOCR, một mô hình OCR (Optical Character Recognition) được huấn luyện trước với tiếng Việt, để nhận diện và trích xuất thông tin từ các kí tự trong các box.
      * Sắp xếp và định dạng lại dữ liệu Dữ liệu trích xuất từ kí tự được sắp xếp và định dạng lại để đảm bảo tính cấu trúc và đúng định dạng.
      * Gửi dữ liệu về cho người dùng Sau khi xử lý và định dạng lại dữ liệu, kết quả cuối cùng được gửi về cho người dùng để sử dụng và xem xét.
      * Với các bước xử lý được mô tả trên, hệ thống có khả năng xử lý dữ liệu từ căn cước công dân một cách hiệu quả và tổ chức dữ liệu trích xuất thành thông tin có cấu trúc, từ đó cung cấp giá trị và hỗ trợ cho ứng dụng và dịch vụ khác. Bằng cách sử dụng mô hình YOLOv8 và VietOCR, chúng tôi đảm bảo hiệu suất và độ chính xác tốt trong quá trình xử lý dữ liệu.

### Mô phỏng quá trình train model

Bằng việc sử dụng Google Colab, các công việc cần xử lý khi training dữ liệu được hỗ trợ thực hiện thông qua các bước:

Để cài đặt và training model trên Colab làm các bước sau

- Cài đặt thư viện ultralitics



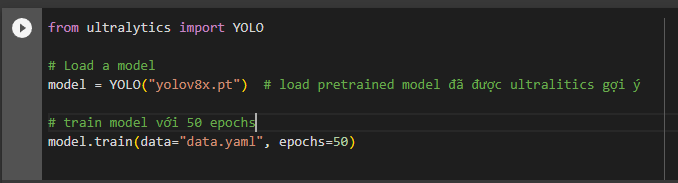
* + - * Upload dataset lên google drive và connect colab với drive



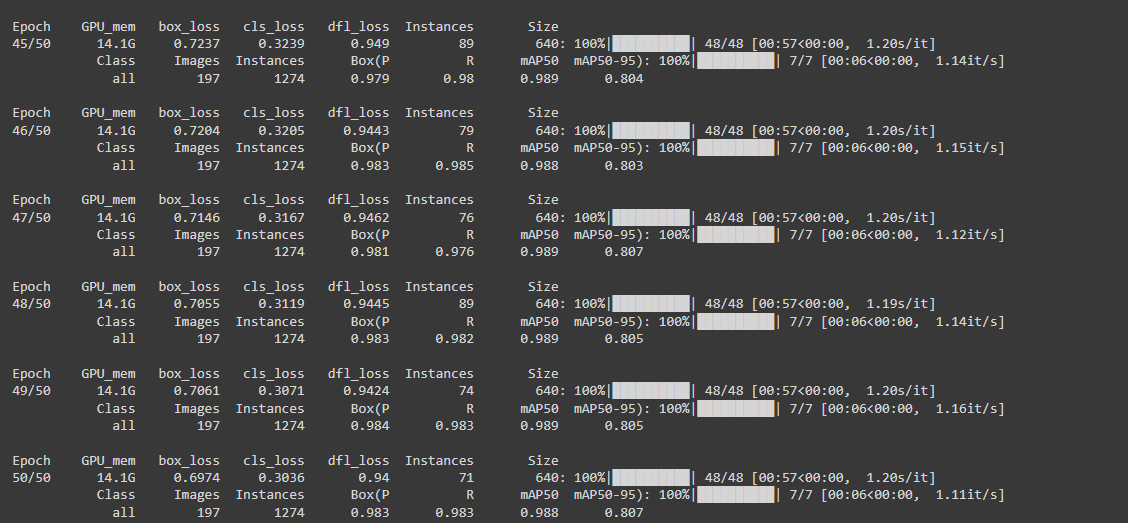
* + - * Unzip dataset



* + - * Load model và train model



* + - * Quá trình train model



### Thiết kế hoạt động tương tác giữa Client và Server

* + - * Liên kết giữa Client và Server được thực hiện thông qua một RESTful API (Representational State Transfer Application Programming Interface). RESTful API là một tiêu chuẩn phổ biến cho việc thiết kế các giao diện giao tiếp giữa các hệ thống phân tán.
      * RESTful API Design: API được thiết kế theo kiến trúc REST, tuân thủ các nguyên tắc như sử dụng các phương thức HTTP (GET, POST, PUT, DELETE) để thao tác với tài nguyên, sử dụng các đường dẫn thể hiện tài nguyên, và truyền dữ liệu dưới dạng JSON hoặc XML.
      * RESTful API cung cấp một cách tiêu chuẩn và linh hoạt để Client và Server giao tiếp và trao đổi thông tin. Điều này giúp tăng tính linh hoạt, mở rộng và khả năng mở rộng của hệ thống.