**Hệ thống quản lý bãi đỗ xe**

1. PipeLine hệ thống

A diagram of a diagram

Description automatically generated

Hình 1: Pipeline Hệ thống

2. Xây dựng bộ dữ liệu

2.1 Thông tin thu thập dữ liệu

* Nguồn thu thập dữ liệu: MiAI.com

2.2 Công cụ label dữ liệu

* [LabelImg](https://github.com/tzutalin/labelImg). Lý do em chọn LabelImg để label:
  + Giao diện khá tốt với đầy đủ chức năng: open, load, autosave, ...
  + Định dạng của nhãn gồm 3 format:XML, YOLO, JSON.
  + Dễ cài đặt và sử dụng
* Quy tắc khi label:
  + Đối với .bộ dữ liệu Plate Detection
    - Label bounding box ôm gọn vùng chứa biển số xe.
  + Đối với bộ dữ liệu Character Recognition
    - Label bounding box ôm gọn các kí tự trong biển số xe.
    - Đối với những kí tự mờ chúng ta không nhận ra được thì sẽ không tiến hành label.
* Đối với mỗi mỗi bức ảnh sau khi gán nhãn sẽ tạo ra 1 file txt được gọi là file annotation.
* Các thành phần của 1 file annotaion như sau:
  + Mỗi dòng là thông tin của 1 bounding box:
  + id: Thứ tự của class do mình định nghĩa.
  + Center: Là tọa độ tâm của bounding box (X\_center, Y\_center). Tính như sau:

[Diagram

Description automatically generated](https://github.com/lynguyenminhuit/CS114.M11/blob/master/Final_Project/Image%20in%20report/center.png?raw=true)

* + Width, Height là chiều dài của bounding box theo chiều ngang và chiều cao của bức ảnh.

2.3 Kết quả thu thập dữ liệu

2.3.1. Bộ dữ liệu Plate Detection

* Bộ dữ liệu Plate Detection sau khi thu thập gồm 7719 ảnh.
* Cách chia dữ liệu: Chia theo tỉ lệ 6:2:2. Trong đó:
  + Tập train: gồm 4632 ảnh
  + Tập valid: gồm 1537 ảnh
  + Tập test: gồm 1550 ảnh

2.3.2. Bộ dữ liệu Character Recognition

* Bộ dữ liệu sau khi thu thập gồm 3190 ảnh.
* Cách chia dữ liệu: Chia theo tỉ lệ 6:2:2. Trong đó:
  + Tập train: gồm 1913 ảnh
  + Tập valid: gồm 640 ảnh
  + Tập test: gồm 637 ảnh
* Trong tập training:

A graph of blue bars

Description automatically generated

Hình 2: Histogram các class tập training

* Trong tập Valid:

A graph of blue bars

Description automatically generated

Hình 3: Histogram các class tập Valid

* Trong tập Test:

A graph of blue bars

Description automatically generated

Hình 4: Histogram tập test

=> Nhân xét: Các tập dữ liệu các sự chênh lệch rất lớn giữa các class.

3. Training và đánh giá Model

3.1. Môi trường huấn luyện và đánh gíá

* Môi trường train và đánh giá:
  + Google colab là một virtual cloud machine được google cung cấp miễn phí cho các nhà nghiên cứu. Đây là môi trường lý tưởng để phát triển các mô hình vừa và nhỏ. Điểm tuyệt vời ở google colab đó là môi trường của nó đã cài sẵn các packages machine learning và frame works deep learning thông dụng nhất.

Text

Description automatically generated

Hình 5: Môi trường huấn luyện và đánh giá

* Do quá trình tải dữ liệu lên Colab tốn thời gian, và sau mỗi phiên colab (khoảng 5 tiếng) thì dữ liệu sẽ mất hết. Do đó chúng em lưu trữ dữ liệu bài toán trên google drive, sau đó kết nối drive với colab.

3.2. Quy trình training

* Để bắt đầu train, chúng em sử dụng file Pretrained Weights để tiếp tục train cho model của mình. Sử dụng weight này là vì:
  + Điều kiện thiết bị không đáp ứng để train mô hình từ đầu.
  + Sử dụng file Pretrained Weights giúp tiết kiệm thời gian train so với train lại toàn bộ model từ đầu.
  + Pretrain này được train trên tập MSCOCO, nhưng lớp cuối (Lớp dùng để phân loại) không được sử dụng để train tiếp trong bài này. Bằng cách sử dụng pretrain này, chúng ta có thể phát hiện những đặc trưng như: đường tròn, đường thẳng, các đặc trưng phức tạp từ tập MSCOCO, từ đó áp dụng tốt hơn vào bài toán.
* Quá trình training model:
  + Gitclone repo:
    - YOLOv7: <https://github.com/WongKinYiu/yolov7>
  + Cài môi trường như hướng dẫn trong repo.
  + Set up lại cây thư mục chứa data riêng đối với từng mô hình.
  + Setup lại file .yalm.
  + Chạy lệnh để train. Sau khi chạy hết 1 epoch, file weight sẽ tự động lưu trong ../runs/train.

3.3. Kết quả đánh giá mô hình

A graph of a graph

Description automatically generated

Hình 6: Quá trình Huấn luyện mô hình LP detection

A graph of a graph

Description automatically generated

Hình 7: Quá trình huấn luyện tập OCR

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | IOU thresh | Epochs Training | mAP@.5:.95 | mAP@.5 | Tốc độ detect (trên ảnh 640x640) | Kích thước mô hình |
| LP\_Detection | 0.5 | 35 | 0.473 | 0.95 | 10.7 ms | 71.3 MB |
| OCR\_Detection | 0.7 | 80 | 0.699 | 0.919 | 14.3 ms | 71.6 MB |

Bảng 1: Kết quả trên tập testing của 2 mô hình



Hình 8: Kết quả trên tập test mô hình LP Detection

A table of numbers with numbers

Description automatically generated

A number with numbers and a number on it

Description automatically generated

Hình 9: Kết quả trên tập test mô hình OCR

=> Nhận xét trên mô hình OCR tồn tại các class O,Q,W có các chỉ số đánh giá rất thấp. Lý do là bởi số lượng các phần tử các class quá ít, cụ thể là 1.

4. Công cụ phát triển ứng dụng:

* Front-end: Tkiner Python
* Back-end: Database using myPHPAdmin

5. Demo

* Cần tải xampp để điều khiển database
* Môi trường chạy thực nghiệm:
  + OS: Window 10
  + Python: 3.10
* Cách cài đặt vào sử dụng:
  + B1: Tải source code:
  + Git clone https://github.com/VanQuyen02/Parking\_System\_Managment\_Using\_YOLOv7.git
  + B2: pip install -r requirement.txt
  + B3: python gui.py