

Lab4: Object detection (YOLO)

3099704 AI for Digital Health (2025/2)

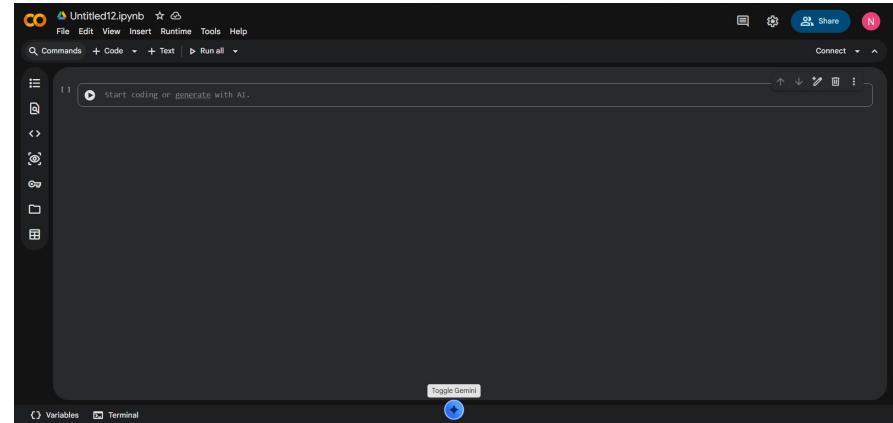
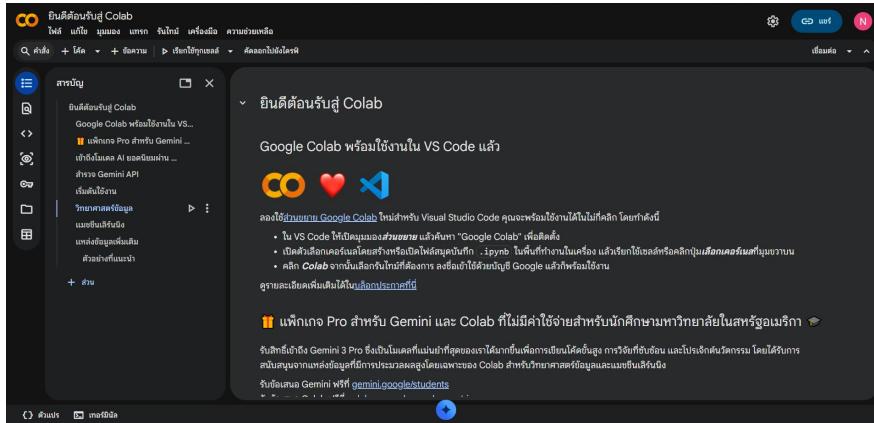
Objective

- **Create & train** object detection model from ultralytics hub and ultralytics library
- **Inference & evaluate** performance each model



Material

- With **Google Colab**, you don't need to install any software. All you need is a Google account, and you can start using it right away. Simply visit: <https://colab.research.google.com/> or select NEW NOTEBOOK to start a new file.



Material



- **Ultralytics HUB** is a web-based platform that can be accessed via a browser at <https://hub.ultralytics.com/>, enabling users to upload datasets, train YOLO models, and monitor performance without any local software installation.

The image displays three screenshots of the Ultralytics HUB platform:

- Landing Page:** Shows the Ultralytics HUB logo and tagline: "Visualize, train, and deploy all your YOLOv5, YOLOv8, and YOLO11 models in one place for free." It features a "Get started for free now!" button and three main navigation tabs: "UPLOAD DATA", "TRAIN A MODEL", and "DEPLOY TO REAL WORLD".
- Sign-In Page:** A standard sign-in form with "Sign Up" and "Sign In" buttons. It includes social login options for Google, GitHub, and Apple, and a "Or with your email" input field. Fields for "Work Email" (name@company.com) and "Password" are also present, along with "Forgot Password?" and "Sign In" buttons.
- Home Dashboard:** The central workspace for managing AI projects. It includes sections for "Recent" datasets (test_segment_33, test_32, test_segment_32), "Storage" (61.9 MB / 20.0 GB), and "Current plan" (FREE). On the left, a sidebar lists "Home", "Datasets", "Projects", "Models", "Integrations", "Support", and "Trash". The main area features a "Welcome Back!" message, a "Get Started" tutorial, and sections for "Datasets", "Projects", and "Models".

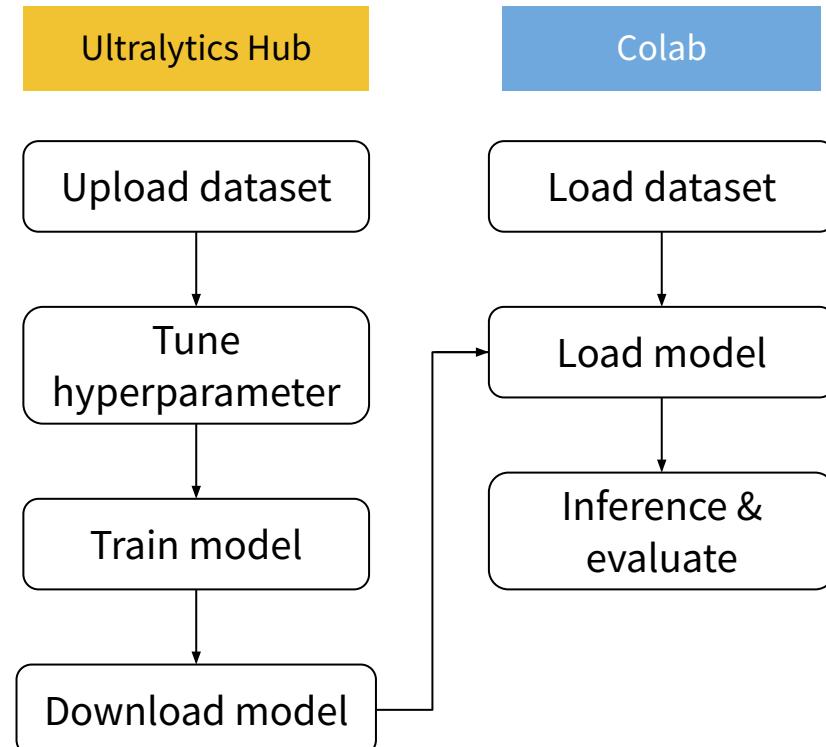
Material



- **Ultralytics is a Python library for computer vision tasks**, best known for developing and training **YOLO (You Only Look Once)** models.
- It supports common tasks such as **image classification, object detection, and image segmentation**.
- Ultralytics models can be trained and used **on a local computer or on Google Colab**.
- Install the library using **pip**

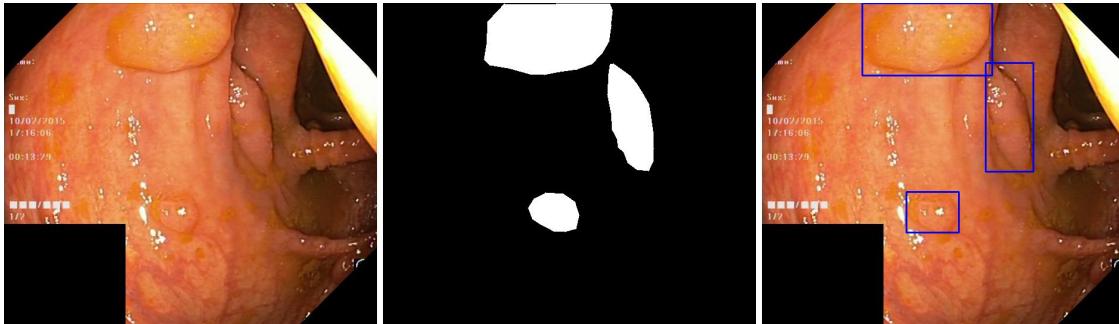
Lab4.1: YOLOv8n (Ultralytics Hub)

In this lab, you will create an image detection model using **Ultralytics HUB** and then evaluate its performance using a **Google Colab notebook**.



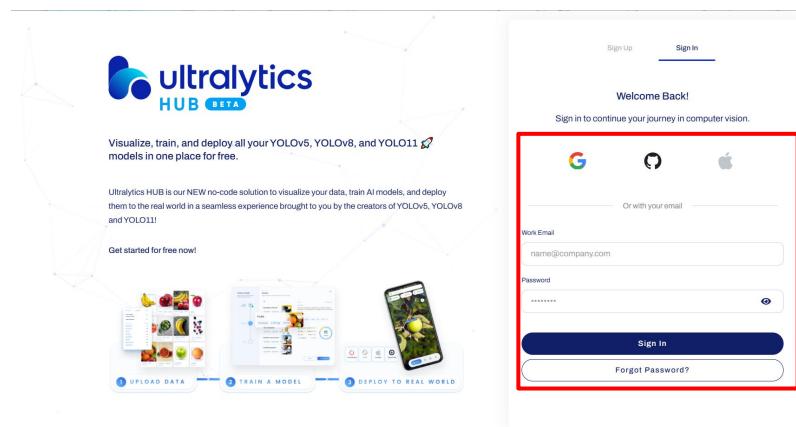
Dataset: kvasir dataset

- **Kvasir dataset** consists of images, annotated and verified by medical doctors (experienced endoscopists), including several classes showing anatomical landmarks, pathological findings or endoscopic procedures in the GI tract.
- The dataset consist of the images with different resolution from 720x576 up to 1920x1072 pixels in JPG format and documents in JSON format
- To simplify the experiment, we selected only 500 images containing polyps and prepared the dataset in a format compatible with YOLO training.



Lab4.1: YOLOv8n (Ultralytics Hub)

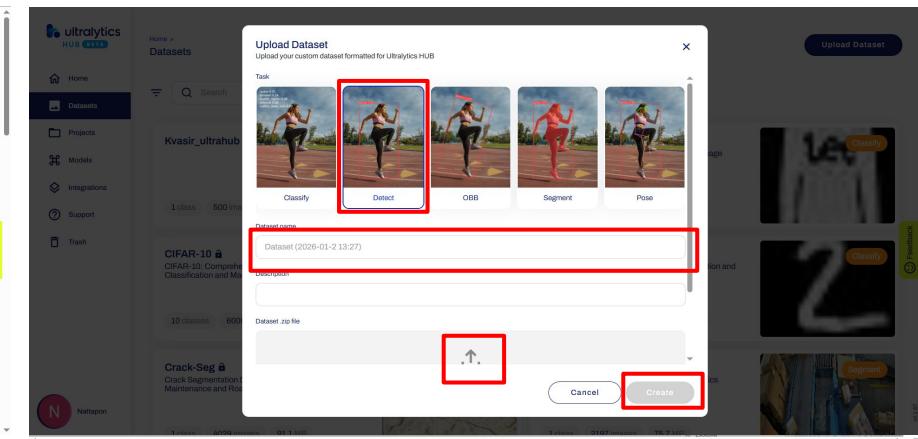
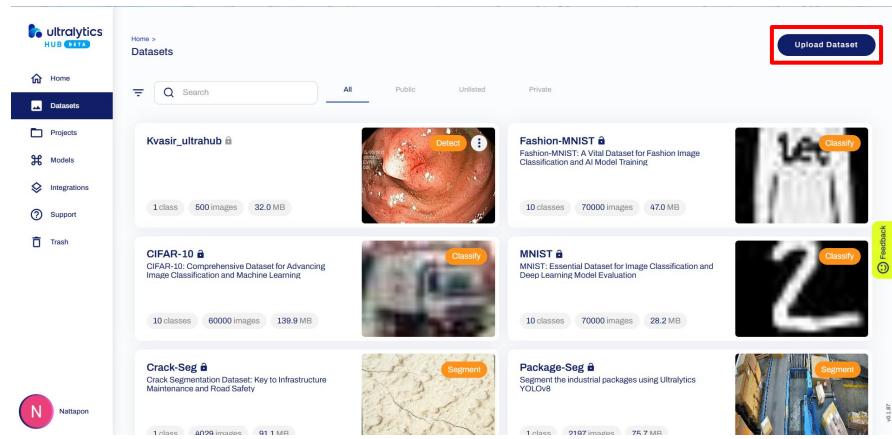
1) Sign in Ultralytic hub



The Ultralytics Hub Home dashboard. On the left, there's a sidebar with navigation links: Home, Datasets, Projects, Models, Integrations, Support, and Trash. A user profile icon for "Nattapon" is at the bottom. The main area has sections for "Get Started" (with a "Start Welcome Tutorial" button), "Datasets" (with a "Upload Dataset" button), "Projects" (with a "Create Project" button), and "Models" (with a "Train Model" button). On the right, there's a "Recent" section showing recent projects like "test_segment_33", "test_32", and "test_segment_32". A "Storage" section shows "61.9 MB / 20.0 GB" with a "Upgrade" button. A "Feedback" button is also present.

Lab4.1: YOLOv8n (Ultralytics Hub)

2) Download the dataset from [GitHub](#), then upload in **Ultralytic hub**.



Lab4.1: YOLOv8n (Ultralytics Hub)

3) Create new project

The image shows two screenshots of the Ultralytics Hub interface. The left screenshot displays the 'Projects' page with a list of pre-existing projects: 'YOLOv11 Segment', 'YOLOv11 Pose', 'YOLOv11 OBB', and 'YOLOv11 Classify'. A red box highlights the 'Create Project' button at the top right of the page. The right screenshot shows a modal window titled 'Create Project' where the 'Project name' field is filled with 'polyp_detection'. A red box highlights this input field. Below it is a 'Description' field and a 'Cover image' section with a placeholder 'Drag and drop your image here or browse your computer'. At the bottom of the modal are 'Cancel' and 'Create' buttons, with a red box highlighting the 'Create' button.

Lab4.1: YOLOv8n (Ultralytics Hub)

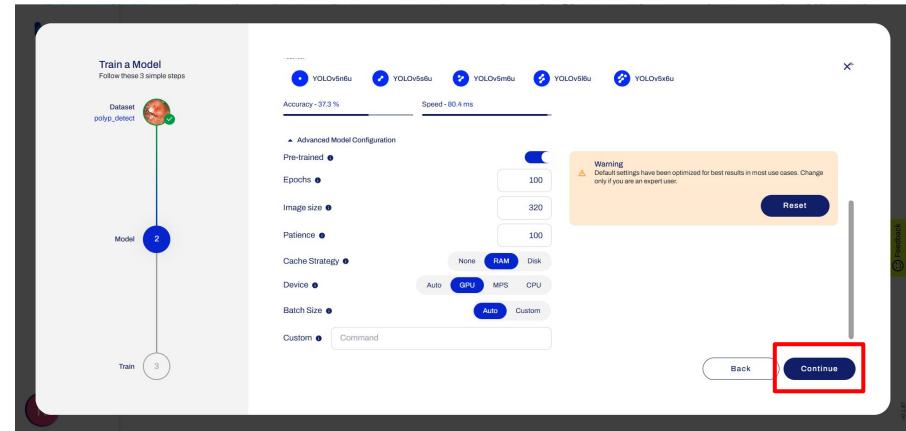
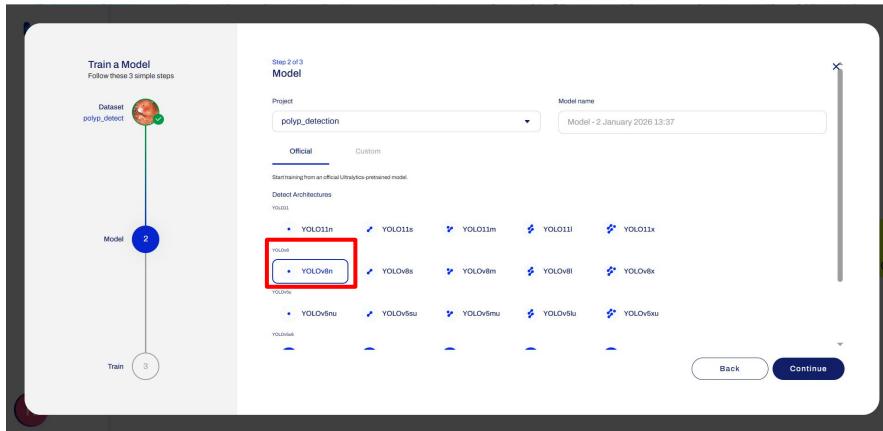
4) Train Model (choose dataset)

The image shows two screenshots of the Ultralytics Hub interface. The left screenshot displays the 'Train Model' screen for a project named 'polyp_detection'. It shows a flowchart with three steps: 'Dataset', 'Model', and 'Train'. A red box highlights the 'Train Model' button. Below the flowchart, a message says 'You don't have any models in this project.' and 'Train a model with Ultralytics YOLO to start your project.' A green 'Project Created' button is at the bottom. The right screenshot shows the 'Dataset' selection step. It lists three datasets: 'polyp_detect' (selected), 'Kvasir_ultr...', and 'CarParts'. The 'polyp_detect' entry has a red box around it. On the right, there's a data split summary: Train (300, 60.0%), Validation (100, 20.0%), Test (100, 20.0%), and Unlabelled (0, 0.0%). A circular progress bar indicates 500 items. A red box highlights the 'Continue' button at the bottom right.

Lab4.1: YOLOv8n (Ultralytics Hub)

4) Train Model (set configuration)

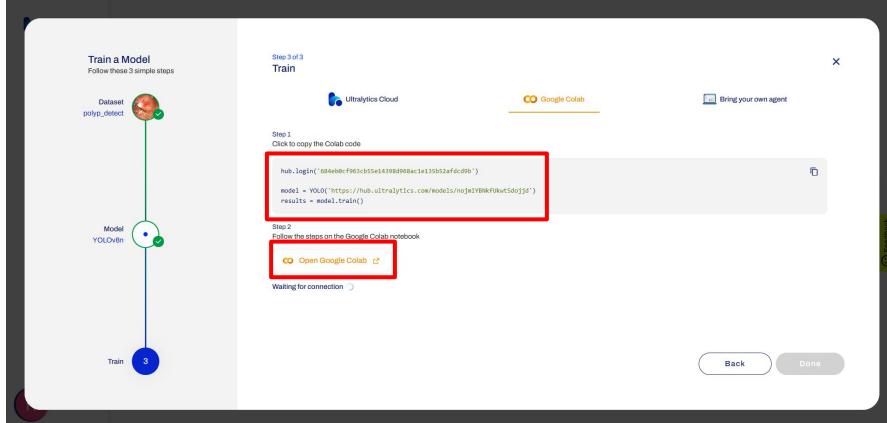
- model: YOLOv8n, Epochs:100, Image size: 320, Cache: RAM, Device: GPU



Lab4.1: YOLOv8n (Ultralytics Hub)

4) Train Model (training)

- Copy code and run in colab



Ultralytics HUB Save in GitHub to keep changes

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all Copy to Drive

Setup

```
Pip install ultralytics and dependencies and check software and hardware.  
pip v3.24! download 147M python 3.8 | 3.9 | 3.10 | 3.11 | 3.12
```

```
[1]: # pip install ultralytics  
from ultralytics import YOLO, checks, hub  
checks() # Verify system setup for Ultralytics training  
ultralytics 8.3.99 Python 3.11.11 torch 2.0.0+cu24 CUDA 10.2 (Tesla T4, 15099MiB)  
Setup complete ✓ (2 CPUs, 12.7 GB RAM, 39.6/112.6 GB disk)
```

Start

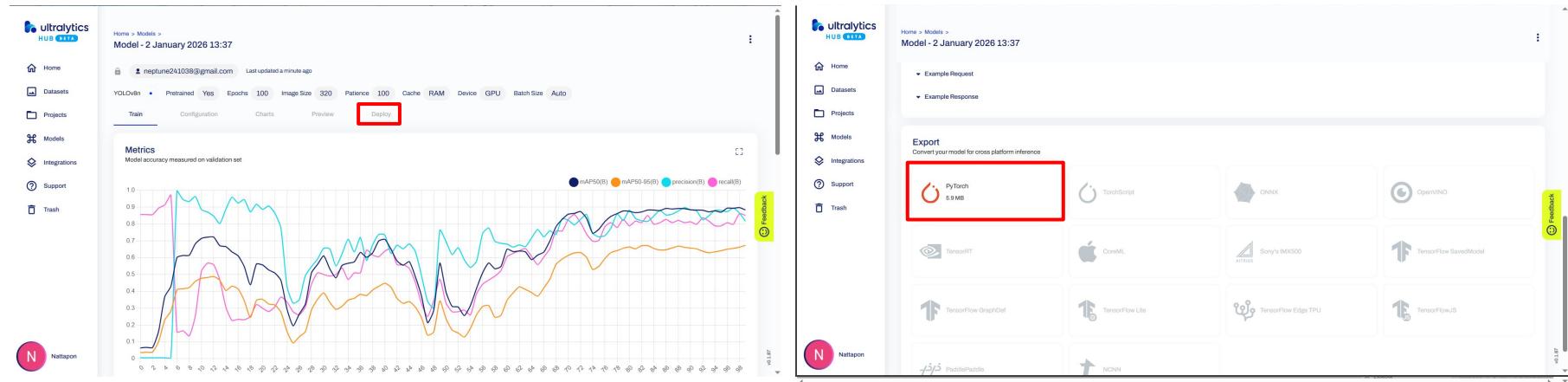
```
# Login with your API key, load your YOLO model, and start training in 3 lines of code!
```

```
[1]: # Login to HUB using your API key (https://hub.ultralytics.com/settings/tb/apikeys)  
hub.login('684e4bcf0d3cb5e14398d08ac1e110b2afcd9b')  
  
# Load your model from HUB (replace 'YOLO_MODEL_ID' with your model ID)  
model = YOLO("https://hub.ultralytics.com/models/nojxtBkRfukw5dejsd")  
  
# Train the model  
results = model.train()
```

Variables Terminal

Lab4.1: YOLOv8n (Ultralytics Hub)

5) Export model (Pytorch version)



Lab4.1: MobileNet (Teachable Machine)

5) Open [Lab_4_1_ultralyricshub](#) (in **colab**)

The screenshot shows a Google Colab notebook interface. The title bar reads "Lab_4_1_ultralyricshub.ipynb". The main content area displays the following text:

Lab 4.1 – Polyp detection: YOLO8 (Ultralytics hub)

This notebook is used to evaluate the performance of polyp detection models trained on Ultralytics hub.

This example code will consist of:

1. Setup
2. Load Dataset
3. Load Model
4. Inference & Evaluate

1) Setup

The code below install and import all required libraries and defines utility functions that will be used in the rest of this notebook.

```
# Download library  
!pip install ultralytics opencv-python tqdm matplotlib -q  
# Import library  
import os  
import shutil
```

At the bottom, there are tabs for "Variables" and "Terminal".

Lab4.2: YOLOv8n (Ultralytics)

Colab

In this lab, you will create an image detection model and evaluate its performance using **Ultralytics library** in colab

Load dataset

Create Model

Train Model

Load best model

Inference & evaluate

Lab4.2: YOLOv8n (Ultralytics)

Run [Lab_4_2_ultralytics.ipynb](#) (in colab)

- 1) Setup
- 2) Load Data
- 3) Train model
- 4) Inference & Evaluate

Lab 4.2 – Polyp detection: YOLO8 (Ultralytics)

This notebook is used to evaluate the performance of polyp detection models trained on Ultralytics hub.

This example code will consist of:

1. Setup
2. Load Dataset
3. Train Model
4. Inference & Evaluate

1) Setup

The code below install and import all required libraries and defines utility functions that will be used in the rest of this notebook.

```
# Download library
!pip install ultralytics opencv-python tqdm matplotlib -q
```

... 1.2/1.2 MB 51.8 MB/s eta 0:00:00

```
# import library
import os
import shutil
```

Variables Terminal

YOLOv8

In this lab, we chose to use YOLOv8, which has the following key architectural innovations:

- **Backbone:** Uses CSPNet with improved **C2f modules** for efficient feature extraction and richer representations.
- **Neck:** Combines **FPN + PAN** to fuse multi-scale features, enabling detection of both small and large objects.
- **Head:** Anchor-free and **decoupled** design, separating classification and localization for higher accuracy.
- **Loss Functions:** Employs task-specific losses (e.g., BCE for classification, DFL for bounding box regression).

YOLOv8

