

Lab4: Object detection (YOLO)

3099704 AI for Digital Health (2025/2)

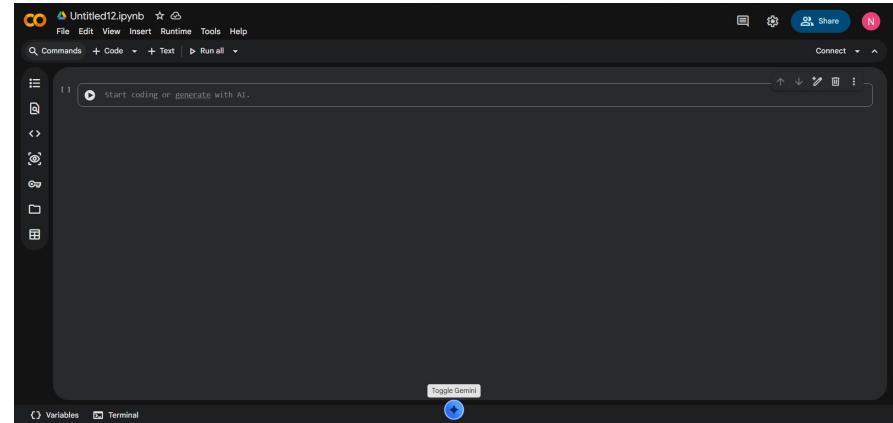
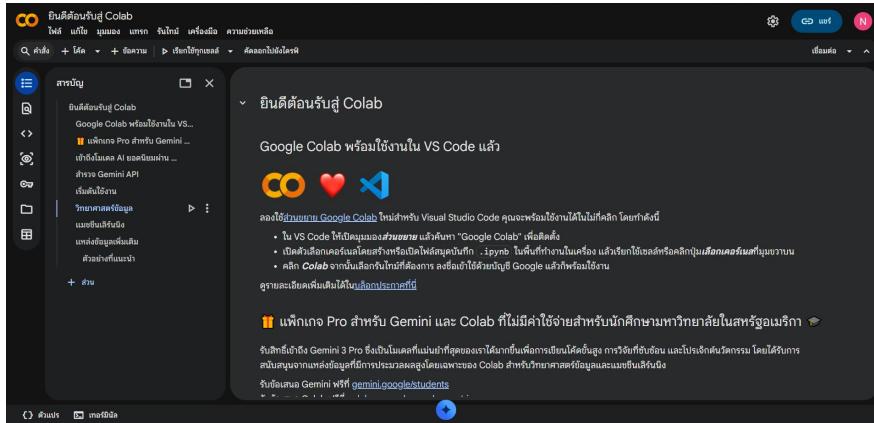
Objective

- **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".

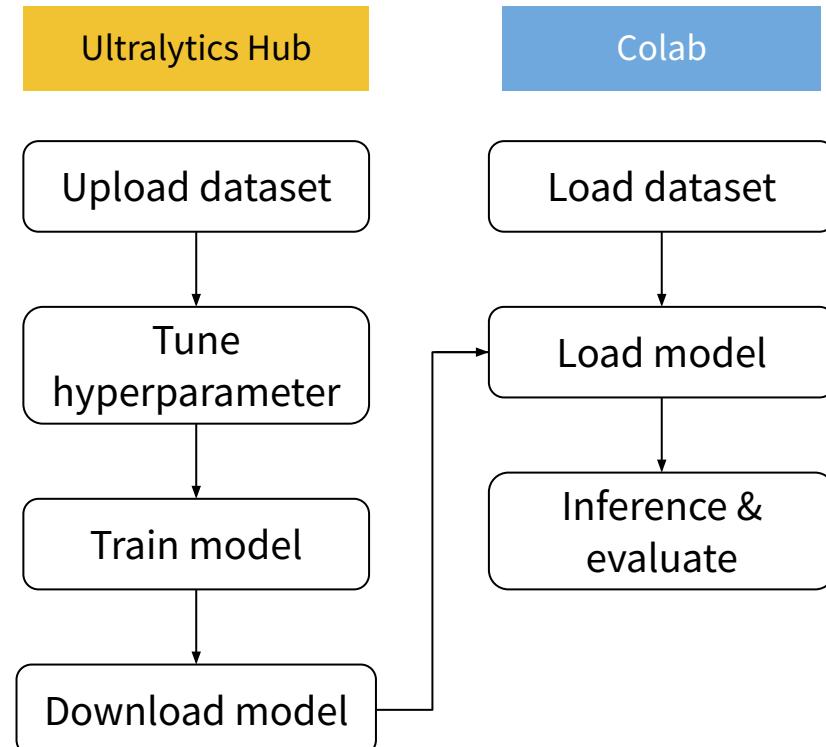
Material



- **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**

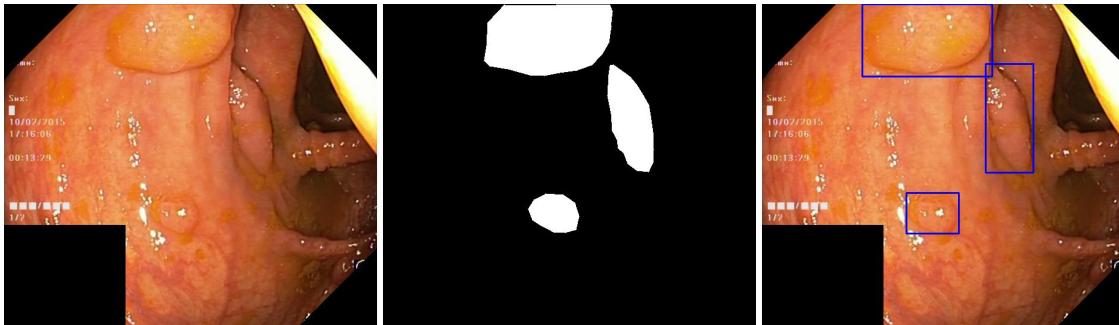
Lab4.1: YOLOv8n (Ultralytics Hub)

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.



kaggle™

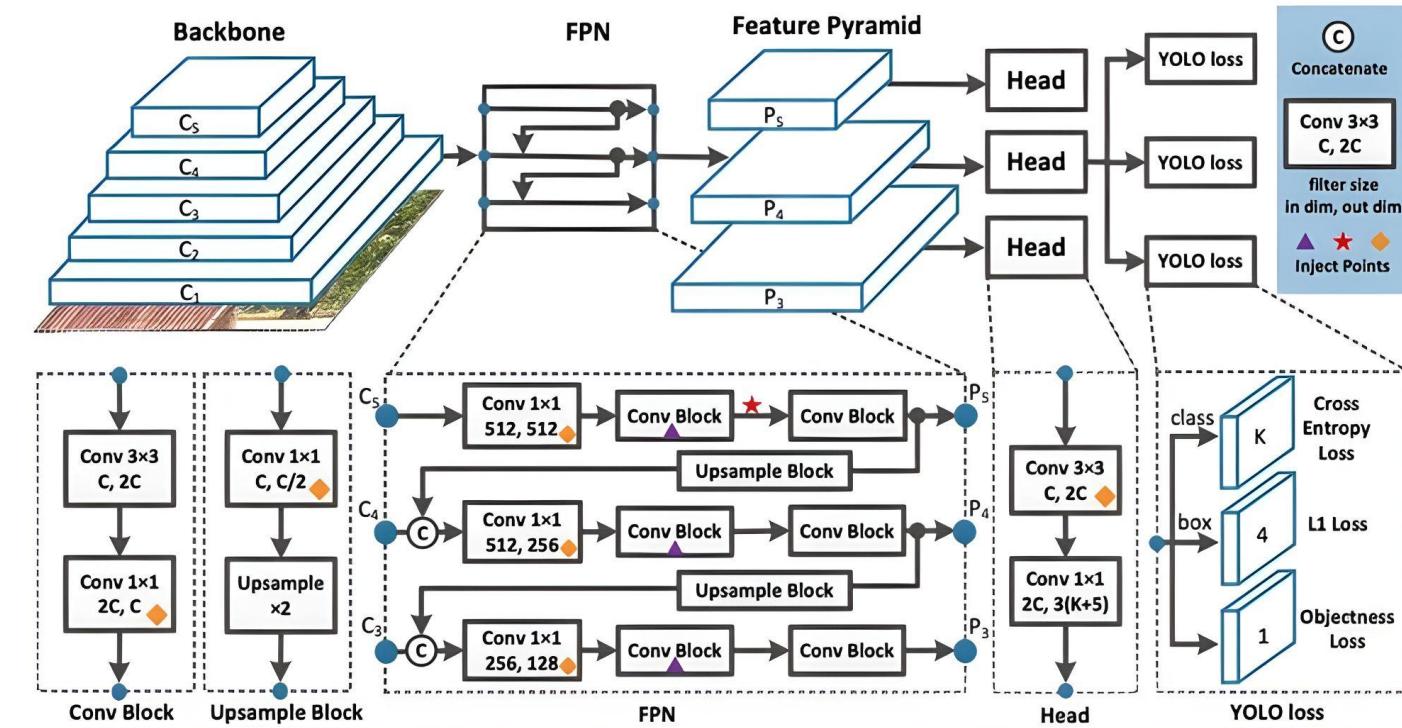


YOLOv8

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



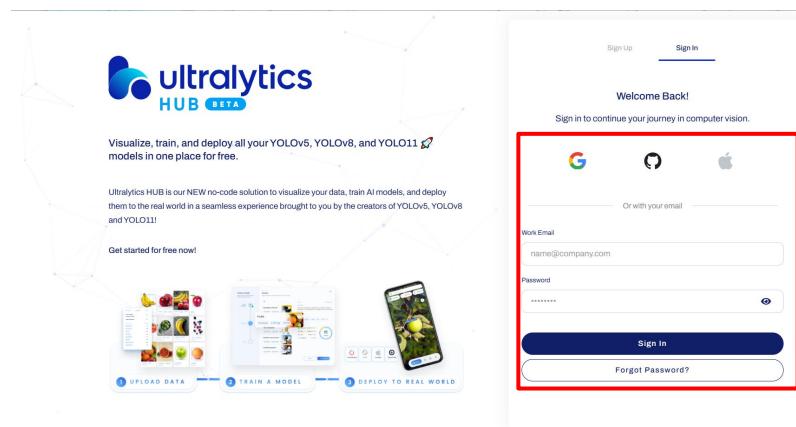
YOLOv8

YOLOv8 is available in multiple model sizes: n (nano), s (small), m (medium), l (large), and x (extra-large). These models share the same overall architecture but differ in the number of channels (width) and repeated blocks (depth). Due to limitations, we use the nano version, which has the lowest computational cost.

Model	size (pixels)	mAP ^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Lab4.1: YOLOv8n (Ultralytics Hub)

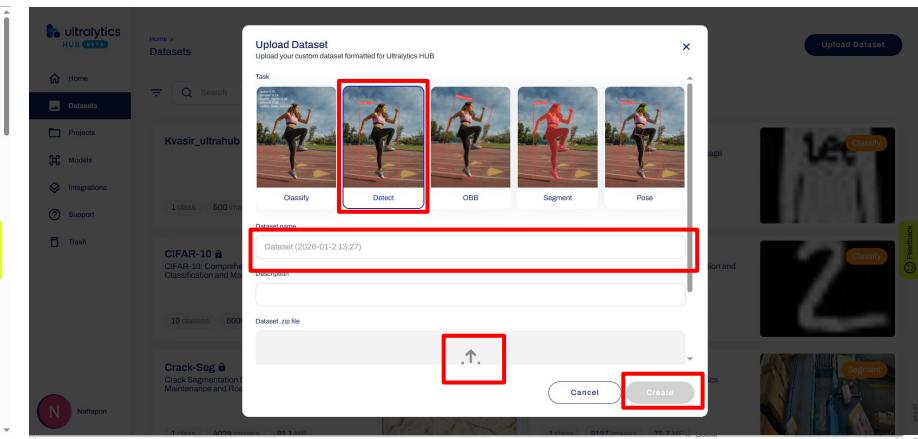
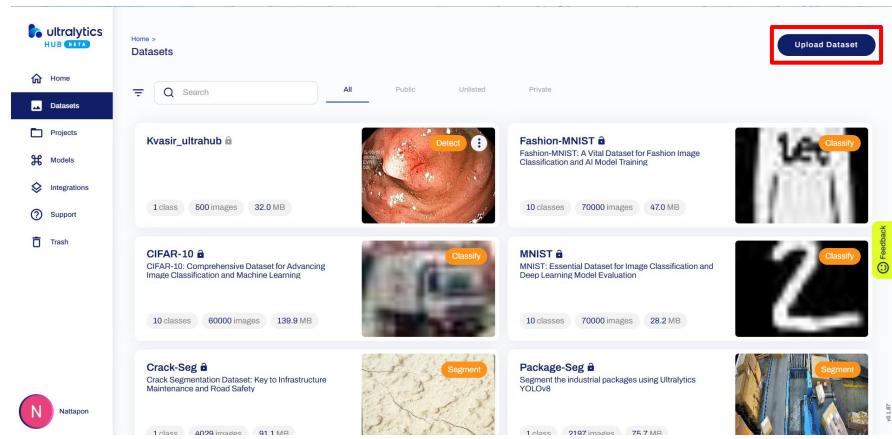
1) Sign in Ultralytic hub



The Ultralytics Hub Home dashboard. It includes a sidebar with navigation links for Home, Datasets, Projects, Models, Integrations, Support, and Trash. A user profile "Nattapon" is shown at the bottom left. 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", and a "Storage" section showing "Hello Nattapon!" and "61.9 MB / 20.0 GB". A "Feedback" button is located at the bottom right.

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, a 'Cover image' section with a placeholder 'Drag and drop your image here or browse your computer', and a 'Create' button at the bottom right, which is also highlighted with a red box.

Lab4.1: YOLOv8n (Ultralytics Hub)

4) Train Model (choose dataset)

The screenshot shows the Ultralytics Hub interface for a project named "polyp_detection".

Left Panel: Shows the project navigation bar with "Home", "Datasets", "Projects", "Models", "Integrations", "Support", and "Train". A "Train Model" button is highlighted with a red box.

Middle Panel: Displays the "Train a Model" workflow with three steps: "Dataset", "Model", and "Train".

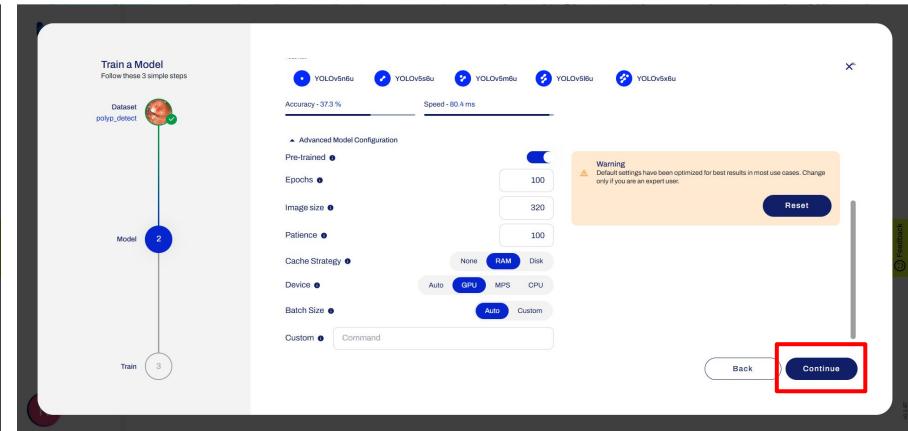
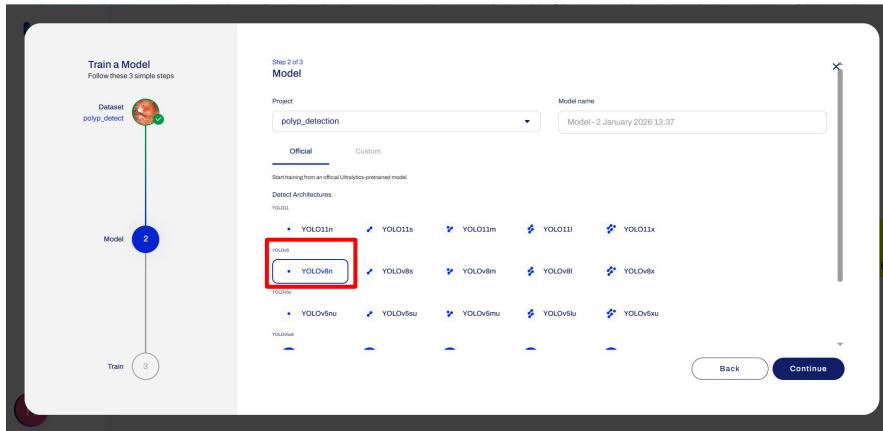
Right Panel: Shows the "Dataset" selection screen with the following details:

- Dataset:** polyp_detect
- Classes (1):** polyp
- Data split:**
 - Train: 300 (60.0%)
 - Validation: 100 (20.0%)
 - Test: 100 (20.0%)
 - Unlabelled: 0 (0.0%)
- Sample Images:** Three images are shown: one labeled "Detected" (polyp), one labeled "Detected" (Kvasir_ultr...), and one labeled "Segment" (CarParts).
- User Information:** Neptune241038@gmail.com
- Summary:** 500 images in total.
- Buttons:** "Continue" (highlighted with a red box) and "Feedback".

Lab4.1: YOLOv8n (Ultralytics Hub)

4) Train Model (set configuration)

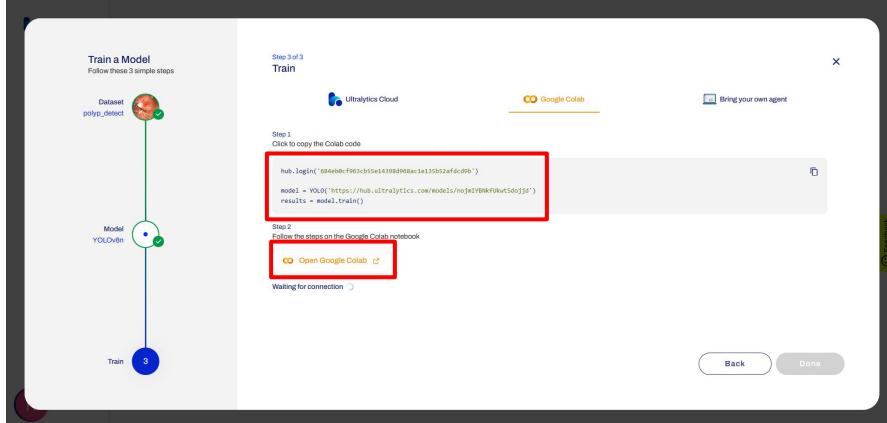
- model: YOLOv8n, Epochs:100, Image size: 320, Cache: RAM, Device: GPU



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-8 (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!
# Login to HUB using your API key (https://hub.ultralytics.com/settings/tb/apikeys)
hub.login('684e4bcf0d3cb5e14398d08ac1e115b2afcd9b')

# 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: YOLOv8n (Ultralytics Hub)

5) Open [Lab_4_1_ultralyricshub](#) in Colab, upload the pt. file exported from Ultralytics Hub, and then run the code in the notebook to evaluate the model's performance.

The screenshot shows a Google Colab interface with the following details:

- Title:** Lab_4_1_ultralyricshub.ipynb
- Toolbar:** File, Edit, View, Insert, Runtime, Tools, Help; Share, Connect, etc.
- Content Area:**
 - Section 1:** Lab 4.1 – Polyp detection: YOLO8 (Ultralytics hub)
Description: This notebook is used to evaluate the performance of polyp detection models trained on Ultralytics hub.
List: This example code will consist of:
 1. Setup
 2. Load Dataset
 3. Load Model
 4. Inference & Evaluate
 - Section 2:** 1) Setup
Description: The code below install and import all required libraries and defines utility functions that will be used in the rest of this notebook.
Code:

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

Output: 1.2/1.2 MB 30.3 MB/s eta 0:00:00

```
# Import library
import os
import shutil
```
- Bottom Navigation:** Variables, Terminal

Lab4.1: YOLOv8n (Ultralytics Hub)

Results may vary between runs due to random seed initialization and hyperparameter tuning; however, the overall performance should be similar to the results shown on this page.

```
Testing 100 test images...
```

```
Evaluating YOLOv8 Detection: 100%|██████████| 100/100 [00:06<00:00, 14.81it/s]
```

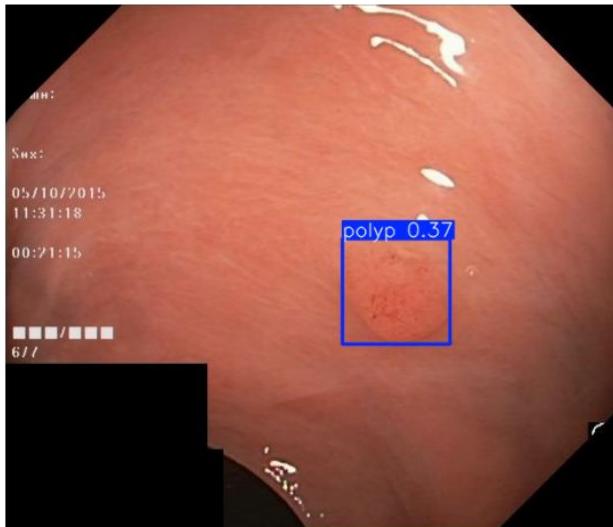
```
YOLOv8 Detection Performance:
```

- Mean IoU: 0.8022
- Avg inference time: 64.15 ms/img

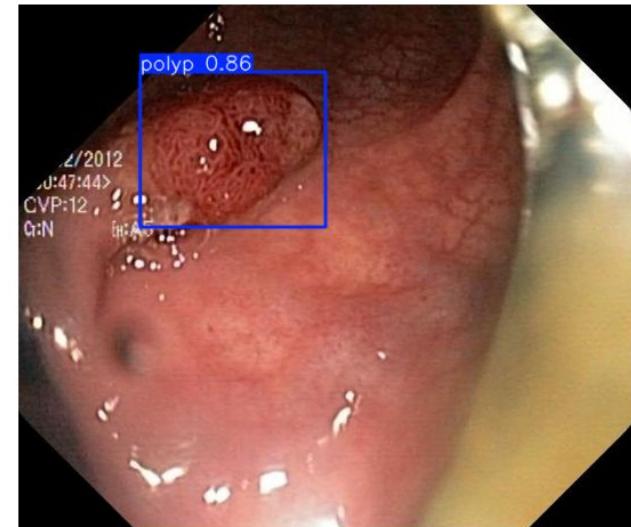
Lab4.1: YOLOv8n (Ultralytics Hub)

Results may vary between runs due to random seed initialization and hyperparameter tuning; however, the overall performance should be similar to the results shown on this page.

Prediction 1



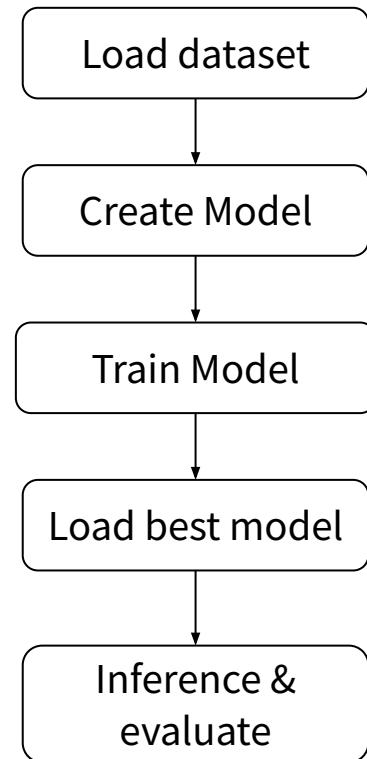
Prediction 2



Lab4.2: YOLOv8n (Ultralytics)

Colab

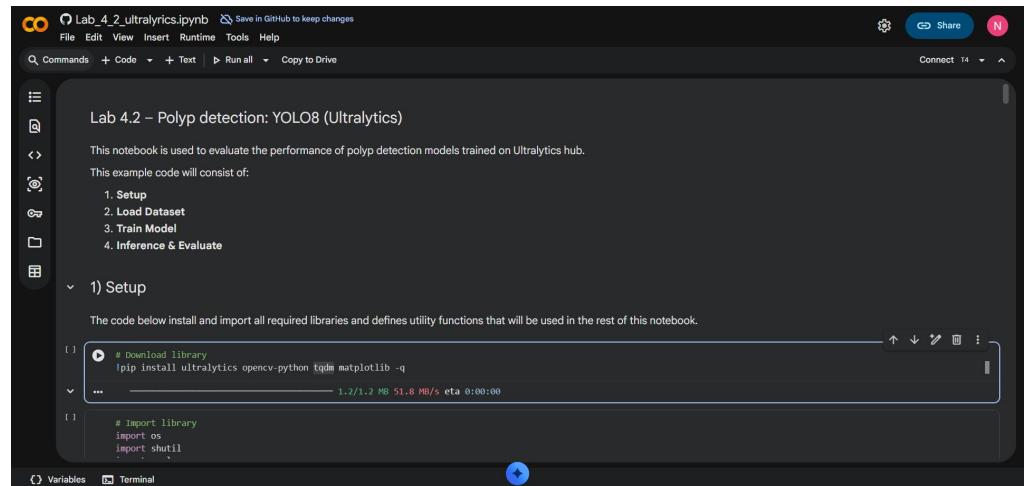
In this lab, you will create an image detection model and evaluate its performance using **Ultralytics library** in colab



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  
...  
# Import library  
import os  
import shutil
```

Lab4.2: YOLOv8n (Ultralytics)

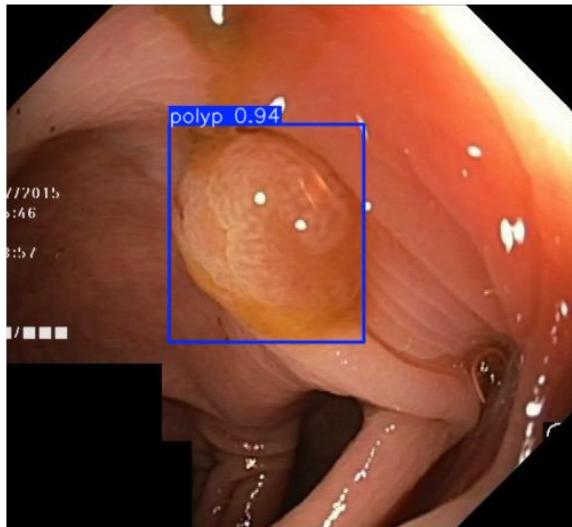
Results may vary between runs due to random seed initialization and hyperparameter tuning; however, the overall performance should be similar to the results shown on this page.

```
Testing 100 test images...
Evaluating YOLOv8 Detection: 100%|██████████| 100/100 [00:01<00:00, 84.49it/s]
YOLOv8 Detection Performance:
• Mean IoU: 0.8439
• Avg inference time: 9.78 ms/img
```

Lab4.2: YOLOv8n (Ultralytics)

Results may vary between runs due to random seed initialization and hyperparameter tuning; however, the overall performance should be similar to the results shown on this page.

Prediction 1



Prediction 2

