BÁO CÁO ĐỒ ÁN TRÍ TUỆ NHÂN TẠO CHO HỆ THỐNG NHÚNG

Số thứ tự đề tài: 2 (RSUD20K)

Nhóm SV thực hiện: 2ae

Phan Phước Đại – 22520181

Nguyễn Phạm Quang Bình - 22520133

I. Thiết kế mô hình

1. Tổng quan

- Quá trình thực hiện mô hình:

+ Tìm hiểu về các mô hình object detection trên mạng (mobilenet ssd, YOLO, …).

+ Chọn mô hình YOLO vì dễ thực hiện và hiệu quả.

- Mô tả vắn tắt (từ 2-3 câu) nội dung từng bước thực hiện:

+ **Bước 1**: Tạo file notebook trên Kaggle.

+ **Bước 2**: Upload tập dữ liệu lên Kaggle.

+ **Bước 3**: Tinh chỉnh các thiết lập cần thiết cho quá trình huấn luyện.

+ **Bước 4**: Đánh giá mô hình bằng lệnh CLI.

+ **Bước 5**: Chuyển mô hình qua định dạng tflite.

+ **Bước 6**: Chạy mô hình trên Pi.

- Lý do thực hiện:

+ Kaggle cho phép người dùng sử dụng tận 30 tiếng với GPU của họ, hơn hẳn Google Colab.

+ Mô hình YOLO được hỗ trợ CLI từ Ultralytics nên rất dễ hiện thực hóa.

2. Nội dung chi tiết

2.1 Bước 1

Tạo một file notebook để tiến hành huấn luyện mô hình trên Kaggle.

2.2 Bước 2

Upload tất cả các tập dữ liệu gồm tập gốc có 18681 ảnh train, 1004 ảnh val, 649 ảnh test và 200 ảnh test đã được lọc ra. Có thể upload thông qua chức năng Upload -> New dataset trên notebook Kaggle.

2.3 Bước 3

- Chọn Accelerator là GPU P100 và ngôn ngữ Python. Cài đặt các thư viện cần thiết bằng lệnh pip để cài ultralytics.

- Sau khi đã cài đặt xong các thư viện cần thiết thì tiến hành tinh chỉnh file data.yaml sao cho file chỉ đúng tới đường dẫn chứa dataset và đúng số lượng class.

- Huấn luyện mô hình bằng CLI có sẵn của Ultralytics.

2.4 Bước 4

Sau khi đã có mô hình thực hiện đánh giá bằng lệnh val của Ultralytics trên tập val, tập test và tập test đã chọn lọc.

2.5 Bước 5

Thực hiện việc export bằng CLI của Ultralytics ra định dạng .tflite.

2.5 Bước 6

- Thực hiện SSH laptop cá nhân với Pi, cài đặt thư viện opencv lên Pi.

- Tải phần mềm FileZilla và chuyển tất cả các file main.py, model.py, metrics.py, nms.py, mô hình tflite và tập dữ liệu 200 ảnh sang Pi.

- Chạy mô hình và nhận kết quả.

II. Thực nghiệm và đánh giá

1. Thực nghiệm

- Do trong quá trình huấn luyện, nhóm em nhận thấy mô hình bắt đầu không tăng mAP sau 70 epochs nên đã chọn 70 epochs làm tổng số epochs huấn luyện.

- Đối với các hyperparameter khác như learning rate và các augmentation, sau khi đã thử nghiệm với các learning rate và augmentation khác nhau thì nhận thấy ít cải thiện hoặc thậm chí không cải thiện mAP nên đã lựa chọn giữ nguyên hyperparameter gốc.

2. Kết quả

```
!yolo val model=/content/best_float32.tflite data=/content/data.yaml imgsz=320

WARNING ⚠️ Unable to automatically guess model task, assuming 'task=detect'. Explicitly de
Ultralytics 8.3.152 🚀 Python-3.11.13 torch-2.6.0+cu124 CPU (Intel Xeon 2.20GHz)
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1749565415.990228    4843 cuda_dnn.cc:8310] Unable to register cuDNN factory: A
E0000 00:00:1749565415.997864    4843 cuda_blas.cc:1418] Unable to register cuBLAS factory
Loading /content/best_float32.tflite for TensorFlow Lite inference...
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Setting batch=1 input of shape (1, 3, 320, 320)
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.t
100% 755k/755k [00:00<00:00, 24.2MB/s]
val: Fast image access ✅ (ping: 0.0±0.0 ms, read: 75.4±9.3 MB/s, size: 477.3 KB)
val: Scanning /content/full_dataset/labels/val... 1004 images, 0 backgrounds, 0 corrupt: 10
val: New cache created: /content/full_dataset/labels/val.cache
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 1
                   all       1004       7385       0.65      0.469      0.536      0.391
                person        566       1917      0.801      0.459      0.597      0.352
              rickshaw        648       1587      0.762      0.643       0.73      0.538
         rickshaw van         160        240      0.529      0.402      0.415      0.206
        auto rickshaw         388        590      0.722      0.718      0.768      0.598
                 truck         62         65      0.588      0.446      0.533      0.449
           pickup truck         66         74      0.526      0.405      0.388      0.253
            private car        776       1420      0.826      0.661      0.767      0.582
              motorcycle       550        860      0.784       0.59      0.665      0.455
                 bicycle       135        146      0.585      0.222      0.291       0.17
                     bus       169        182      0.743      0.544      0.641       0.48
               micro bus       215        241       0.67      0.502      0.595      0.512
             covered van        38         40       0.69      0.325      0.395      0.335
           human hauler         22         23      0.231      0.174      0.186      0.157
Speed: 0.8ms preprocess, 57.9ms inference, 0.0ms loss, 1.7ms postprocess per image
```

Kết quả chạy trên tập val trên Colab

```
!yolo val model=/content/best_float32.tflite data=/content/data.yaml imgsz=320

WARNING ⚠️ Unable to automatically guess model task, assuming 'task=detect'. Explicitly d
Ultralytics 8.3.152 🚀 Python-3.11.13 torch-2.6.0+cu124 CPU (Intel Xeon 2.20GHz)
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1749565574.321443    5495 cuda_dnn.cc:8310] Unable to register cuDNN factory:
E0000 00:00:1749565574.329631    5495 cuda_blas.cc:1418] Unable to register cuBLAS factory
Loading /content/best_float32.tflite for TensorFlow Lite inference...
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Setting batch=1 input of shape (1, 3, 320, 320)
val: Fast image access ✅ (ping: 0.0±0.0 ms, read: 53.6±11.1 MB/s, size: 543.1 KB)
val: Scanning /content/full_dataset/labels/test... 649 images, 0 backgrounds, 0 corrupt: 1
val: New cache created: /content/full_dataset/labels/test.cache
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95):
                   all        649       3805      0.658      0.601      0.643      0.479
                person        365        844      0.719      0.598      0.674      0.437
              rickshaw        511       1129      0.786      0.798      0.863      0.678
         rickshaw van          78         83      0.528      0.614        0.6      0.343
        auto rickshaw         168        228      0.737      0.749      0.806      0.666
                 truck         23         29      0.254      0.138      0.104     0.0675
           pickup truck         65         65      0.665      0.671      0.731      0.502
            private car        355        543      0.803      0.808      0.864      0.687
              motorcycle       343        509      0.807      0.756      0.811      0.542
                 bicycle       114        121      0.739      0.554      0.649      0.461
                     bus        66         86      0.516      0.581      0.519      0.395
               micro bus       105        105       0.65      0.602      0.715      0.608
             covered van        24         24      0.626      0.349      0.398      0.344
           human hauler         28         39      0.718       0.59       0.62      0.501
Speed: 0.6ms preprocess, 50.9ms inference, 0.0ms loss, 1.3ms postprocess per image
```

Kết quả chạy trên tập test (RSUD20K) trên Colab

```
!yolo val model=/content/best_float32.tflite data=/content/data.yaml imgsz=320

WARNING ⚠️ Unable to automatically guess model task, assuming 'task=detect'. Explicitly d
Ultralytics 8.3.152 🚀 Python-3.11.13 torch-2.6.0+cu124 CPU (Intel Xeon 2.20GHz)
WARNING: All log messages before absl::InitializeLog() is called are written to STDERR
E0000 00:00:1749565684.758865    5959 cuda_dnn.cc:8310] Unable to register cuDNN factory:
E0000 00:00:1749565684.766434    5959 cuda_blas.cc:1418] Unable to register cuBLAS factory
Loading /content/best_float32.tflite for TensorFlow Lite inference...
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Setting batch=1 input of shape (1, 3, 320, 320)
val: Fast image access ✅ (ping: 0.0±0.0 ms, read: 80.6±20.4 MB/s, size: 509.6 KB)
val: Scanning /content/test_dataset/rsud20k/labels/test... 200 images, 0 backgrounds, 0 co
val: New cache created: /content/test_dataset/rsud20k/labels/test.cache
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95):
                   all        200       1087      0.593      0.537      0.583      0.433
                person         78        202      0.762      0.635      0.701      0.461
              rickshaw        131        302      0.813      0.821      0.875       0.72
          rickshaw van         21         23      0.598      0.652      0.667      0.341
         auto rickshaw         31         37      0.741      0.784      0.829       0.72
                 truck         15         21       0.47       0.19       0.31      0.207
            pickup truck          5          5          0          0     0.0663     0.0481
             private car        106        190      0.813      0.874        0.9       0.69
              motorcycle        136        196      0.939      0.788      0.864      0.564
                 bicycle          7          7      0.573      0.714      0.672        0.5
                     bus          9         12      0.131      0.417      0.127     0.0797
               micro bus         77         77       0.86      0.558      0.872      0.752
             covered van         10         10      0.639        0.2      0.378       0.32
            human hauler          5          5      0.365      0.353      0.313       0.22
Speed: 0.6ms preprocess, 49.2ms inference, 0.0ms loss, 1.3ms postprocess per image
```

Kết quả chạy trên tập 200 ảnh test trên Colab

```
Average FPS: 0.043
               Class  Instances          P          R      mAP50   mAP50-95         F1
                 all       1087      0.569      0.163      0.368      0.233      0.244
Score: 0.009
```

Kết quả chạy trên kit Pi

3. Đánh giá, nhận xét

- Kết quả nhận được trên Colab là tạm ổn nhưng trên kit Pi thì mAP, F1 và FPS khá thấp.

- Có 2 lý do, lý do thứ nhất là vì mô hình còn ở dạng float32 nên việc inference trên Pi là khá chậm, lý do mô hình vẫn còn ở float32 là vì nhóm chúng em không đủ khả năng để tinh chỉnh file model.py để có thể chạy được float16 hoặc full int.

- Lý do thứ hai là vì Pi Zero W khá hạn chế về mặt phần cứng, điển hình là việc phải tăng tensor arena size từ 4MB lên 7MB để chạy được mô hình.

- Hơn nữa, nhóm em **không chạy được code inference trên Colab** vì lỗi cài đặt thư viện hoặc môi trường của file main.py hoặc model.py, **nhóm chỉ có thể chạy được code inference trên Pi**.

- Vì vậy nhóm chúng em đánh giá mô hình này còn chậm và chưa đáp ứng được yêu cầu.

III. Bảng phân công công việc

Họ và tên	MSSV	Phân công	Đánh giá
Phan Phước Đại	22520181	Huấn luyện mô hình, chạy mô hình trên Pi	- Tỷ lệ đóng góp: 100% - Điểm nhóm đánh giá: 10/10
Nguyễn Phạm Quang Bình	22520133	Huấn luyện mô hình, chạy mô hình trên Pi	- Tỷ lệ đóng góp: 100% - Điểm nhóm đánh giá: 10/10

IV. Tài liệu tham khảo

[1] Ultralytics, "YOLOv11 Models," *Ultralytics Documentation*, 2024. [Online]. Tại: https://docs.ultralytics.com/models/yolo11/ [Tham khảo: 12-5-2025].

[2] E. Juras, "Train_YOLO_Models.ipynb," *Google Colab*, 2023. [Online]. Tại: https://colab.research.google.com/github/EdjeElectronics/Train-and-Deploy-YOLO-Models/blob/main/Train_YOLO_Models.ipynb [Tham khảo: 7-5-2025].

[3] E. Juras, "How to Train YOLO Object Detection Models in Google Colab," *YouTube*, 2025. [Online]. Tại: https://www.youtube.com/watch?v=r0RspiLG260 [Tham khảo: 7-5-2025].